



Intelligence Artificielle

Harmony Search Algorithm

Réalisé par :

Asbagui Youssef

Desplaces Koumtossa

Sakhi Mehdi

Lahlou Mohammed

Sous la supervision de :

Mr. Idoumghar Lhassane

Sommaire :

- **Introduction**
- **Présentation général**
- **Etapes de l'algorithme**
- **Formulation Mathématique du problème**
- **Les classes**
- **Résultats de l'application**
- **Conclusion**
- **Références**

• **Introduction :**

En écoutant et en appréciant une pièce de musique classique, qui s'est demandé s'il peut y avoir un lien entre le fait de jouer de la musique et celui de trouver une solution optimale à un problème complexe tel que la conception d'un réseau d'eau potable ou un autre problème d'ingénierie moderne ? Ce lien est représenté par un algorithme nommé « Harmony Search Algorithm (HSA) » développé par Geem et al [1]. C'est une méta heuristique relativement nouvelle, dont l'avantage et l'efficacité ont été démontrés dans plusieurs applications. Depuis son apparition en 2001, HSA a été appliqué à plusieurs problèmes d'optimisation, entre autres : l'optimisation de fonctions, l'optimisation de l'ingénierie, conception de réseaux de distribution d'eau, le problème de transport, la classification par l'algorithme des k plus proches voisins et l'optimisation de la production dans les réseaux électriques [2, 3, 4, 5, 6, 7, 8, 9, 10]. D'autres applications intéressantes de HSA hybridée [11, 12, 13, 14] avec d'autres algorithmes d'optimisation tel l'algorithme d'optimisation par essaim particulaire [15] ou la méthode de recherche réactive [16] ont été proposées et étudiées. D'autres applications récentes de HSA sont décrites en référence [17]

• Présentation générale :

HSA a été conçu en s'inspirant du processus musical de la recherche d'un état parfait de l'harmonie. Les performances musicales cherchent à trouver une harmonie plaisante (l'état parfait) définie par un standard esthétique de la même manière que les processus d'optimisation cherchent à trouver une solution globale (l'état parfait) définie par une fonction objectif. Le ton de chaque instrument de musique détermine la qualité esthétique de la même manière que la valeur de la fonction objective est déterminée par les valeurs prises par chaque variable de décision. La figure II.1 [17] montre l'analogie entre l'improvisation musicale et le processus d'optimisation. En musique, chaque musicien joue un ton dans l'intervalle des tons possibles ce qui donne un vecteur d'harmonie. Si le résultat obtenu est une bonne harmonie, l'expérience est « sauvegardée » dans la mémoire de chaque musicien ce qui augmente la possibilité d'obtenir une bonne harmonie à la prochaine improvisation.

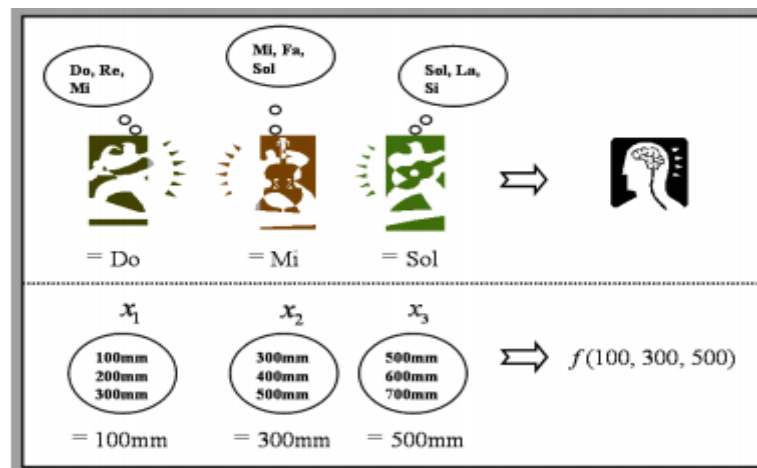


Figure II.1 Analogie entre improvisation musicale et processus d'optimisation

De façon similaire, en ingénierie, chaque variable de décision prend une valeur dans l'intervalle des valeurs possibles ce qui donne un vecteur d'une solution. Si toutes les valeurs des variables de décision représentent une bonne solution, l'expérience est sauvegardée en mémoire de chaque variable ce qui augmente la possibilité d'obtenir une bonne solution à l'itération suivante.

