

---

الجمهورية الجزائرية الديمقراطية الشعبية

Ministère de L'Enseignement Supérieur et de la Recherche  
Scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département de Mathématiques



MEMOIRE DE MASTER

En Mathématiques

Option : Recherche Opérationnelle

THÈME :

---

Application de l'algorithme d'optimisation chaotique

---

Réalisé par

Boumaza Houria & Hammou Asma

Soutenu devant le Jury :

Mme Djemia Nora	MAA	USD Blida 1	Présidente
Mme Arrache Saida	MAA	USD Blida 1	Examinatrice
Mme Messaoudi Nadia	MCB	USD Blida 1	Promotrice

Septembre 2022

---

DÉDICACES

*Je dédie ce mémoire à :*  
*Mon cher père et ma chère mère*  
*À mes chères sœurs et mes chers frères*  
*A toute ma famille*  
*A toutes mes amies*

Boumaza Houria

## DÉDICACES

Je dédie ce modeste mémoire à :

Ma très chère mère :

Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j'éprouve pour vous, vos sacrifices innombrables et votre dévouement firent pour moi un encouragement. Vous avez guetté mes pas, et m'avez couvé de tendresse, ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Vous m'avez aidé et soutenu pendant de nombreuses années avec à chaque fois une attention renouvelée. Puisse Allah, tout puissant vous combler de santé, de bonheur et vous procurer une longue vie.

Mon père

Mon mari

pour l'amour et l'affection qui nous unissent.

Tu m'as toujours encouragé, incité à faire de mon mieux ton soutien m'as permis de réaliser le rêve tant attendu.

Ma fille TALINE et mon fils YAHIA NAZIM

Ma belle mère ZOUBIDA, mes belles soeurs NAWEL et IMENE et toutes ma belle famille.

Hammou Asma

## REMERCIÉMENTS

*Nous tenons à remercier tout d'abord ALLAH qui nous a donné durant toutes ces années la santé, le courage et la patience pour en arriver là.*

*Nous remercions notre promotrice madame Messaoudi Nadia pour avoir dirigé notre travail. Merci pour vos conseils.*

*Nous tenons à exprimer notre profonde gratitude tous les enseignants de la spécialité Mathématique.*

*Nous remercions également tous les membres du jury pour nous avoir honoré par leur présence pour avoir accepté d'évaluer ce travail.*

*Nous tenons aussi à remercier tous les membres de nos familles et sans oublier nos amies.*

*Enfin, nous remercions tous ceux qui ont participé de près ou de loin à l'élaboration de ce mémoire.*

# TABLE DES MATIÈRES

<b>Introduction Générale</b>	<b>1</b>
<b>1 Notions générales sur les fonctions d'évolutions chaotiques</b>	<b>3</b>
1.1 Généralités . . . . .	3
1.2 Définition du chaos . . . . .	4
1.3 Histoire de chaos . . . . .	4
1.4 Définitions mathématiques du chaos . . . . .	5
1.5 Fonctions d'évolutions chaotiques . . . . .	6
1.5.1 Application de Lozi . . . . .	6
1.5.2 Application Logistique . . . . .	8
1.5.3 Application Tente . . . . .	9
1.5.4 Application Circulaire . . . . .	9
1.6 Quelques applications de la théorie du chaos . . . . .	10
1.6.1 Médecine . . . . .	11
1.6.2 Biologie . . . . .	11
1.6.3 Ingénierie . . . . .	12
1.6.4 Informatique . . . . .	12
1.6.5 Communication . . . . .	12
1.6.6 Management et finance . . . . .	12
<b>2 Méthode d'optimisation chaotique</b>	<b>13</b>
2.1 Définitions . . . . .	14
2.2 Méthodes d'optimisation globale . . . . .	15
2.2.1 Approches déterministes et approches probabilistes . . . . .	15
2.2.2 Méthodes de descente généralisées . . . . .	16
2.2.3 Recherches aléatoires . . . . .	17
2.2.4 Méthode de multi-départ . . . . .	17

2.2.5	Méthodes de regroupement . . . . .	18
2.2.6	Méthodes évolutionnaires . . . . .	18
2.2.7	Généralités sur les méthodes d'optimisation globale . . . . .	19
2.3	Les Algorithmes de recherche chaotiques . . . . .	20
2.3.1	La Recherche chaotique de l'optimum . . . . .	20
<b>3</b>	<b>Algorithme de recherche chaotique et résultats numériques</b>	<b>22</b>
3.1	Principe et étapes de l'algorithme d'optimisation chaotique . . . . .	23
3.2	Simulation numériques . . . . .	25
3.2.1	Les fonctions de tests . . . . .	25
3.2.1.1	La fonction de Rastrigin . . . . .	25
3.2.1.2	La fonction de Griewank . . . . .	26
3.2.1.3	La fonction de Sphère . . . . .	26
3.2.2	Résultats numériques . . . . .	27
3.2.2.1	Résultats pour la fonction de Rastrigin . . . . .	27
3.2.2.2	Résultats pour la fonction de Griewank . . . . .	28
3.2.2.3	Résultats pour la fonction de Sphère . . . . .	29
	<b>Conclusion Générale</b>	<b>31</b>
	<b>ANNEXE</b>	<b>32</b>
	<b>Bibliographie</b>	<b>41</b>

## TABLE DES FIGURES

1.1	l'application de Lozi pour $a = 1.3$ et $b = 0.4$ . . . . .	7
1.2	l'application de Lozi pour $a = 1.7$ et $b = 0.5$ . . . . .	7
1.3	L'application logistique pour $a = 2$ . . . . .	8
1.4	L'application logistique pour $a = 4$ . . . . .	8
1.5	L'application de Tente . . . . .	9
1.6	L'application Cercle pour $a = 0.7$ et $b = 0.1$ . . . . .	10
1.7	L'application Cercle pour $a = 0.5$ et $b = 0.2$ . . . . .	10
2.1	Exemple d'une fonction multimodale . . . . .	14
2.2	Minima locaux et minima globaux d'une fonction multimodale . . . . .	15
2.3	Comparaison entre la recherche aléatoire et recherche chaotique (où $x^* = x_n, f^* = y_n$ ) . . . . .	21
3.1	Fonction de Rastrigin . . . . .	26
3.2	Fonction de Griewank . . . . .	26

## LISTE DES TABLEAUX

3.1	L'optimum de la fonction de Rastrigin pour $n = 3$ avec différents chaos map. . . . .	27
3.2	L'optimum de la fonction de Rastrigin pour $n = 4$ avec différents chaos map. . . . .	28
3.3	L'optimum de la fonction de Griewank pour $n = 3$ avec différents chaos map. . . . .	28
3.4	L'optimum de la fonction de Griewank pour $n = 4$ avec différents chaos map. . . . .	29
3.5	L'optimum de la fonction de Sphère pour $n = 2$ avec différents chaos map.	29
3.6	L'optimum de la fonction de Sphère pour $n = 3$ avec différents chaos map.	30

Ce travail consiste à utiliser un algorithme d'optimisation chaotique pour résoudre des problèmes d'optimisation. L'algorithme d'optimisation chaotique est une méthode d'optimisation stochastique qui donne une solution globale.

Cet algorithme utilise directement les variables chaotiques pour rechercher la solution optimale. La sensibilité aux conditions initiales et la propriété stochastiques du chaos font l'optimisation chaotique pour mieux obtenir la solution optimale globale que d'autres méthodes ayant été adoptée avant. Il peut facilement s'échapper de minima locaux que les autres algorithmes stochastiques.

Nous avons appliqué cet algorithme sur trois fonctions tests (Benchmark) : fonction Griewank, fonction Sphère, fonction Rastrigin. Dont on génère chaotiquement les variables de ces fonctions selon quatre applications chaotiques : Logistic map, Circl map, Lozi map, Tente map.

Nous avons réalisé un programme en MATLEB qui donne une solution globale aux trois fonctions tests.

Les résultats obtenus par l'algorithme sont satisfaisants pour l'optimisation des différents fonctions tests.

**Mots clés :** Algorithme d'optimisation chaotique, Chaos, optimisation globale.

This work consists of using chaotic optimization algorithm for solving optimization problems.

Chaotic optimization algorithm is an stochastic optimization method that gives a global solutions. This algorithm uses directly the chaotic variables to search the optimal solution. The sensitive dependence on initial conditions and intrinsic stochastic property of chaos make chaotic optimization to obtain the global optimal solution more possible than the method having been adopted before. It can more easily escape from local minima than other stochastic algorithms.

We have applied the algorithm chaotic under three benchmark functions : Griewank function, Sphere function and Rastrigin function, in each run the variables of these functions are generating chaotically according to for chaotic maps : Logistic map, Circle map, Lozi map and Tent map.

The algorithm was simulated by using a developed program in MATLAB that gives global solution for three benchmark functions.

The obtained results of the simulation algorithm are satisfies for the optimization of benchmark functions.

**Keywords :** Chaos optimization algorithm, Chaos, global optimization.

## ملخص

يتمثل هذا العمل من استعمال خوارزمية تحسين الفوضى لحل مشكلات التحسين. خوارزمية التحسين الفوضوي هي طريقة تحسين عشوائية تعطي حلا عاما. تستخدم هذه الخوارزمية مباشرة المتغيرات الفوضوية للعثور على الحل الأمثل. الحساسية للظروف الأولية والخاصية العشوائية للفوضى تجعل التحسين الفوضوي الافضل في ايجاد الحل الأمثل من الطرق الأخرى التي تطرقنا اليها من قبل. يمكن أن يتجنب بسهولة الحد المحلي الأدنى عكس الخوارزميات العشوائية الأخرى. قمنا بتطبيق هذه الخوارزمية على ثلاث دوال اختبار: الدالة Griewank، الدالة Rastrigin و الدالة Sphère حيث يتم إنشاء متغيرات هذه الوظائف بشكل عشوائي وفقاً لأربعة تطبيقات فوضوية: Logistic map, Lozi map, Tent map, Circl map. قمنا بانجاز برنامج في Matlab يقدم حلا عاما لدوال الاختبار الثلاث. النتائج التي تحصلنا عليها من الخوارزمية مرضية لتحسين دوال الاختبار المختلفة.

لكلمات المفتاحية: خوارزمية التحسين الفوضوي، فوضى، تحسين عام.

## INTRODUCTION GÉNÉRALE

Durant ces dernières années, de nombreuses études ont été menées dans le domaine de l'optimisation comme le montre le nombre important de publications sur ce thème dans les revues du domaine. Aujourd'hui, l'optimisation s'applique à tous les domaines de la science et même à notre vie quotidienne. Chacun cherche souvent à mieux gérer son temps, son argent, minimiser certaines consommations, ce sont autant de problèmes d'optimisation.

En fait, l'optimisation cherche à améliorer une performance en se rapprochant d'un point optimum. En mathématique, la meilleure solution (ou optimum) se recherche au sein d'un domaine initial. Cette solution est souvent soumise à des contraintes qui correspondent à des obligations ou des souhaits à respecter. Le critère permettant de distinguer qu'une solution est la meilleure s'appelle la fonction objectif. L'optimisation mathématique consiste à chercher dans le domaine initial une solution qui maximise ou minimise une fonction objectif tout en respectant des contraintes. Pour un domaine continu, on distingue classiquement deux types d'optimisation [12] :

- **L'optimisation locale** est la recherche d'une solution qui est meilleure localement (dans un voisinage de cette solution). Cette solution est appelée un optimum local.
- **L'optimisation globale** est la recherche de la meilleure solution sur tout le domaine ; c'est-à-dire que dans tout le domaine il n'existe aucune solution qui lui soit meilleure tout en respectant les contraintes. Cette solution est appelée l'optimum global. Par définition, l'optimum global est aussi une solution locale. En revanche, il est bien plus épineux de trouver l'optimum global, car lorsque l'on pense avoir trouvé cet optimum, sa démonstration se révèle bien souvent particulièrement ardue.

L'optimisation garantit en effet que personne ne peut avoir une solution meilleure que celle trouvée. Or, pour une entreprise, cette information a son importance, car la différence entre la solution globale et une solution locale est bien souvent significative.

Dans de nombreux problèmes, l'optimum global est la seule solution mathématique correspondant à une réalité physique. C'est ce qu'illustre par exemple la recherche de

la quantité de chaque élément présent dans un mélange chimique à l'équilibre.

Afin de résoudre des problèmes d'optimisation globale de nombreux méthodes existent. Selon la nature du processus de recherche mis en oeuvre, les méthodes employées pour l'optimisation globale sont classées en deux catégories : les méthodes déterministes et les méthodes stochastiques.