La modélisation du processus d'improvisation d'un ensemble de musiciens se base sur les règles qui dirigent chaque interprète dans l'élaboration de l'harmonie. En effet, lorsqu'un musicien improvise un ton, le plus souvent, il suit l'une des trois règles :

- a. jouer un ton de sa mémoire
- b. jouer un ton adjacent au ton de sa mémoire
- c. jouer un ton totalement aléatoire dans l'ensemble des sons possibles

Par analogie, quand HSA affecte une valeur à une variable de décision, il suit l'une des trois règles [18] :

- d. le choix d'une valeur quelconque de la Mémoire des Harmonies (Harmony Memory : HM) : « considération de la mémoire »
- e. le choix d'une valeur adjacente à la valeur de HM : « ajustement du ton »
- f. le choix d'une valeur totalement aléatoire dans l'intervalle des valeurs possibles : « randomisation ».

Ces trois règles de HSA sont dirigées en utilisant deux paramètres, qui sont :

- le taux de considération de la mémoire HM (Harmony Memory Considering Rate : HMCR)
- le taux d'ajustement du ton (Pitch Adjusting Rate : PAR).

- **Etapes de l'algorithme :**

La procédure d'optimisation de HSA est représentée dans la figure suivante :

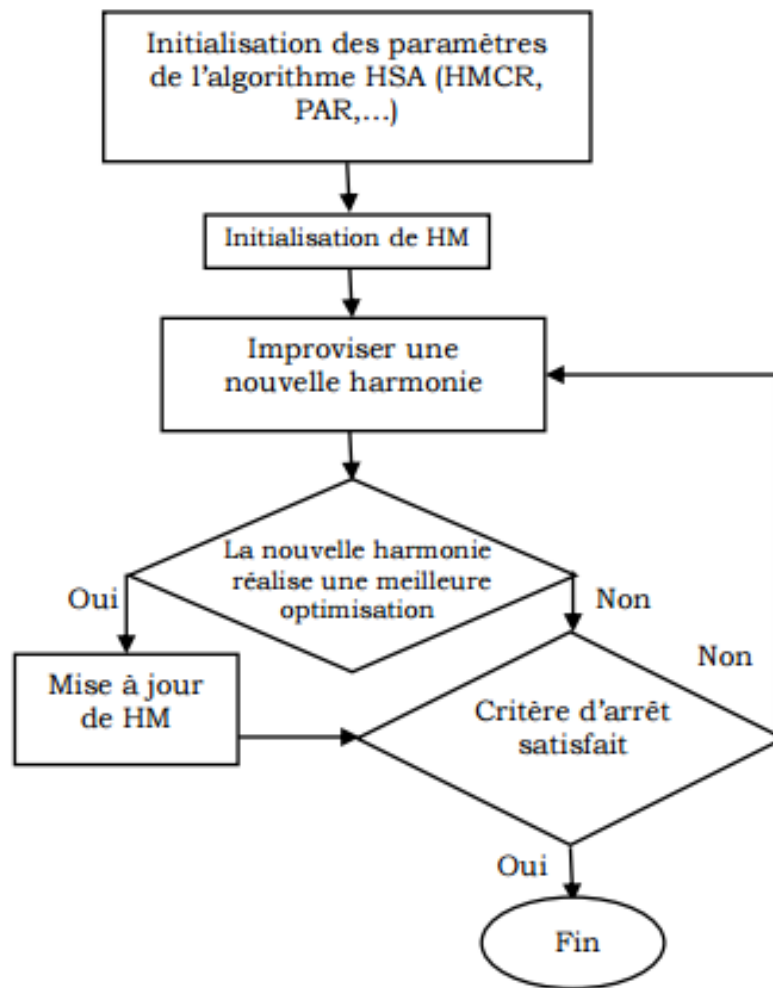


Figure III.1 procédure d'optimisation de HSA

Cette procédure est constituée des 5 étapes suivantes :

✓ **Etape 1 : Initialisation des paramètres de l'algorithme HSA**

Il faut noter que le problème d'optimisation spécifiquement résolu par HSA est sous la forme :

$$\begin{cases} \text{Min } f(x) \\ \text{avec } x^t = (x_1, x_2, \dots, x_i, \dots, x_N) \\ \text{et } x_i^{\text{MIN}} \leq x_i \leq x_i^{\text{MAX}} \quad \forall i = 1..N \end{cases}$$

$f(x)$ est la fonction objectif. x est le vecteur solution du problème. x_i est la valeur possible de la i ème variable de décision. Cette valeur est bornée par $\text{MIN } x_i$ et $\text{MAX } x_i$, N est le nombre de variables de décision. Dans cette étape, les paramètres suivant sont définis [30]:

- ✓ Le nombre d'exécutions de l'algorithme : Independent Runs
- ✓ Le nombre d'évolution que va subir le vecteur mémoire pour chaque itération de Independent Runs : Number Evolution Steps
- ✓ La taille du vecteur mémoire contenant les solutions : Harmony Memory Size (HMS) ou encore Population Size
- ✓ La taille d'une solutions : Solutions Size
- ✓ Le taux de considération de la mémoire d'harmonie : Harmony Memory Considering Rate (HMCR).
- ✓ Le taux d'ajustement du ton : Pitch Adjusting Rate (PAR).
- ✓ La valeur maximale d'ajustement : Band Width (BW)

✓ **Etape 2 : Initialisation du vecteur mémoire (HM)**

HM est une matrice décrite par l'équation III.2. Cette matrice est remplie initialement par des solutions générées aléatoirement selon la formule de génération III.3 :

$$HM = \begin{bmatrix} x_1^1 & \cdots & x_{HMS}^1 \\ \vdots & \ddots & \vdots \\ x_1^{HMS} & \cdots & x_{HMS}^{HMS} \end{bmatrix}$$

Figure III.2 la matrice HM

$$x^j = \begin{pmatrix} x_1^j \\ \vdots \\ x_i^j \\ \vdots \\ x_N^j \end{pmatrix} \quad x_i^j = x_i^{MIN} + r \times (x_i^{MAX} - x_i^{MIN})$$

r étant un nombre aléatoire dans l'intervalle $[0, 1]$.

Figure III.3 une solution de HM

✓ **Etape 3: Improvisation d'une nouvelle harmonie**

L'improvisation d'une nouvelle harmonie se fait variable par variable.

La valeur de chaque variable de décision x_i^j est choisie dans HM selon une forte probabilité HMCR sinon elle peut être générée aléatoirement (III.3) selon la probabilité $(1 - \text{HMCR})$. Ceci indique que le rôle de HMCR est décisif. En effet, si ce taux est trop petit, la sélection de HM ne se fera que pour une faible population de variables et l'algorithme risque de ne pas converger rapidement. Par contre, si HMCR est trop grand (proche de 1), c'est la génération de nouvelles variables qui sera limitée et donc l'exploration risque d'en souffrir. Pour ces raisons, la valeur typique de HMCR est généralement prise dans l'intervalle $[0.7, 0.95]$. Toute variable sélectionnée dans HM avec une probabilité HMCR est ajustée selon la probabilité PAR. Cet ajustement (III.4) consiste à remplacer la valeur de la variable courante par une autre valeur de son voisinage :

$$x_i^j = \begin{cases} x_i^j \pm bw & \text{avec une probabilité } PAR \\ x_i^j & \text{avec une probabilité } (1 - PAR) \end{cases}$$

Figure III.4 l'ajustement d'une variable

BW (Band Width) est un autre paramètre de l'algorithme. Ce paramètre représente le pas de variation de chaque variable de décision dans son voisinage. Notons, finalement, que pour toute nouvelle improvisation, chaque variable de décision va se trouver dans l'un des trois cas suivants :

- ✓ Générée aléatoirement selon l'équation III.3.
- ✓ Sélectionnée dans HM et gardée inchangée.
- ✓ Sélectionnée dans HM et ajustée selon l'équation III.4.