Les méthodes déterministes requièrent des hypothèses restrictives sur la fonction à optimiser, telles que la continuité, la dérivabilité ou les conditions de Lipchitz. De plus, ces méthodes ne sont applicables que pour des problèmes de faible dimension. Pour les autres problèmes, seules les méthodes stochastiques peuvent être utilisées. Les méthodes stochastiques, contrairement aux méthodes déterministes, sont basées sur une approche en partie ou entièrement guidée par un processus stochastique. Les algorithmes que nous allons étudier font partie des méthodes stochastiques globales permettant de résoudre des problèmes généraux d'optimisation. Par ailleurs, ils permettent de fournir le plus haut degré de certitude concernant l'optimalité globale des solutions trouvées. Ces algorithmes s'appellent : Algorithmes d'Optimisation Chaotiques (en anglais chaotic optimisation algorithm).

D'après le travail de Caponetto et al.[6, 25], des applications du chaos ont attiré beaucoup d'attention. Grâce aux propriétés naturelles des systèmes chaotiques, telles que leur sensibilité aux conditions initiales et leur évolution dans une large bande de fréquence, les systèmes chaotiques sont de bons candidats pour l'optimisation.

Notre mémoire s'articule autour de trois chapitres :

Le premier chapitre, présente les fonctions d'évolutions chaotiques (ou applications chaotiques), ainsi que quelques applications de la théorie du chaos.

Dans chapitre deux on donne des définitions sur l'optimisation, ensuite nous présentons quelques méthodes d'optimisation globales.

Le troisième chapitre donne le principe de l'algorithme d'optimisation chaotique (COA), et la définition de trois fonctions tests. Cet algorithme est appliqué sur ces fonctions dont les variables de ces fonctions sont générées chaotiquement par des applications chaotiques (en anglais chaotic maps) : Logistic map, Lozi map, Tente map et cercle map. On présente également les résultats numériques obtenus.

On termine notre travail par une conclusion.

# CHAPITRE 1

## NOTIONS GÉNÉRALES SUR LES FONCTIONS D'ÉVOLUTIONS CHAOTIQUES

### 1.1 Généralités

La théorie du chaos[44] s'applique à de nombreux modèles développés pour expliquer des situations rencontrées dans presque tous les domaines de la connaissance scientifique. L'étude du chaos touche donc de larges champs de recherche : la physique, la psychologie, l'économie et la biologie. Elle donne un cadre mathématique permettant une étude quantitative (chiffrée) de phénomènes auparavant étudiés de manière qualitative.

La théorie du chaos décrit le comportement des systèmes aperiodiques sensibles aux conditions initiales. Dans ces systèmes, un petit changement dans les conditions initiales a d'importantes conséquences sur le comportement à moyen terme.

La théorie du chaos consiste à étudier le comportement des systèmes dynamiques non linéaires[17] représenté par des équations déterministes qui peuvent avoir un comportement chaotique si elles sont utilisées de façon itérative en des suites dites chaotiques [21]. Le comportement chaotique de ces suites est dû à la forte sensibilité et dépendance à leurs paramètres et conditions initiales [17].

Ce chapitre présente des généralités sur le chaos, l'histoire et des définitions mathématiques du chaos. Il présente aussi quelques fonctions d'évolutions chaotiques et quelques applications concrètes de la théorie du chaos.

## 1.2 Définition du chaos

Le mot chaos [24] n'a pas ici le même sens que l'usage dans la vie courante.

On retrouve la trace de ce mot du grec Khaos dans les écrits de Christine de Pisan (Chemin de long estude) qui définit le chaos comme un " état de confusion des éléments ayant précédé l'organisation du monde "

Au XVIème siècle, Desportes le décrit dans ses Elegies comme " Toute sorte de confusion, de désordre "

Le chaos, dans son sens familier aujourd'hui, c'est le désordre et la violence, mais aussi l'inintelligibilité.

Loin de ces considérations historiques et mythologiques, chaos : un terme souvent utilisé comme métaphore du désordre.

## 1.3 Histoire de chaos

Dans le milieu scientifique, le concept a émergé dans la seconde partie des années 1970 [10] en tant que science des phénomènes non linéaires complexes montrant certaines caractéristiques communes, le mathématicien Henri Poincaré qui a démontré que certains systèmes mécaniques, dont l'évolution temporelle et gouvernée par des équations hamiltoniennes, peuvent exhiber un mouvement chaotique, malheureusement ceci fut considéré par beaucoup de physiciens comme simple curiosité. Il a fallu 70 ans pour que le météorologiste E.N.Lorenz (1963) découvre que même un simple ensemble de trois équations (non linéaire couplées le premier ordre) peut donner lieu à des trajectoires complètement chaotiques. Ainsi, Lorenz a mis en évidence un des premiers exemples de chaos déterministe dans les systèmes dissipatifs. Dans les années qui ont suivi, à cause des résultats théoriques, de la puissance incrémentale des ordinateurs, et des techniques expérimentales de plus en plus raffinées, il est devenu vraisemblable que ce phénomène est abondant dans la nature et a beaucoup de conséquences et de ramifications dans de nombreux domaines scientifiques. Il faut noter que la non linéarité est une condition nécessaire, mais pas suffisante pour générer le chaos.

En 1890 Le Roi Oscar 2 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 2. Etant le plus proche à résoudre le problème de n- corps. Il a découvert que l'orbite de trois corps célestes agissants l'une sur l'autre peut engendrer un comportement instable et imprévisible. Ainsi, le chaos naît (mais pas encore mentionné!).

En 1963 Edward Lorenz découvre le premier système chaotique dans la météo ou encore appelé attracteur étrange.

En 1975, Tien-yien Li et James A. Yorke ont présenté pour la première fois le terme "chaos" dans un article intitulé "Period three implies chaos".

En 1978, Mitchell Feigenbaum a introduit un nombre universel associé au chaos.

En 1990, Lou Pecora a réalisé la synchronisation des systèmes chaotiques.

## 1.4 Définitions mathématiques du chaos

**Définition 1.1** [49] :

L'application  $z : J \rightarrow J$  possède une sensibilité aux conditions initiales s'il existe  $\delta > 0$  tel que, pour un certain  $x \in J$  et un certain voisinage  $V \subset J$  de  $x$ , il existe  $y \in V$  tel que  $\|z^n(x) - z^n(y)\| > \delta$ .

Avant d'entamer la notion d'irréductibilité, il est nécessaire d'introduire la notion d'ensemble dense.

**Définition 1.2** [49] : (Ensemble dense)

Supposons que  $X$  est un ensemble et  $Y$  un sous-ensemble de  $X$ .  $Y$  est dense dans  $X$  si, pour n'importe quel élément  $x \in X$ , 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  $X$  :  $\overline{Y} = X$ . Ce qui revient à dire que  $Y$  est dense dans  $X$  si pour tout  $x \in X$  on peut trouver une séquence  $\{y_n\} \in Y$  qui convergent vers  $x$ .

### Exemple

l'ensemble des nombres rationnels et l'ensemble des nombres irrationnels sont denses dans l'ensemble des nombres réels.

### Remarques

- L'irréductibilité empêche le système chaotique d'être décomposé en sous-systèmes (sous-ensembles ouverts invariants) qui n'interagissent pas sous l'application. Cette propriété s'appelle la transitivité topologique. Autrement dit, une application transitive topologiquement possède des points qui évoluent d'un petit voisinage arbitraire vers n'importe quel autre.
- Notons qu'une application possédant une orbite dense est transitive topologiquement et que l'inverse est également vrai.

**Définition 1.3** : La définition du chaos au sens de Devaney [33].

L'application  $z : J \rightarrow J$  est dite *transitive topologiquement* si pour toute paire d'ensembles ouverts  $U, V \subset J$ , il existe  $k > 0$  tel que  $z^k(U) \cap V \neq \emptyset$ .

### Remarque

- Contrairement aux comportements purement aléatoires, les systèmes chaotiques possèdent une certaine régularité qui se traduit par le fait que les points périodiques sont denses. La densité des points périodiques exprime l'infinité des comportements dynamiques que prodigue le chaos. Nous sommes maintenant prêts pour donner une définition, ou plutôt une caractérisation du chaos :

**Définition 1.4** [49] : Soit un ensemble  $V$ , l'application  $z : J \rightarrow J$  est dite *chaotique sur  $V$*  si

1.  $z$  possède une sensibilité aux conditions initiales,
2.  $z$  est topologiquement transitive,
3. les points périodiques sont denses dans  $V$ .

Cette définition est intéressante car les concepts sur lesquels elle repose sont facilement observables. De plus, elle s'applique à un très grand nombre de systèmes dynamiques chaotiques et dans certains cas elle est même facilement vérifiable.

## 1.5 Fonctions d'évolutions chaotiques

### 1.5.1 Application de Lozi

L'application de Lozi (en anglais Lozi Map) peut être présentée comme suit[26] :

$$y_1(k) = 1 - a \cdot |y_1(k-1)| + y(k-1)$$

$$y(k) = b \cdot y_1(k-1)$$

$$z(k) = \frac{y(k) - \alpha}{\beta - \alpha} \tag{1.1}$$

où  $k$  désigne le numéro d'itérations, la valeur de  $z$  est dans l'intervalle  $[0, 1]$ . Ainsi  $y \in [-0.6418, 0, 6716]$  et  $[\alpha, \beta] = (-0, 6418, 0, 6716)$ .

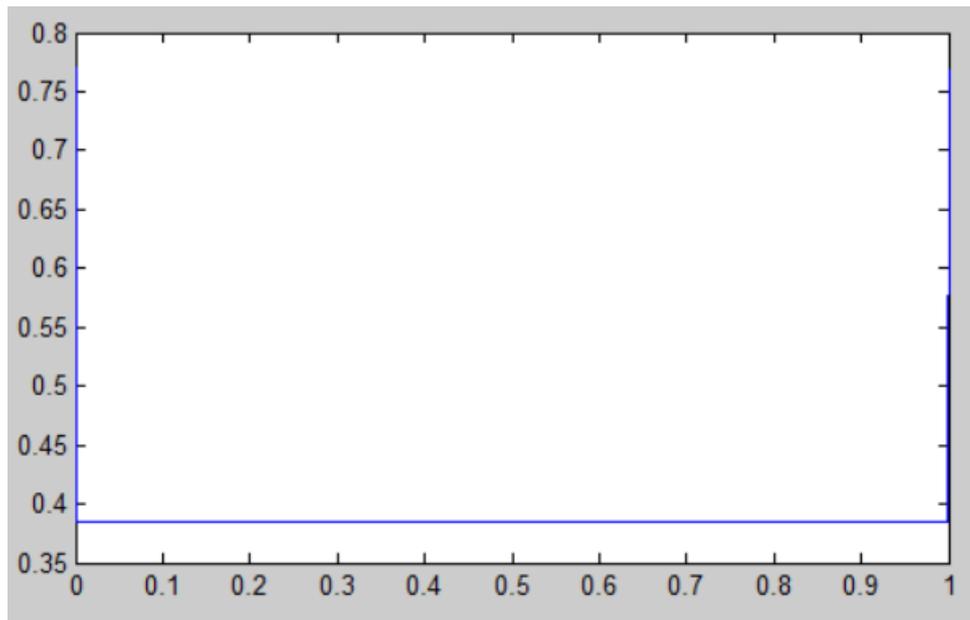


FIGURE 1.1 – l'application de Lozi pour  $a = 1.3$  et  $b = 0.4$

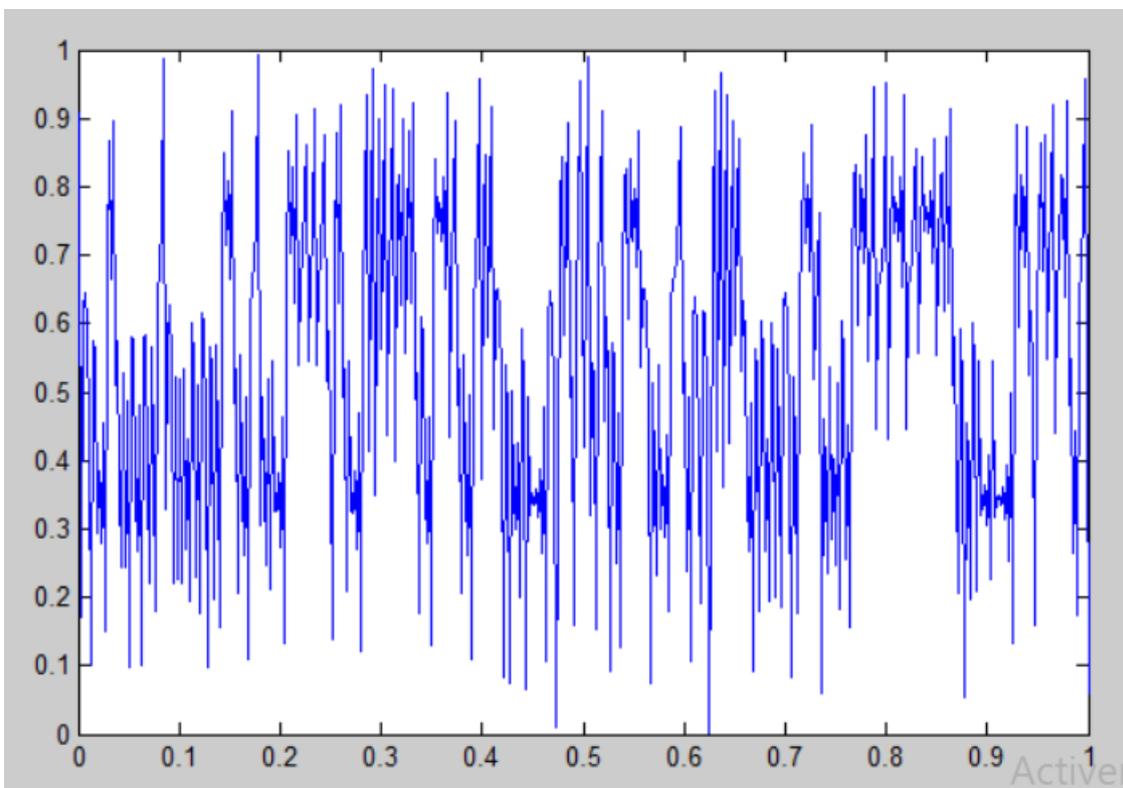


FIGURE 1.2 – l'application de Lozi pour  $a = 1.7$  et  $b = 0.5$

L'application de Lozi est chaotique pour  $a = 1,7$  et  $b = 0,5$ .

## 1.5.2 Application Logistique

L'application logistique ou (en anglais Logistic Map) est l'une des cartes les plus simples qui a été portée à l'attention des scientifiques par Sir Robert May en 1976 [27] qui est définie comme suit :

$$X_{k+1} = aX_k(1 - X_k) \quad (1.2)$$

où  $X_k \in [0,1]$  ,  $n$  désigne le numéro d'itérations, sous la condition que le point initiale  $X_0 \in [0,1]$  .

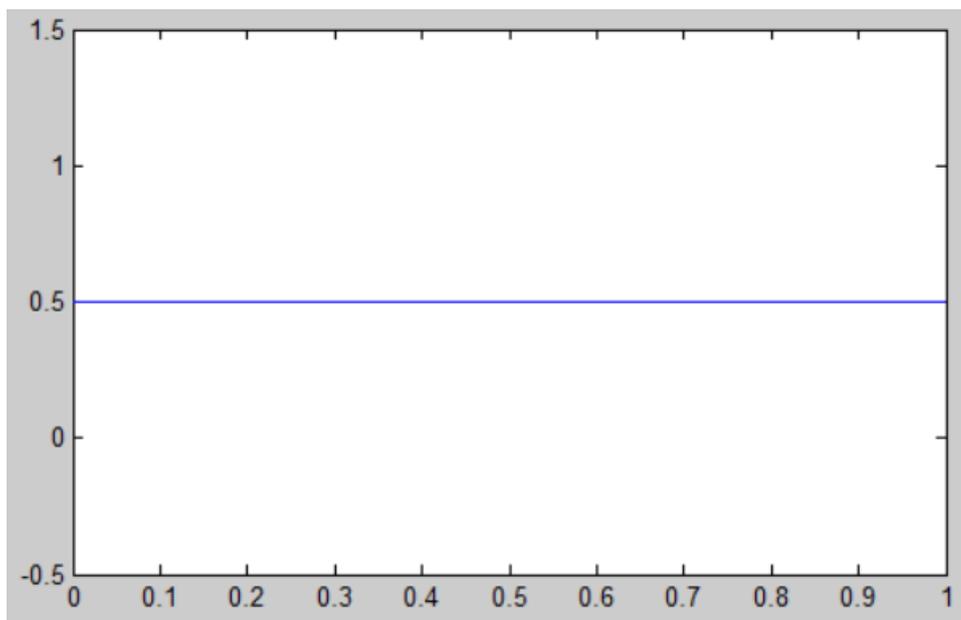


FIGURE 1.3 – L'application logistique pour  $a = 2$

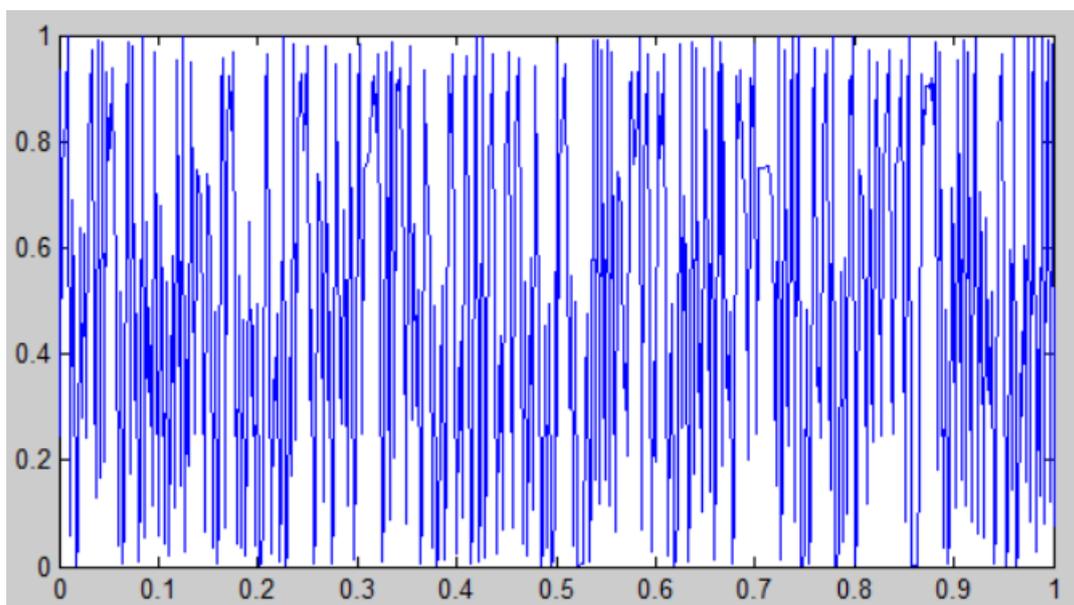


FIGURE 1.4 – L'application logistique pour  $a = 4$

L'application logistique est chaotique pour  $a = 4$ .

### 1.5.3 Application Tente

L'application de tente, en anglais dite tent Map ressemble à l'application logistique. Elle génère des séquences chaotiques dans  $[0, 1]$  avec  $X_0 \in [0, 1]$  prenant la forme[29] :

$$X_{k+1} = \begin{cases} X_k & X_k < 0.7 \\ 0.7 & \\ \frac{10}{3X_k(1-X_k)} & \text{ailleurs} \end{cases} \quad (1.3)$$

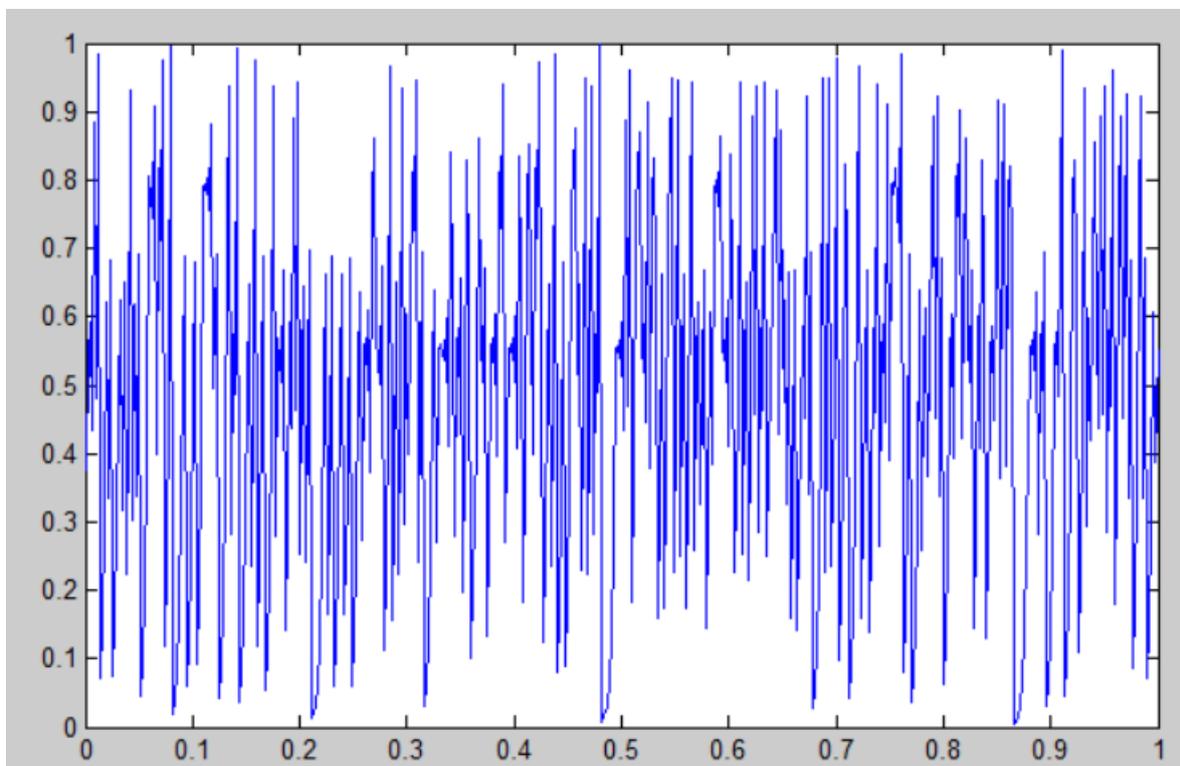


FIGURE 1.5 – L'application de Tente

### 1.5.4 Application Circulaire

L'application Cercle (en anglais circl map) est représentée par[48] :

$$X_{k+1} = X_k + b - \left(\frac{a}{2\pi}\right) \sin(2\pi X_k) \text{mod}(1) \quad (1.4)$$

Avec  $X_0 \in [0, 1]$ , elle génère une suite chaotique dans  $[0, 1]$ .