✓ **Etape 4 : Mise à jour de HM**

Si la nouvelle harmonie obtenue à l'étape 3 réalise une meilleure optimisation de la fonction objectif que la plus mauvaise harmonie déjà présente dans HM, alors HM est mis à jour en remplaçant cette plus mauvaise harmonie par cette nouvelle.

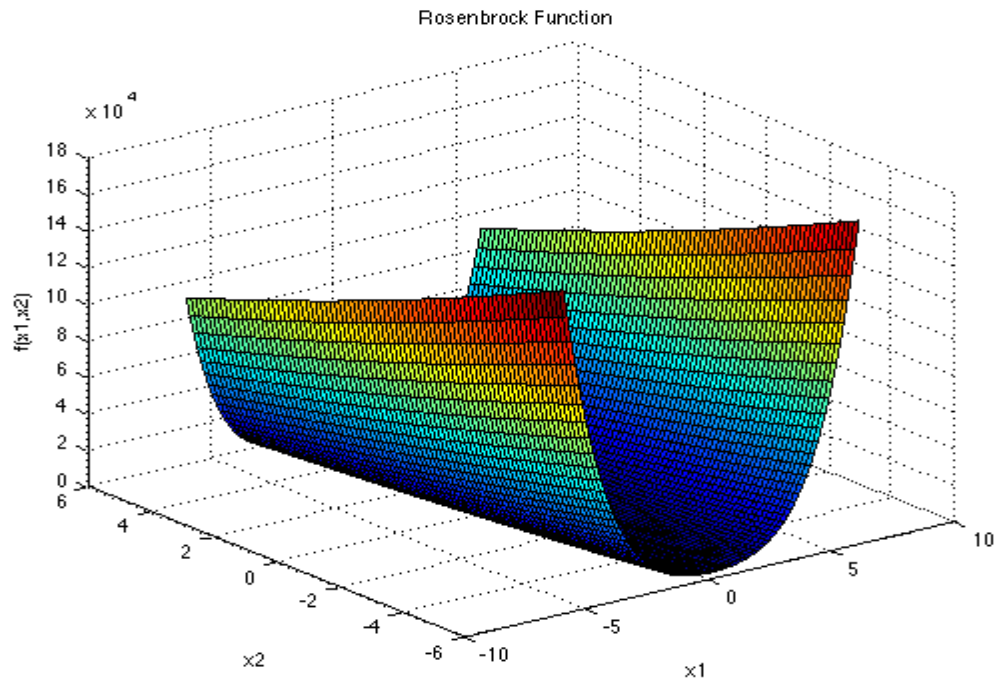
✓ **Etape 5 : Réitération jusqu'à satisfaction du critère d'arrêt**

Réitérer les étapes 3 et 4 jusqu'à ce que le nombre d'itérations maximum prédéfini soit atteint.

- **Formulation Mathématique du problème :**

Pour la validation de ce Projet, il nous a été demandé de valider 5 Benchmarks qui sont :