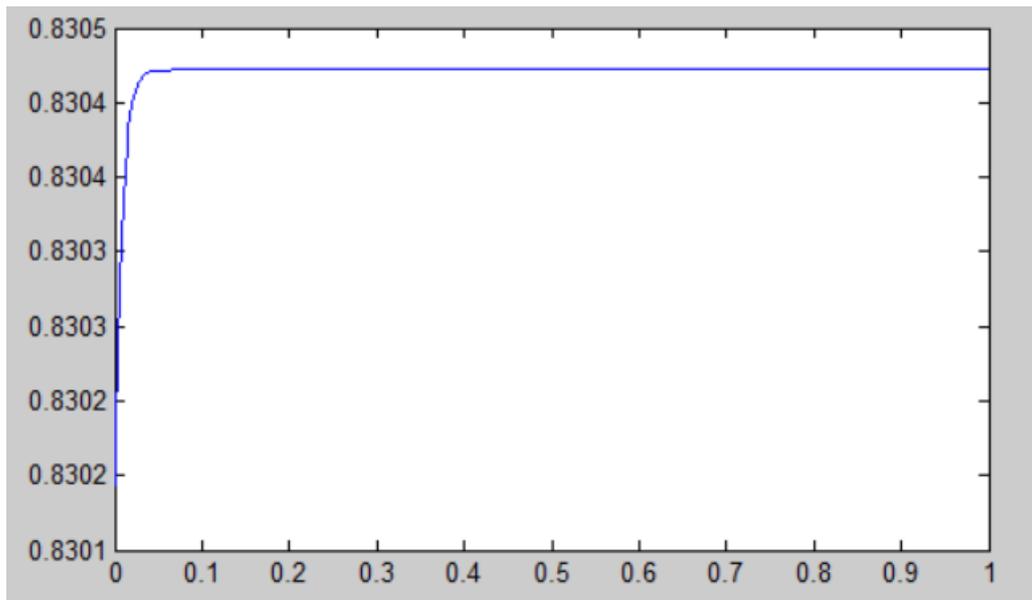


FIGURE 1.6 – L'application Cercle pour  $a = 0.7$  et  $b = 0.1$

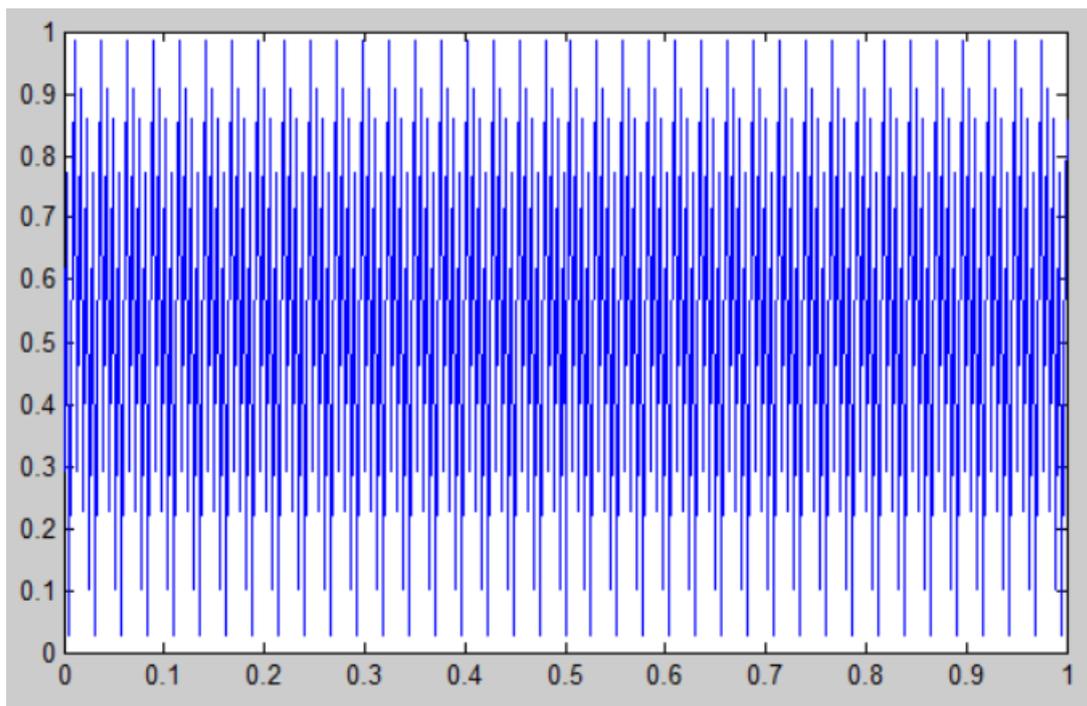


FIGURE 1.7 – L'application Cercle pour  $a = 0.5$  et  $b = 0.2$

L'application Cercle est chaotique pour  $a = 0.5$  et  $b = 0.2$ .

## 1.6 Quelques applications de la théorie du chaos

Les domaines d'applications du chaos[14] sont très nombreux, on peut citer les domaines suivants :

### 1.6.1 Médecine

En médecine nous appliquons le chaos sur la cardiologie, analyse du rythme du coeur, prédiction et contrôle d'activité irrégulière du coeur. Par exemple :

#### Détection du cancer du sein

Une méthode de détection du cancer du sein fondée sur les structures fractales a récemment été proposée. Traditionnellement, le diagnostic du cancer du sein est basé sur des évaluations morphologiques des cellules, telles que la taille, la forme ou la régularité. Mais la part de subjectivité de ces critères rend difficile un diagnostic objectif. Andrew Einstein de l'université de Colombie et ses collaborateurs ont développé une méthode de diagnostic plus objective, qui exploite les propriétés fractales de ces aspects morphologiques des cellules du sein. L'irrégularité du noyau par exemple, qui est observée de manière subjective par le pathologiste, a pu être quantifiée à l'aide de deux dimensions fractales. L'équipe d'Einstein a réalisé une étude sur un groupe de 41 femmes. Le diagnostic basé sur l'approche fractale s'est avéré correct dans 95 pour-cent des cas. Cette étude ouvre la voie à des essais à plus grande échelle, qui pourraient sans doute révolutionner la détection de cancers.

### 1.6.2 Biologie

#### Évolution d'une population

L'évolution d'une population est soumise à de nombreux facteurs, comme l'introduction d'un prédateur ou une augmentation des ressources alimentaires. Les biologistes ont ainsi commencé à créer des modèles mathématiques simples pour étudier l'évolution des populations.

Le modèle le plus simple est le schéma malthusien : la population est soumise à une croissance infinie, soumise à aucune limite.

$$\text{population nouvelle} = \text{facteur de croissance} * \text{population ancienne}$$

Ainsi, si la population initiale est de 1000 et que le taux de croissance est de 10 pour-cent par an, la population sera de 1100 l'année suivante. Mais ce modèle n'est pas réaliste, car il ne tient pas compte des contraintes qui freinent la croissance : famines, maladies, . . .

Ils incorporèrent donc dans le modèle un facteur de freinage :

$$\text{Population nouvelle} = \text{facteur de croissance} * \text{population ancienne} * (1 - \text{population ancienne})$$

Cette équation est appelée équation logistique. Elle est aujourd'hui largement utilisée

et permet des prédictions très réalistes. Contrairement à la première équation, celle-ci est chaotique et donne naissance à des structures très complexes.

L'équation logistique a été mise en pratique par des entomologistes pour calculer l'effet des insectes sur les vergers, par des généticiens pour mesurer la variation d'apparition de certains gènes au sein d'une population.

### **1.6.3 Ingénierie**

Nous appliquons le chaos sur le contrôle de vibrations, stabilisation des circuits, réactions chimiques, turbines, étages de puissances, lasers, combustion, etc

### **1.6.4 Informatique**

Nous utilisons le chaos sur cryptage, contrôle du chaos dans les systèmes robotiques, etc

### **1.6.5 Communication**

Par exemple la compression et stockage d'images conception et management des réseaux d'or.

### **1.6.6 Management et finance**

Nous pouvons appliquer le chaos sur les prévisions économiques, analyse financière, et prévision du marché.

## CHAPITRE 2

# MÉTHODE D'OPTIMISATION CHAOTIQUE

### Introduction

L'optimisation[40] est l'une des branches des mathématiques appliquées modernes, et de nombreuses recherches, à la fois pratiques et théoriques, lui sont consacrées. dans lequel on définit une fonction cout, que l'on cherche à minimiser par rapport à ses paramètres. Par exemple, dans le célèbre problème du voyageur de commerce, on cherche à minimiser la longueur de la tournée d'un "voyageur de commerce", qui doit visiter un certain nombre de villes, avant de retourner à la ville de départ. La définition du problème d'optimisation est souvent complétée par la donnée de contraintes : tous les paramètres (ou variables de décision) de la solution proposée doivent respecter ces contraintes, faute de quoi la solution n'est pas réalisable.

L'optimisation consiste à rechercher le minimum ou le maximum d'une fonction avec ou sans contraintes. L'optimisation possède ses racines au 18ième siècle dans les travaux de : -Taylor, Newton , Lagrange, qui ont élaboré les bases des développements limités.

- En 1847, Cauchy fut le premier à mettre en œuvre une méthode d'optimisation, méthode du pas de descente, pour la résolution de problèmes sans contraintes. Il n'existe pas de méthode unique permettant de résoudre efficacement tous les problèmes d'optimisation. C'est pourquoi de nombreuses méthodes d'optimisation ont été développées pour résoudre différents types de problèmes d'optimisation.

Dans ce chapitre, nous avons donné des définitions sur l'optimisation globale. Ensuite, nous avons présenté quelques méthodes d'optimisation globale.

## 2.1 Définitions

**Définition 2.1** Un problème d'optimisation est usuellement formulé comme un problème de minimisation et s'écrit sous la forme[40] :

$$\left\{ \begin{array}{l} \min_x f(x), \\ \text{tel que,} \\ h_i(x) \leq 0, i = 1, \dots, m, \\ g_j(x) = 0, j = 1, \dots, p, \\ x \in S, \\ m, p \in \mathbb{N}, \end{array} \right. \quad (2.1)$$

où :  $f$  est la fonction (scalaire) à minimiser, appelée " fonction coût " ou " fonction objectif ",  $x$  représente le vecteur des variables d'optimisation,  $h_i$  sont les contraintes d'inégalité et  $g_j$  les contraintes d'égalité,  $S$  est l'espace des variables (appelé aussi espace de recherche).  $S$  indique le type de variables considérées : réelles, entières, mixtes (réelles et entières dans un même problème), discrètes, continues, bornées, etc.

**Définition 2.2** [44]  $x^*$  est un minimum global de  $f$  si et seulement si  $f(x^*) \leq f(x); \forall x \in S$ .  $x^*$  est un minimum local de  $f$  si et seulement si  $f(x^*) \leq f(x); \forall x \in S / \|x - x^*\| \leq \varepsilon, \varepsilon > 0$ .

**Définition 2.3** Une fonction multimodale [44] présente plusieurs minimaux (locaux et globaux), et une fonction unimodale n'a qu'un minimum, le minimum global.

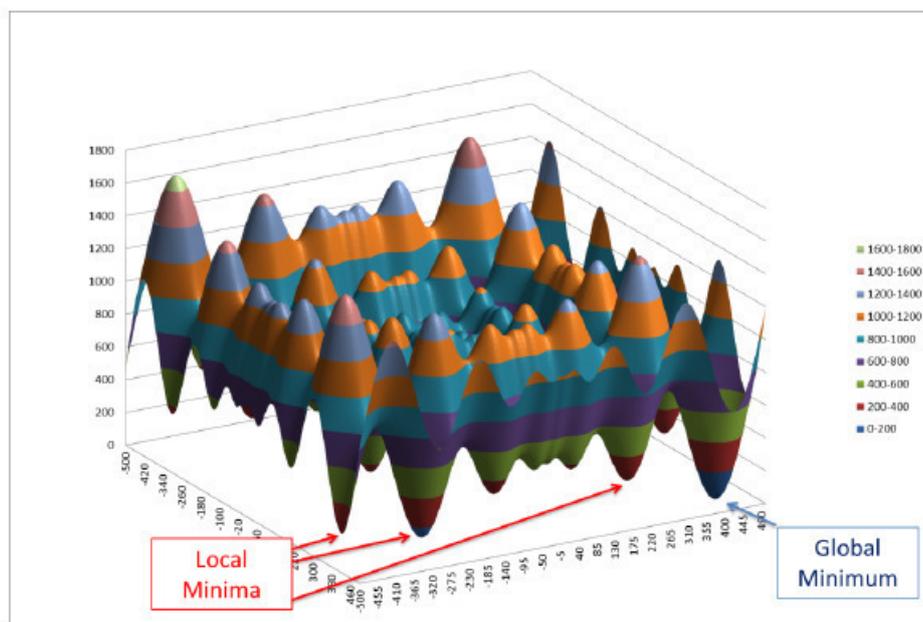


FIGURE 2.1 – Exemple d'une fonction multimodale

**Définition 2.4** On appelle "méthode locale" [44] celle qui converge vers un minimum local. Les recherches locales partent usuellement d'un point initial  $x_0$  avec un pas initial  $\rho_0$ .

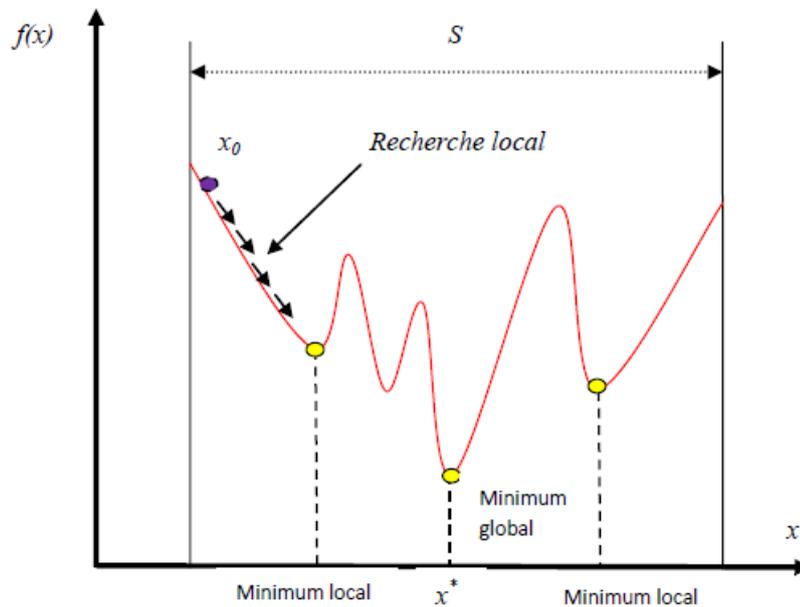


FIGURE 2.2 – Minima locaux et minima globaux d'une fonction multimodale

## 2.2 Méthodes d'optimisation globale

On trouve dans la littérature, différentes classifications des méthodes globales. Dans [32] une classification à été introduite. On présente dans cette section les approches d'optimisation globale.

### 2.2.1 Approches déterministes et approches probabilistes

#### Approches déterministes

Dans ce type de méthodes [9], l'aléatoire n'intervient pas, c'est-à-dire que pour résoudre un problème, l'algorithme se comportera toujours de la même façon et donnera toujours la même réponse. Ces algorithmes peuvent se classer en fonction du type de problèmes pouvant être résolu les programmes linéaires, les problèmes convexes, quadratiques, polynomiaux ou plus généraux.