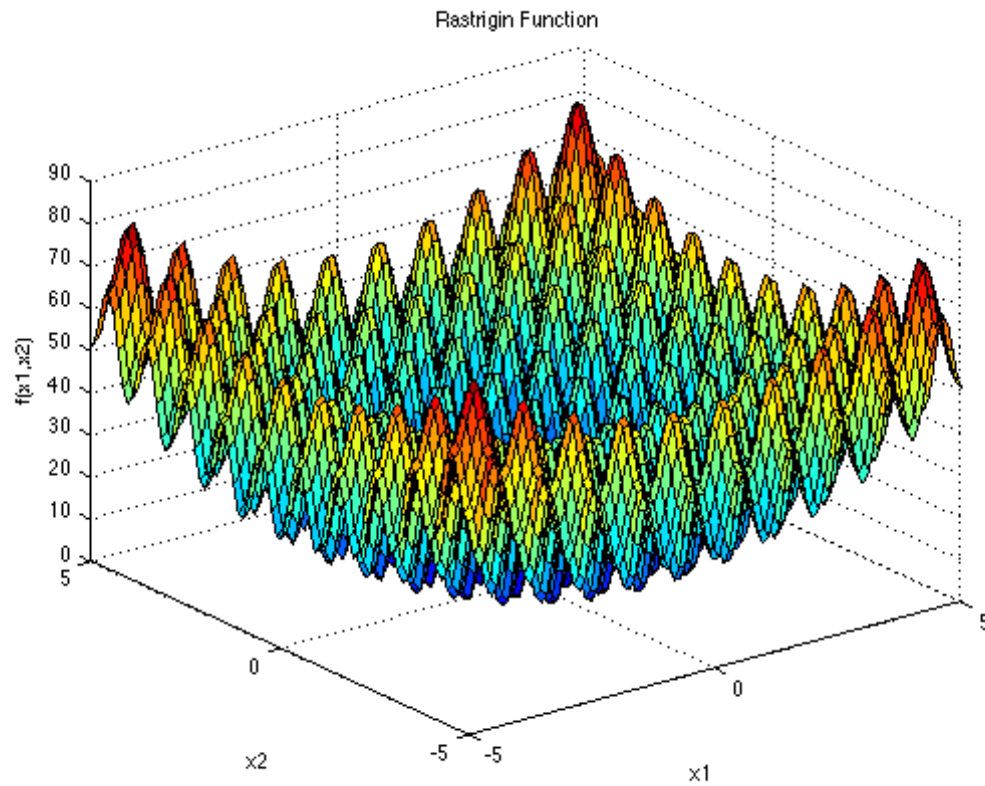
- ✓ **Rosenbrock :**



$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

Avec :

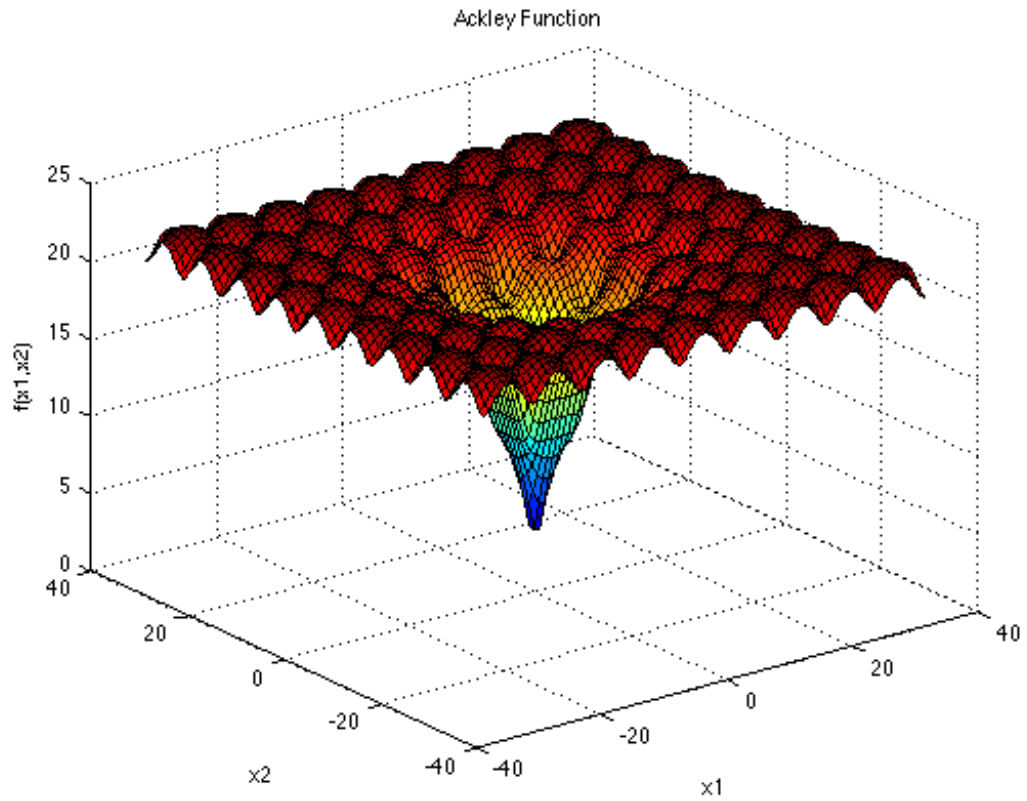
- **d** : la dimension du problème
- **[-2.048, 2.048]** : le domaine des valeurs des solutions