Ces techniques ont généralement l'avantage de ne pas nécessiter de point de départ. Mais avant tout elles fournissent une réponse déterminante sur la qualité des solutions trouvées : l'optimum est-il local ou global? Quel est le degré de certitude? etc. Cette

précision a une importance significative, car il est souvent beaucoup moins coûteux de trouver une solution que de prouver qu'il s'agit bien de l'optimum global.

Pour ces méthodes, l'exploration de l'espace des solutions se fait grâce à des procédures de recherche qui sont élaborées à partir de la constante de Lipschitz ou celle de Holder, des dérivées ou d'autres informations locales et globales concernant la fonction objectif. Dans le cas où la fonction objectif est donnée sous forme de "boîte noire" le problème d'optimisation est particulièrement difficile. Les modèles déterministes ne traitent pas adéquatement les informations disponibles sur la fonction objectif.

## Approches stochastiques

Les approches stochastiques [9] peuvent souvent faire face à ce genre de problèmes plus facilement et plus efficacement que les algorithmes déterministes. Les algorithmes stochastiques explorent l'espace des solutions grâce en partie à des procédures de transitions aléatoires. Ainsi, plusieurs exécutions successives de ces algorithmes, pourront conduire à des résultats différents (pour un même point initial). L'avantage des méthodes stochastiques est leur simplicité et leur pertinence pour les problèmes où les évaluations de la fonction objectif sont corrompues par un bruit aléatoire, ainsi que leur robustesse à l'égard de la croissance de la dimension. L'inconvénient majeur de ces méthodes est qu'elles peuvent diverger, et passer plusieurs fois à côté de la solution. L'obtention de l'optimum global n'est pas garantie, il est seulement repéré avec une probabilité proche de 1.

De nombreux algorithmes, où l'aléatoire et l'argument statistique sont impliqués ont été proposés heuristiquement. Ces algorithmes sont basés sur des analogies avec les processus naturels. Les exemples les plus connus de tels algorithmes sont l'optimisation évolutive, le recuit simulé, l'algorithme génétique et la recherche tabou.

### 2.2.2 Méthodes de descente généralisées

Dans ces méthodes [44] un algorithme d'optimisation local est employé itérativement, et la direction de descente est modifiée de manière à éviter la convergence vers un optimum local déjà trouvé. Une façon de réaliser ceci est de pénaliser la fonction  $f$  au fur et à mesure que l'optimiseur s'approche d'un minimum local  $x_i^*$  déjà trouvé. Par exemple,

$$f_p = f + P(x_1^*, x_2^*, \dots, x_c^*)$$

où

$$P = \sum_{i=1}^c \frac{K_i}{|x - x_i^*|^\alpha}$$

et  $c$  est le nombre de minima locaux trouvés. Un désavantage de la pénalisation directe de  $f$  est que l'on peut introduire des optima locaux dans  $f_p$  qui n'existent pas dans  $f$ .

Une autre manière d'explorer d'autres régions après avoir trouvé des minima locaux est d'utiliser les méthodes de tunnel [23]. Elles cherchent des points de départ dans des vallées différentes. Cette méthode présente deux phases qui sont exécutées en alternance.

Dans la première phase, une recherche locale est effectuée sur la fonction  $f$ , à partir d'un point de départ  $x_0^i$ . Elle convergera vers un minimum (local)  $x_i^*$ .

Ensuite, la phase de "tunnel" se déroule. Un nouveau point de départ  $x_0^{i+1}$  est cherché dans une autre vallée, à l'aide d'une fonction tunnel  $T$ .  $x_0^{i+1}$  doit vérifier la condition  $f(x_0^{i+1}) < f(x_i^*)$  pour garantir qu'il se situera dans une vallée plus basse, donc différente des précédentes.  $x_0^{i+1}$  doit satisfaire  $T(x_0^{i+1}) < 0$ , la fonction tunnel étant, par exemple,

$$T(x) = \frac{f(x) - \min_{i=1}^c f(x_i^*)}{\prod_{i=1}^c \|x - x_i^*\|^{2\alpha_i}}$$

Où  $\alpha_i$  est un paramètre qui contrôle "la force" de répulsion autour de  $x_i^*$ .

Ces méthodes souffrent typiquement de mauvais conditionnements numériques liés à  $P(x)$  et  $T(x)$ .

### 2.2.3 Recherches aléatoires

Dans ce groupe de méthodes[44] on peut avoir des recherches aléatoires pures, qui consistent à tirer un point, au hasard, à chaque itération. La fonction coût est évaluée en ce point, et s'il y a une amélioration, ce point et la fonction correspondante sont enregistrés, et le processus continue. Les recherches aléatoires peuvent aussi être associées aux recherches locales. Ainsi des points au hasard sont pris pour ré-initialiser des recherches locales. Ces ré-initialisations sont susceptibles de converger plusieurs fois vers les mêmes minima locaux. De plus, il n'y a pas de discrimination entre régions prometteuses ou non prometteuses[20].

### 2.2.4 Méthode de multi-départ

La méthode multi-départ[42] diffère simplement de la recherche aléatoire par le fait qu'elle applique une procédure de recherche locale au point initial avant de tester

si son image est meilleure. Elle présente donc l'avantage de traiter le point généré, mais par contre le point obtenu sera souvent le même. En effet, l'inconvénient de Multi-départ est que le même minimum local peut s'identifier plusieurs fois menant à une recherche globale inefficace.

### 2.2.5 Méthodes de regroupement

Les méthodes de groupements[44] peuvent être considérées comme une forme modifiée de l'algorithme standard de multi-départ. Ces méthodes essaient d'éviter l'inefficacité introduite par la méthode de multi-départ en choisissant soigneusement les points auxquels la recherche locale est lancée. Les trois étapes principales des méthodes de groupements sont :

- Échantillonnage du domaine de recherche.
- Transformation des points pour les grouper autour des minima locaux.
- Application d'une technique groupante pour identifier les groupes que représentent les voisinages des minima locaux.

Bien évidemment, la difficulté consiste à regrouper judicieusement les points de manière à ce que tous les points ayant le même groupe. Et si ce procédé identifie avec succès ces groupes ; qui représentent les voisinages des minima locaux ; alors un nombre important des recherches locales peuvent être évité en commençant simplement une recherche locale d'un certain point dans chaque groupe[3, 31]. Deux méthodes existent pour exécuter cette opération de groupement : le groupement de densité et la liaison simple, détaillées dans[34, 43]. Un désavantage de ces méthodes est qu'elles sont peu performantes pour les fonctions ayant de nombreux minima. Dans ces cas, en effet, un grand nombre d'échantillons est nécessaire.

### 2.2.6 Méthodes évolutionnaires

Les méthodes évolutionnaires [44] sont des méthodes d'optimisation probabilistes qui sont souvent présentées par analogie avec la théorie de la sélection naturelle de Darwin. Les individus les plus performants d'une population ont une plus grande probabilité de survivre et de se reproduire, en donnant des descendants encore mieux adaptés. La correspondance avec la minimisation d'une fonction est la suivante : un " individu " est un point  $x$ , la " population " est un ensemble de points, et " la performance d'un individu "  $x$  est évaluée par sa valeur de fonction coût  $f(x)$ . En partant d'une échantillonnage de l'espace de recherche (une population initiale), ces méthodes réalisent des tirages de nouvelles populations (des générations) à l'aide d'opérations de sélection, croisement et mutation[15, 28].

Les différents algorithmes évolutionnaires se distinguent par la mise en oeuvre de ces opérations. Historiquement, ils ont été développés indépendamment en trois variantes[2,

19, 36] : algorithmes génétiques, stratégies d'évolution et programmation évolutive.

- Un exemple de structure des procédures évolutionnaire est le suivant :
  - initialisation de la population  $P(t)$ ,  $t = 0$  (au hasard, par exemple);
  - évaluation de la population courante  $P(t)$  : calcul de la fonction coût pour tous les points de la population;
  - sélection de certains individus de  $P(t)$  en fonction de leur fonction coût  $f$  par exemple les meilleurs : les parents  $P_0(t)$ ,
  - croisement entre parents et création des "enfants"  $\Rightarrow P_1(t)$ ,
  - Mutation perturbation aléatoire des individus de  $P_1(t) \Rightarrow P_2(t)$ ,
  - Évaluation de  $P_2(t)$ ,
  - Formation de la nouvelle population  $P(t+1)$  à partir des individus de  $P(t)$  et de  $P_2(t)$ , par exemple en choisissant les meilleurs individus appartenant à  $P(t)$  et à  $P_2(t)$ ;
  - $t \leftarrow t + 1$ ;
  - Si le critère d'arrêt n'est pas satisfait, retour en 3.

Une quatrième famille de méthodes évolutionnaires, plus distante de la métaphore biologique et au formalisme plus rigoureux, est la famille des algorithmes d'optimisation statistique. A partir des individus sélectionnés (étape 3), une densité de probabilité de présence de l'optimum est estimée. Cette densité servira à générer des nouveaux individus (étapes 4 et 5). Ces méthodes peuvent être considérées comme des algorithmes évolutionnaires sans opérateurs.

Les algorithmes évolutionnaires permettent de réaliser des recherches globales, formellement grâce à l'étape 5. Leurs intérêts pratiques réside dans leurs applicabilité à une très large gamme de problèmes car ils n'utilisent que la fonction coût (pas de dérivées) et peuvent accommoder des variables mixtes. Cependant, pour trouver un optimum, ils ont besoin d'un grand nombre d'évaluations de la fonction coût. Ainsi, ils ne sont considérés directement utilisables que pour les problèmes où le calcul de la fonction ne nécessite pas de ressources importantes.

### 2.2.7 Généralités sur les méthodes d'optimisation globale

De manière synthétique, on notera que les méthodes d'optimisation globales présentent toujours deux composantes. Une composante globale ou exploratrice, qui permet d'explorer l'espace de recherche, et une composante locale ou exploitatrice, qui vise à exploiter les informations (relation entre  $x$  et  $f$ ) recueillies. Les méthodes d'optimisation globales sont, souvent, stochastiques, à travers leur composante exploratrice. Ainsi on ne peut pas les juger à partir d'une seule exécution. Pour les qualifier, il est nécessaire d'effectuer une analyse statistique (moyenne, écart-type, etc.) de plusieurs exécutions.

## 2.3 Les Algorithmes de recherche chaotiques

On a traité dans la section précédente une classe des approches probabilistes basées sur l'observation. Parmi ces méthodes, l'algorithme de recherche aléatoire est probablement le plus robuste, le plus simple et n'est pas limité par le dimensionnement de l'espace de recherche, ce pendant il est probablement le moins efficace. Les algorithmes de recherche aléatoire peuvent être utilisés d'une manière intelligente, de manière à ce que leur efficacité puisse être améliorée et leurs avantages restent valables en même temps. L'idée principale est de construire des algorithmes de recherche chaotique qui explorent l'espace de recherche d'une manière chaotique. Les algorithmes d'optimisation chaotiques comme une méthode émergente de l'optimisation globale ont attiré beaucoup d'attention dans les applications d'ingénierie. Leurs bonnes performances ont été soulignées. Dans le cadre d'algorithmes évolutionnaires, l'utilisation de séquences chaotiques au lieu de celles aléatoires a été introduite par Caponetto et al.[6].

Depuis leur travail original, la littérature sur l'optimisation chaotique est florissante. Ils sont utilisés dans le cadre de la méthode de réglage pour déterminer les paramètres du PID d'un régulateur de tension automatique, ou également pour l'optimisation de la conception technique et dans beaucoup d'autres problèmes physiques, économiques et biologiques.

### 2.3.1 La Recherche chaotique de l'optimum

La méthode de la recherche chaotique est une méthode non affectée par la dimension du problème d'optimisation. Cette méthode est basée sur l'exploration de l'espace de recherche  $S$  d'une manière chaotique afin de trouver un point qui minimise la fonction objective  $f$ . Cette méthode consiste à sélectionner à chaque itération une solution au hasard. La fonction objective  $f$  est évaluée en ce point. La nouvelle valeur est comparée à la précédente, si elle est meilleure que la précédente, cette valeur est enregistrée, ainsi que la solution correspondante, et le processus continue. Sinon on repart du point précédent et on recommence le procédé, jusqu'à ce que les conditions d'arrêt soient atteintes.

La figure suivante montre une comparaison entre la méthode de recherche aléatoire dont les variables de la fonction objectif sont générées selon une loi uniforme sur l'espace de recherche avec la méthode de recherche chaotique, ici la variable de la fonction objectif sont générées chaotiquement selon un modèle chaotique (ou une application chaotique) sur l'espace de recherche  $S$ .

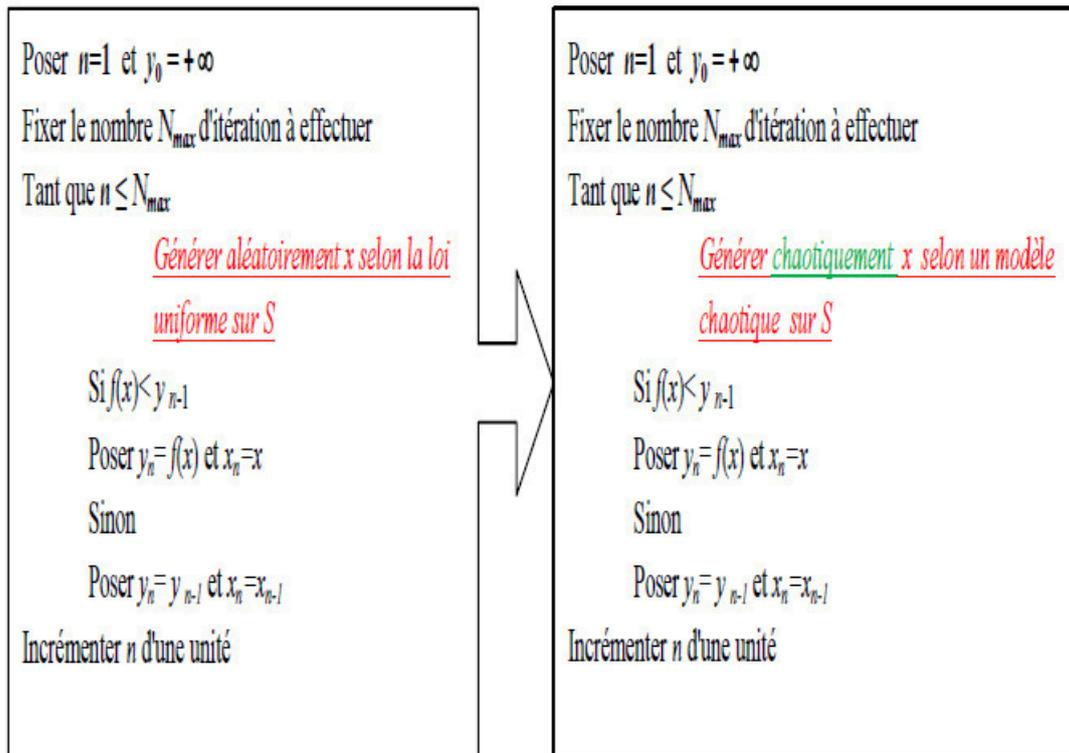


FIGURE 2.3 – Comparaison entre la recherche aléatoire et recherche chaotique (où  $x^* = x_n, f^* = y_n$ )

## CHAPITRE 3

# ALGORITHME DE RECHERCHE CHAOTIQUE ET RÉSULTATS NUMÉRIQUES

### Introduction

L'identification de l'optimum global d'une fonction multimodale reste encore une tâche très difficile, malgré la puissance de la technologie de l'information d'aujourd'hui et les multitudes de techniques de recherche qui existent.

Le chaos est désormais un concept bien établi et il y a une vaste littérature sur la nature du chaos. Depuis plus de trois décennies, le comportement inhabituel des systèmes chaotiques a attiré l'attention de plusieurs différentes communautés scientifiques. Les comportements chaotiques ont été observés dans différents domaines des sciences, par exemple l'ingénierie, la médecine, la biologie et l'économie.

Le chaos est mathématiquement défini comme un modèle aléatoire généré par des systèmes déterministes simples. En général, le chaos a trois importantes propriétés[16, 41, 45, 46] :

- la sensibilité aux conditions initiales.
- la propriété de semi-stochastique.
- la propriété quasi-aléatoire.

L'utilisation des systèmes chaotiques au lieu des processus aléatoires a été remarqué dans plusieurs domaines. Un de ces champs est la théorie de l'optimisation.

Dans les algorithmes d'optimisation aléatoire le rôle du hasard peut être joué par une dynamique chaotique.

Des études expérimentales ont affirmé que les avantages de l'utilisation des modèles chaotiques au lieu des modèles aléatoires sont souvent évidents mais il n'est pas encore mathématiquement prouvé[4, 6].

A partir de ces propriétés du chaos, certains nouveaux algorithmes de recherche

appelés algorithmes d'optimisation chaotique (COAs) sont présentés dans la littérature[5, 7, 25].

Les méthodes (COA) peuvent facilement échapper à des minima locaux contrairement aux algorithmes d'optimisation stochastiques.

L'algorithme d'optimisation stochastique s'échappe souvent à des minima locaux en acceptant quelques mauvaises solutions selon une certaine probabilité. Mais les COAs s'échappent à des minima locaux à cause de la régularité des mouvements chaotiques.

On présente dans ce chapitre l'algorithme d'optimisation chaotique. Puis on définit trois fonctions tests. Ensuite, on applique cet algorithme sur ces fonctions, en employant quatre chaos map. Nous présentons également les résultats numériques obtenus.

### 3.1 Principe et étapes de l'algorithme d'optimisation chaotique

Beaucoup de problèmes d'optimisation sans contraintes avec des variables continues peuvent être formulés comme des problèmes d'optimisation de la manière suivante :

$$\left\{ \begin{array}{l} \min f(X), \\ \text{telque,} \\ X = [x_1, x_2, \dots, x_n] \\ x_i \in [L_i, U_i], i = 1, 2, \dots, n, \end{array} \right. \quad (3.1)$$

où :  $f$  est appelée fonction objectif,  $X$  représente le vecteur des variables d'optimisation

La procédure de recherche chaotique basée sur certain modèle chaotique peut être illustrée comme suit[8, 37, 38] :

**Données :**

$M_g$  : le nombre maximal d'itérations de la recherche globale chaotiques.

$M_l$  : le nombre maximum d'itérations de la recherche locale chaotiques.

$M_l + M_g$  : le critère d'arrêt de la méthode d'optimisation chaotique.

$\lambda$  : le pas de la recherche locale chaotique.

**Sorties :**

$\bar{X}$  : la meilleure solution d'exécution en cours de recherche chaotique.

$\bar{f}$  : la meilleure valeur de la fonction objectif.

Le processus de base de COA comporte généralement deux phases principales ;

De plus, définir un générateur de séquences chaotiques basées sur une application chaotique.

puis on génère un point candidat  $x_c$  dans la région de recherche  $[L, U]$ , l'optimum  $\bar{x}$ , est le point candidat ayant obtenu la plus petite valeur de  $f(x_c)$ .

Le processus est schématisé par l'algorithme suivant, les points  $x_c$  sont générés dans le domaine  $[L, U]$  au moyen du vecteur chaotique  $z$ .

Dans cet algorithme, nous supposons que les composantes de  $z$  sont restreintes à l'intervalle  $[0, 1]$ .

A chaque itération, un nouveau vecteur de séquences chaotiques est généré en utilisant le modèle chaotique.

---



---

Algorithm : COA

---

**Begin**

**-Step 0 :** Fixed the number  $Mg, Ml$  of chaotic search. Initialization of variables and initial conditions Set  $k = 1, y(0)$  of chao map. Set the initial best objective function  $\bar{f} = +\infty$

**-Step 1 : algorithm of chaotic global search :**

**while**  $k \leq Mg$  **do**

$$x_i(k) = L_i + z_i(k) \cdot (U_i - L_i)$$

**if**  $f(X(k)) < \bar{f}$  **then**

$$\bar{X} = X(k); \bar{f} = f(X(k))$$

**end if**

$$k = k + 1$$

**end while**

**-Step 2 : algorithm of chaotic local search :**

**while**  $k \leq (Mg + Ml)$  **do**

**if**  $r \leq 0.5$  **then** ( $r$  is a uniformly distributed random variable within range  $[0, 1]$ )

$$x_i(k) = \bar{x}_i + \lambda z_i(k) \cdot |U_i - \bar{x}_i|$$

**else**

$$x_i(k) = \bar{x}_i - \lambda z_i(k) \cdot |\bar{x}_i - L_i|$$

**end if**

**if**  $f(X(k)) < \bar{f}$  **then**

$$\bar{X} = X(k); \bar{f} = f(X(k))$$

**end if**

$$k = k + 1$$

**end while**

**End.**

---

On obtient un nouvel optimum local  $\bar{x}$ .

La première étape de cet algorithme est utilisée pour obtenir un bon optimum pour l'étape de raffinement où la recherche chaotique locale.  $z$  est toujours le modèle chaotique et  $\lambda$  est un paramètre scalaire lié au rayon de la recherche autour de  $\bar{X}$ . Les optima locaux sont mis à jour chaque fois qu'un point est trouvé de mieux telle que la procédure se poursuit pour la recherche au voisinage du nouveau point optimal. L'étape deux de cet algorithme de la recherche chaotique globale est similaire à la technique d'optimisation de Monte-Carlo qui converge lentement et obtient l'optimum global avec une faible probabilité.

## 3.2 Simulation numériques

L'algorithme vise à améliorer la convergence de l'optimisation chaotique. En mathématiques appliquées, les fonctions de test [39, 47] sont utiles pour évaluer les caractéristiques des algorithmes d'optimisation. Pour tester l'efficacité de cet algorithme, nous l'avons appliqué sur trois différentes fonctions tests (La fonction Griewank, La fonction Rastrigin et La fonction Sphère) :

D'autre part, nous avons étudié l'efficacité de quelques modèles en tant que générateurs de variables chaotiques dans cet algorithme. Différents types d'applications chaotiques ont été utilisées, l'application logistique, l'application Tente, l'application de Lozi et l'application cercle.

### 3.2.1 Les fonctions de tests

#### 3.2.1.1 La fonction de Rastrigin

La fonction de Rastrigin [30] est une fonction mathématique souvent utilisée pour évaluer la performance des algorithmes d'optimisation. Elle présente des pièges intéressants, sous la forme de ses nombreux minima et maxima locaux. Elle a été proposée, en 1974, par Rastrigin en deux dimensions et a été généralisée par Muhlenbein et D. Schomisch et J. Born.

Sa définition, en dimension  $n$ , est définie par :

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (3.2)$$

Son minimum global se trouve à l'origine, où sa valeur est nulle. Dans le domaine de recherche  $-5.12 \leq x_i \leq 5.12$ .

La figure suivant donne la représentation graphique de la fonction Rastrigin :

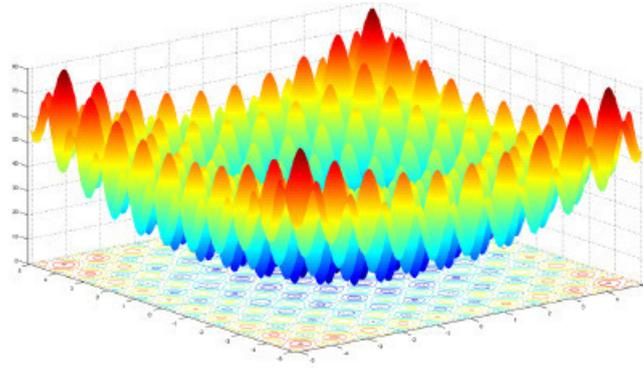


FIGURE 3.1 – Fonction de Rastrigin

### 3.2.1.2 La fonction de Griewank

La fonction Griewank[13] est une fonction largement utilisée pour tester la convergence des fonctions d’optimisation. sa formule est donnée par :

$$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (3.3)$$

Son minimum global est  $X^* = (0, 0, \dots, 0)$  et  $f(X^*) = 0$ . Où  $-500 \leq x_i \leq 500$ .

La fonction de Griewank est donnée par la figure suivante :

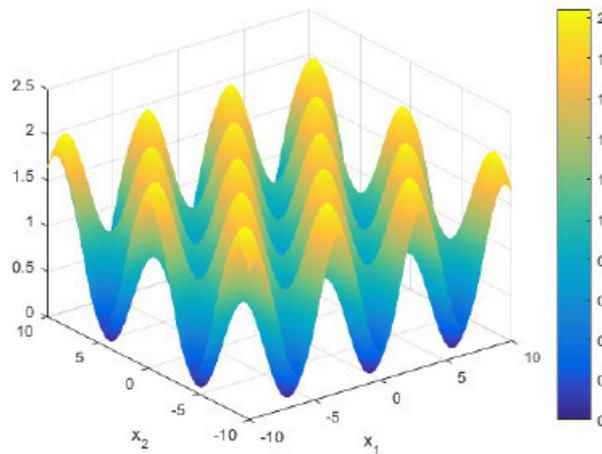


FIGURE 3.2 – Fonction de Griewank

### 3.2.1.3 La fonction de Sphère

Sa formule est donnée par[11] :

$$f(x) = \sqrt{\sum_{i=1}^n x_i^2} \quad (3.4)$$

Son minimum global est  $X^* = (0, 0, \dots, 0)$  et  $f(X^*) = 0$ . Avec  $-100 \leq x_i \leq 100$ .

### 3.2.2 Résultats numériques

Toutes les fonctions tests sont résolues par l'algorithme d'optimisation chaotique (COA) dont les variables de ces fonctions sont générées par les applications chaotiques : (Circle map, Tente map, Lozi map, Logistic map). Elles sont normalisées entre 0 et 1. Nous comparons les résultats obtenus par l'algorithme (COA) avec la solution exacte. Dans chaque étude de cas, 200 itérations ont été effectuées de l'algorithme (COA) afin de trouver l'optimum globale.

Les résultats statistiques tels que la meilleure valeur  $f^*$ , valeur moyenne  $\bar{f}$ , écart-type  $\sigma_f$  des exécutions obtenues par chaque algorithme pour les fonctions Rastrigin, Griewank et Sphère sont respectivement présentées dans les tableaux 3.1, 3.2, 3.3, 3.4, 3.5 et 3.6 pour chercher la fonction chaotique la plus efficace.

#### 3.2.2.1 Résultats pour la fonction de Rastrigin

Pour  $\lambda = 0.1$ ,  $Mg = 10000$ ,  $Ml = 500$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	4.9200e-005	5.0032e-004	4.9693e-004	-0.0012 0.0066 -0.0014
Logistic map	1.8365e-005	6.3713e-005	4.0924e-005	-0.0018 0.0023 -0.0029
Circle map	4.0882e-004	0.0161	0.1999	0.0113 0.0113 0.0113
Tent map	8.4794e-006	5.9921e-005	1.5985e-004	-0.0023 0.0002 0.0016

TABLE 3.1 – L'optimum de la fonction de Rastrigin pour  $n = 3$  avec différents chao map.

Pour  $\lambda = 0.1$ ,  $Mg = 20000$ ,  $Ml = 1000$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	3.0325e-004	0.0522	0.7294	0.0022 0.0022 -0.0042 -0.0161
Logistic map	3.2210e-004	7.7733e-004	8.3055e-004	0.0139 -0.0040 -0.0037 0.0089
Circle map	0.1756	0.3871	0.3786	0.2533 -0.3144 0.0506 -0.0067
Tent map	1.8189e-004	4.0471e-004	5.5324e-004	0.0073 -0.0045 -0.0007 0.0099

TABLE 3.2 – L’optimum de la fonction de Rastrigin pour  $n = 4$  avec différents chao map.

Les tableaux 3.1 et 3.2 représentent respectivement les résultats obtenus par l’algorithme COA pour différentes applications chaotiques de la fonction Rastrigin en dimension 3 et 4. On remarque que les applications Tente et Logistique sont plus efficace que les autres applications chaotiques.

### 3.2.2.2 Résultats pour la fonction de Griewank

Pour  $\lambda = 0.1$ ,  $Mg = 10000$ ,  $Ml = 500$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	1.9810e-004	3.2816e-004	9.3659e-004	-0.3135 0.2128 0.6914
Logistic map	7.3617e-005	1.4537e-004	2.0490e-004	0.3096 -0.0291 -0.3400
Circle map	2.4572e-004	0.0935	0.4130	-0.5208 0.3385 0.5744
Tent map	4.4648e-005	5.9023e-005	1.0632e-004	0.0706 -0.0650 -0.3704

TABLE 3.3 – L’optimum de la fonction de Griewank pour  $n = 3$  avec différents chao map.

Pour  $\lambda = 0.1$ ,  $Mg = 20000$ ,  $Ml = 1000$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	5.8603e-004	0.0010	0.0018	-1.1773 0.1670 0.0026 -0.2590
Logistic map	5.7724e-004	0.0023	0.0027	-0.3602 0.7288 -0.4352 1.0117
Circle map	0.0016	0.2957	0.9667	1.1072 1.1072 1.1072 1.1072
Tent map	6.6562e-004	0.0011	0.0021	-0.9119 -0.4906 -0.1955 0.9149

TABLE 3.4 – L’optimum de la fonction de Griewank pour  $n = 4$  avec différents chao map.

Nous présentons respectivement dans les tableaux 3.3 et 3.4 les résultats obtenus par l’algorithme COA en utilisant quatre applications chaotiques pour la recherche du minimum globale de la fonction Griewank en dimension 3 et 4. Nous avons remarqué que les applications chaotiques Tente et Logistique donnent des meilleurs résultats par rapport aux autres applications.

### 3.2.2.3 Résultats pour la fonction de Sphère

Pour  $\lambda = 0.1$ ,  $Mg = 10000$ ,  $Ml = 500$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	0.0140	0.0192	0.0315	0.0038 -0.0135
Logistic map	0.0022	0.0083	0.0138	-0.0022 0.0004
Circle map	0.0975	0.4878	1.1911	0.0089 -0.0971
Tent map	0.0029	0.0086	0.0168	-0.0011 -0.0027

TABLE 3.5 – L’optimum de la fonction de Sphère pour  $n = 2$  avec différents chao map.

Pour  $\lambda = 0.1$ ,  $Mg = 10000$ ,  $Ml = 500$

Chao map	Meilleure valeur $f^*$	Valeur moyenne $\bar{f}$	Écart type $\sigma_f$	$X^*$
Lozi map	0.0629	0.1559	0.1538	-0.0623 0.0024 0.0083
Logistic map	0.0617	0.1007	0.1113	-0.0537 0.0259 -0.0156
Circle map	0.3835	0.9204	2.2223	0.2214 0.2214 0.2214
Tent map	0.0346	0.1335	0.0891	0.0287 0.0182 0.0064

TABLE 3.6 – L’optimum de la fonction de Sphère pour  $n = 3$  avec différents chao map.

Dans les tableaux 3.5 et 3.6 nous donnons les résultats de l’algorithme COA dont différents applications chaotiques à la fonction Sphère en dimension 2 et 3 respectivement.

On constate que l’application Logistique donne un bon résultat de la fonction Sphère en dimension 2 par rapport aux autres applications chaotiques.

Pour la fonction Sphère en dimension 3, l’application Tente donne meilleur valeur fournit un bon résultat.

## Conclusion

Il convient de souligner que différentes applications chaotiques peuvent fonctionner différemment, mais pour une application donnée, il n’y a pas de meilleure application chaotique à recommander car il existe souvent quelques applications chaotiques qui peuvent fonctionner presque aussi bien. Il est donc pas facile de choisir une application chaotique sans expérimentation réelle.

Nous constatons que les applications chaotiques Tente et Logistique produisent de meilleur résultat, et elle sont très efficaces.

## CONCLUSION GÉNÉRALE

L'optimisation chaotique est un nouvel algorithme d'optimisation stochastique, qui utilise directement les variables chaotiques pour rechercher la solution optimale.

Dans ce mémoire nous avons utilisé un algorithme d'optimisation chaotique pour résoudre des problèmes d'optimisation sans contraintes. Dans un premier temps, une étude bibliographique a été développée afin de maîtriser et de comprendre certaines notions de base sur les fonctions d'évolutions chaotiques. Nous avons cités également quelques domaines d'applications du chaos, ainsi que les méthodes d'optimisation globale. Dans un second temps, nous avons présenté les étapes de l'algorithme d'optimisation chaotique (COA).

En Effet, nous avons établi un programme sur MATLEB qui donne une solution globale aux trois fonctions tests, dont les variables de ces fonctions sont générées chaotiquement par des applications chaotiques : Logistic map, Lozi map, Tente map et cercle map.

Une étude statistique sinificative a été faite afin de déterminer l'application chaotique qui donnera la meilleure solution. Les résultats numériques sont satisfaisants pour les fonctions tests donnent le minimum global proche de zéro.

## 1) Programme de l'algorithme d'optimisation chaotique pour la fonction de Griewank avec l'application Tente sous Matlab :

```
clc; clear all;
Mg=20000;          "nbr maximal d'itération de la recherche globale chaotique"
Ml=1000;          "nbr maximal d'itération de la recherche locale chaotique"
l=0.1;           "le pas de la recherche locale chaotique"
S=0;w=0;
n=4;             "ndr des variables de la fonction test"
m=200;          "ndr d'itérations de l'algorithme"
f=inf;
L=-500;         "la borne inférieure des variables"
U=500;         "la borne supérieure des variables"

for j=1 :m
k=1;F=inf;
for i=1 :n
T0(i)=rand;      "condition initiale de l'application Tente"
end
q=T0;
while k<=Mg
for i=1 :n
if T0(i)<0.7
T(k,i)=T0(i)./0.7;
T0(i)=T(k,i);
```

```
else                                     " L'application Tente "
T(k,i)=10/3.*T0(i).*(1-T0(i));
T0(i)=T(k,i);
end
x(k,i)=L+(T(k,i)*(U-L));
end
f1 = griewangk(k,x,n);
if f1<F
F=f1 ;
X1=x(k, :);
end
k=k+1;
end
disp ('Le minimum est :');
disp('X1*='');disp(X1)
disp('f(X1*)='');disp(F)

    k=1 ;
T0=q;
F=inf;
while k<=Mg+Ml
for i=1 :n
if T0(i)<0.7
T(k,i)=T0(i)./0.7 ;
T0(i)=T(k,i);
else
T(k,i)=10/3.*T0(i).*(1-T0(i));
T0(i)=T(k,i);
end
r=rand;
if r<=0.5
x(k,i)=X1(i)+l*T(k,i)*abs(U-X1(i));
else
x(k,i)=X1(i)-l*T(k,i)*abs(X1(i)-L);
end
end
f1 = griewangk(k,x,n);
if f1<F
F=f1 ;
```

```
X2=x(k, :);
end
k=k+1;
end
disp ('Le minimum global est :');
disp ('X2*=' );disp(X2)
disp ('f(X2*)=' ); disp(F)

    if F<f
f=F;
Xg=X2;
end
S=S+f;
w=w+f2;
end

    disp ('La Meilleure valeur de f* est :');
disp ('X*=' );disp(Xg)
disp ('f(X*)=' ); disp(f)
disp ('La valeur moyenne de f* est :');
Moy=S/m
disp ('L Écart-type de f* est :');
ET=sqrt((w/m)-(Moy)2)
```

## **2) Programme de l'algorithme d'optimisation chaotique pour la fonction de Griewank avec l'application logistique sous Matlab :**

```
clc; clear all;
Mg=20000; Ml=1000; l=0.1; S=0;w=0; n=4; m=200; f=inf;
I=-500; U=500;
for j=1 :m
k=1;F=inf;
for i=1 :n
L0(i)=rand
end
q=L0;
while k<=Mg
```

```
for i=1 :n
L(k,i)=4.*L0(i).*(1-L0(i));
L0(i)=L(k,i);
x(k,i)=I+(L(k,i)*(U-I));
end
f1 = griewangk(k,x,n);
if f1<F
F=f1 ;
X1=x(k, :);
end
k=k+1 ;
end
disp ('Le minimum est :');
disp('X1*=' );disp(X1)
disp('f(X1*)=' );disp(F)
k=1 ; L0=q; F=inf;
while k<=Mg+Ml
for i=1 :n
L(k,i)=4.*L0(i).*(1-L0(i));
L0(i)=L(k,i);
r=rand;
if r<=0.5
x(k,i)=X1(i)+l*L(k,i)*abs(U-X1(i));
else
x(k,i)=X1(i)-l*L(k,i)*abs(X1(i)-I);
end end
f1 = griewangk(k,x,n);
if f1<F
F=f1 ;
X2=x(k, :);
end
k=k+1 ;
end
disp ('Le minimum global est :');
disp ('X2*=' );disp(X2)
disp ('f(X2*)=' ); disp(F)
if F<f
f=F;
Xg=X2;
```

” L’application logistique ”

```
end
S=S+f;
w=w+f2;
end

disp ('La Meilleure valeur de f* est :');
disp ('X*=' );disp(Xg)
disp ('f(X*)=' ); disp(f)
disp ('La valeur moyenne de f* est :');
Moy=S/m
disp ('L Écart-type de f* est :');
ET=sqrt((w/m)-(Moy)2)
```

### **3) Programme de l'algorithme d'optimisation chaotique pour la fonction de Griewank avec L'application de Lozi sous Matlab :**

```
clc; clear all; Mg=20000; Ml=1000; l=0.1; S=0;w=0; n=4;m=200;f=inf;
L=-500; U=500;
for j=1 :m
k=1;F=inf;
for i=1 :n
y01(i)=rand;
y0(i)=rand;
end
q=y01; s=y0;
while k<=Mg
for i=1 :n
y1(k,i)=1-1.7.*abs(y01(i))+y0(i);
y(k,i)=0.5.*y01(i); " L'application de Lozi "
Z(k,i)=(y(k,i)-(-0.6418))/(0.6716-(-0.6418));
y01(i)=y1(k,i); y0(i)=y(k,i);
x(k,i)=L+(Z(k,i)*(U-L));
end
f1 = griewangk(k,x,n);
if f1<F
F=f1;
X1=x(k, :);
```

```
end
k=k+1;
end
disp('Le minimum est :');
disp('X1*=' );disp(X1)
disp('f(X1*)=' );disp(F)
k=1 ; y01=q; y0=s;
while k<=Mg+Ml
for i=1 :n
y1(k,i)=1-1.7.*abs(y01(i))+y0(i);
y(k,i)=0.5.*y01(i);
Z(k,i)=(y(k,i)-(-0.6418))/(0.6716-(-0.6418));
y01(i)=y1(k,i); y0(i)=y(k,i);
r=rand;
if r<=0.5
x(k,i)=X1(i)+l*L(k,i)*abs(U-X1(i));
else
x(k,i)=X1(i)-l*L(k,i)*abs(X1(i)-I);
end end
f1 = griewangk(k,x,n);
if f1<F
F=f1;
X2=x(k, :);
end
k=k+1;
end
disp('Le minimum global est :');
disp('X2*=' );disp(X2)
disp('f(X2*)=' ); disp(F)
if F<f
f=F;
Xg=X2;
end
S=S+f;
w=w+f2;end

disp('La Meilleure valeur de f* est :');
disp('X*=' );disp(Xg)
disp('f(X*)=' ); disp(f)
```

```
disp ('La valeur moyenne de f* est :');  
Moy=S/m  
disp ('L Écart-type de f* est :');  
ET=sqrt((w/m)-(Moy)2)
```

#### **4) Programme de l'algorithme d'optimisation chaotique pour la fonction de Griewank avec L'application Cercle sous Matlab :**

```
clc; clear all; Mg=20000; Ml=1000; l=0.1; S=0;w=0; n=4;m=200;f=inf;  
L=-500; U=500;  
for j=1 :m  
k=1;F=inf;  
for i=1 :n  
C0(i)=rand;  
end  
q=C0;  
while k<=Mg  
for i=1 :n  
a=(2*pi*C0(i)); b=degtorad(a);  
e=C0(i)+0.2-((0.5/2*pi)*(sin(b))); y=floor(e);  
C(k,i)= e-y; " L'application Cercle "  
C0(i)= C(k,i);  
x(k,i)=L+(C(k,i)*(U-L));  
end  
f1 = griewangk(k,x,n);  
if f1<F  
F=f1;  
X1=x(k, :);  
end  
k=k+1;  
end  
disp ('Le minimum est :');  
disp('X1*=' );disp(X1)  
disp('f(X1*)=' );disp(F)  
k=1; C0=q;  
while k<=Mg+Ml  
for i=1 :n
```

```
a=(2*pi*C0(i)); b=degtorad(a);
e=C0(i)+0.2-((0.5/2*pi)*(sin(b))); y=floor(e);
C(k,i)= e-y;
C0(i)= C(k,i);
r=rand;
if r<=0.5
x(k,i)=X1(i)+l*L(k,i)*abs(U-X1(i));
else
x(k,i)=X1(i)-l*L(k,i)*abs(X1(i)-I);
end end
f1 = griewangk(k,x,n);
if f1<F
F=f1;
X2=x(k, :);
end
k=k+1;
end
disp ('Le minimum global est :');
disp ('X2*=' );disp(X2)
disp ('f(X2*=' ); disp(F)
if F<f
f=F;
Xg=X2;
end
S=S+f; w=w+f2;
end
```

```
disp ('La Meilleure valeur de f* est :');
disp ('X*=' );disp(Xg)
disp ('f(X*=' ); disp(f)
disp ('La valeur moyenne de f* est :');
Moy=S/m
disp ('L Écart-type de f* est :');
ET=sqrt((w/m)-(Moy)2)
```

## 5) Programme de la fonction de Griewank sous Matlab :

```
function [ f1 ] = griewangk( k,x,n )
```

```
s=0;
p=1;
for i=1 :n
s=s+(x(k,i)^2/4000);
v = x(k,i)/sqrt(i);
u = degtorad(v);
p = p * cos(u);
end
f1 = s - p + 1;
end
```

## 6) Programme de la fonction de Rastrigin sous Matlab :

```
function [f2 ] = rastrigin( x,n,k )

s=0;
for i=1 :n
f= x(k,i)^2;
a = (2 * pi * x(k,i));
b = degtorad(a);
m = 10 * cos(b);
s = s + f - m;
end
f2 = 10 * n + s;
end
```

## 7) Programme de la fonction de Sphère sous Matlab :

```
function [f3 ] = sphere( x,n,k )

s=0;
for i=1 :n
s=s+(x(k,i).^2);
end
f3 = sqrt(s);
end
```

## BIBLIOGRAPHIE

- [1] Azoug- Seif Eddine , " Développement et implémentation des techniques de cryptage des signaux image et vidéo ".2016, thèse –doctorat Université Ferhat abas –Sétif .
- [2] Back, T. Hammel, U. Schwefel, H.-P. Evolutionary Computation : Comments on the History and Current State. IEEE Transactions on Evolutionary Computations, 1(1), 3-17 (1997).
- [3] Becker, R. W. Lago, G. V. A Global Optimization Algorithm, Proceedings of the 8th Allerton Conference on Circuits and Systems Theory. 1970.
- [4] Bucolo, M. Caponetto,R. Fortuna, L. Frasca, M. Rizzo, A. Does chaos work better than noise? IEEE Circuits and Systems Magazine 2 (3) (2002) 4-19.
- [5] B. Li, W. Jiang, Optimization of complex functions by chaos search, International Journal of Cybernetics and Systems 29 (4) (1998) 409-419.
- [6] Caponetto, R. Fortuna, L. Fazzino, S. Xibilia, M.G. Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 289-304.
- [7] Chen, L. Aihara, K. Optimization by chaotic simulated annealing, in : Proceedings of the International Conference of Sino-Japanese Young Scientist, 1995, pp. 3-57.
- [8] Coelho, L.D.S. : Tuning of PID Controller for an Automatic Regulator Voltage System using Chaotic Optimization Approach. Chaos, Solitons and Fractals. 39(4), 1504-1514 (2009).
- [9] F. Derouiche, thèse doctorat (Analyse de l'effet d'utilisation du chaos sur l'amélioration des performances des méthodes numériques), Université Larbi Ben M'hidi Oum El Bouaghi, 2020-2021.
- [10] F.Zamime, M.A.Slimane Fissa,'Analyse et synchronisation des systèmes chaotiques de type hyperjerk', Mémoire de master, universite de Blida1, 2019.

- [11] Gandomi AH, Yang XS. Benchmark problems in structural optimization. In : Kozziel S, Yang XS, editors. Computational optimization methods and algorithms. Berlin : Springer-Verlag; 2011. p. 259–81.
- [12] Gérard Berthiau, Patrick Siarry Rario operations research. Res. 35 (2001) 329-365.
- [13] Griewangk AO. Generalized descent of global optimization. J Optim Theor Appl 1981 ;34 :11–39.
- [14] Guilhèm Assael, Lucas Blaizot et Geert Huizing : "TPE de 1ere S-SI : La théorie du chaos", Lycée du Sacré-Coeur et lycée Saint-Eloi Année scolaire 2013 - 2014, 20-27.
- [15] Holland, J. H. Genetic Algorithms and the Optimal Allocation of Trials, SIAM Journal on Computing, vol. 2, p. 88-105. 1973.
- [16] Ibbitt, R. P. Donnell, T. Designing conceptual catchment models for automatic fitting methods. IAHS Publication n° 101, 462-475, 1971.
- [17] Julio Alexander AGUILAR ANGULO- " Conception d'un Générateur de Valeurs aléatoires en Technologie CMOS AMS 0.35m" , ECOLE DOCTORALE Equipe conception de circuit Juin 2015. THESE doctorat.
- [18] Jun Lang , Ran Tao, Yue Wang Département "Image encryption based on the multiple-parameter discrete fractional Fourier transform and chaos function " ,2010.
- [19] J. R. Koza, Genetic Programming. MIT Press. 1992.
- [20] Kalos, M.H. et Whitlock, P.A. Monte Carlo Methods, volume I : Basics. John Wiley and Sons, 1986.
- [21] krim mohamed (Implémentation des séquences chaotiques sur les systèmes de communication moderne :Étalement de spectre à séquence directe DS-SS) université de la science et de la technologie MOHAMMED BOUDHIAF a Oran .2018/2019, thèse doctorat.
- [22] Le Riche, R. Cours : Introduction à l'Optimisation Globale. Disponible sur <http://www.emse.fr/~leriche> (2004).
- [23] Levy, A.V., Montalvo, A. The tunneling algorithm for the global minimization of functions. SIAM J. of Sci. and Stat. Comp., 1(15), 15-29 (1985).
- [24] LETELLIER C. Introduction. Le chaos dans la nature. Paris : Vuibert, 2006, 1-5.
- [25] Li, B. and Jiang, W. "Chaos optimization method and its application". Control Theory and Application, vol. 14, Aug. 1997, pp. 613-615.
- [26] Lozi R., (1978), Un Attracteur étrange du Type Attracteur de Hénon ". Journal de Physique. Colloque C5 ; Supplément au no8, Tome 39, pp.9-10.

- [27] May RM. Simple mathematical models with very complicated dynamics. *Nature* 1976;261 :459.
- [28] Michalewicz, Z. Genetic algorithms + Data structures = evolution programs. Springer, 1999.
- [29] Peitgen H, Jurgens H, Saupe D. Chaos and fractals. Berlin, Germany : Springer-Verlag; 1992.
- [30] Rastrigin LA. External control systems. In : Theoretical foundations of engineering cybernetics series. Moscow, Russian : Nauka ; 1974.
- [31] Rinnooy Kan, A. H. G. Timmer, G. T. Stochastic Global Optimization Methods; Part I : Clustering Methods, Mathematical Programming, vol. 39, pp. 27-56, 1987.
- [32] R. Horst H. Tuy, Global Optimization, Deterministic Approach, Springer-Verlag, Berlin. (1993).
- [33] R.L. Devaney, An introduction to chaotic dynamical systems. In Addison-Wesley, Redwood City, CA 37, 2003.
- [34] R. Horst, P. M. Pardalos, Handbook of Global Optimization, Kluwer Academic Publishers, pp. 829-869, 1995.
- [35] R. Horst, P. M. Pardalos N. V. Thoai, Introduction to Global Optimization, Kluwer Academic Publishers, Dordrecht. (1995).
- [36] Schoenauer, M. Michalewicz, Z. Evolutionary Computation : An Introduction. Control and Cybernetics, Special Issue on Evolutionary Computation, 26 (3), pp. 307-338, 1997.
- [37] Shayeghi, H. Jalilzadeh, S. Shayanfar, H.A. Safari, A. : Robust PSS Design using Chaotic Optimization Algorithm for a Multimachine Power System. ECTI-CON 2009, Pattaya, Thailand. 40-43 (2009).
- [38] Shayeghi, H. Shayanfar, H.A. Jalilzade, S. Safari, A. A PSO based Unified Power Flow Controller for Damping of Power System Oscillations. Energy Conversion and Management. 50(10), 2583-2592 (2009).
- [39] S. Janson, and M. Middendorf, A hierarchical particle swarm optimizer and its adaptive variant. IEEE Trans. Syst., Man, Cybern. - Part B : Cybernetics, vol. 35, no. 6, pp. 1272-1282, 2005.
- [40] S.S. Rao, Engineering Optimization Theory and Practice. Wiley, 2009.
- [41] Strogatz SH. Nonlinear dynamics and chaos. Massachusetts : Perseus Publishing ; 2000.
- [42] Torn, A. A. A Program for Global Optimization, Multistart with Clustering (MSC), Proceedings of Euro IFIP 79, North-Holland, pp. 427-434. 1979.
- [43] Torn, A.A., Zilinskas, A. Global Optimization. Springer-Verlag, Berlin (1989).

- [44] T. Hamaizia, (Systemes Dynamiques et Chaos "Application à l'optimisation a l'aide d'algorithme chaotique"), Université de Constantine -1-, 2013, thèse doctorat.
- [45] Wu, X.X. Chen, Z. Introduction of Chaos Theory, Shanghai Science and Technology, Bibliographic Publishing House, 1996.
- [46] X.X. Wu, Z. Chen, Introduction of Chaos Theory, Shanghai Science and Technology, Bibliographic Publishing House, 1996.
- [47] Y. Song, Z. Q. Chen, and Z. Z. Yuan, New chaotic PSO-based neural network predictive control for nonlinear process. IEEE Trans. Neural Networks vol. 18, no. 2, pp. 595-600, 2007.
- [48] Zheng WM. Kneading plane of the circle map. Chaos, Solitons Fractals 1994;4 :1221.
- [49] Z. ELHADJ, thèse doctorat (ÉTUDE DE QUELQUES TYPES DE SYSTÈMES CHAOTIQUES : GÉNÉRALISATION D'UN MODÈLE ISSU DU MODÈLE DE CHEN), UNIVERSITE MENTOURI DE CONSTANTINE, 2006.