✓ **Rastrigin :**

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

Avec :

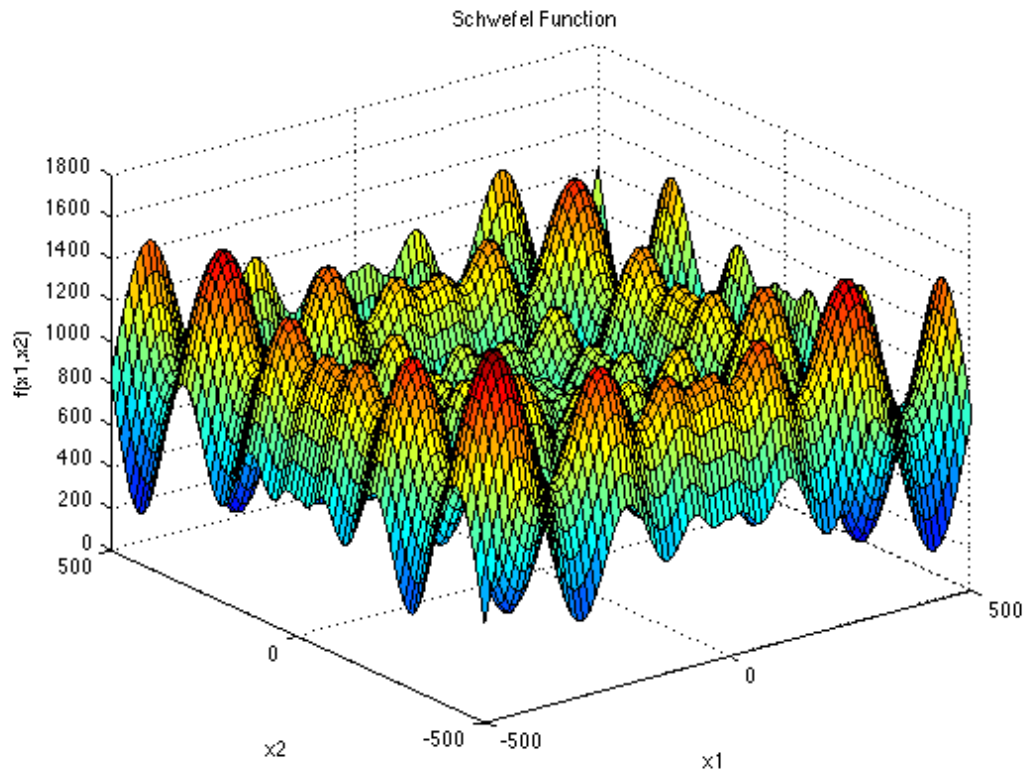
- **d : la dimension du problème**
- **[-5.12, 5.12] : le domaine des valeurs des solutions**

✓ Ackley :

$$f(\mathbf{x}) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

Avec :

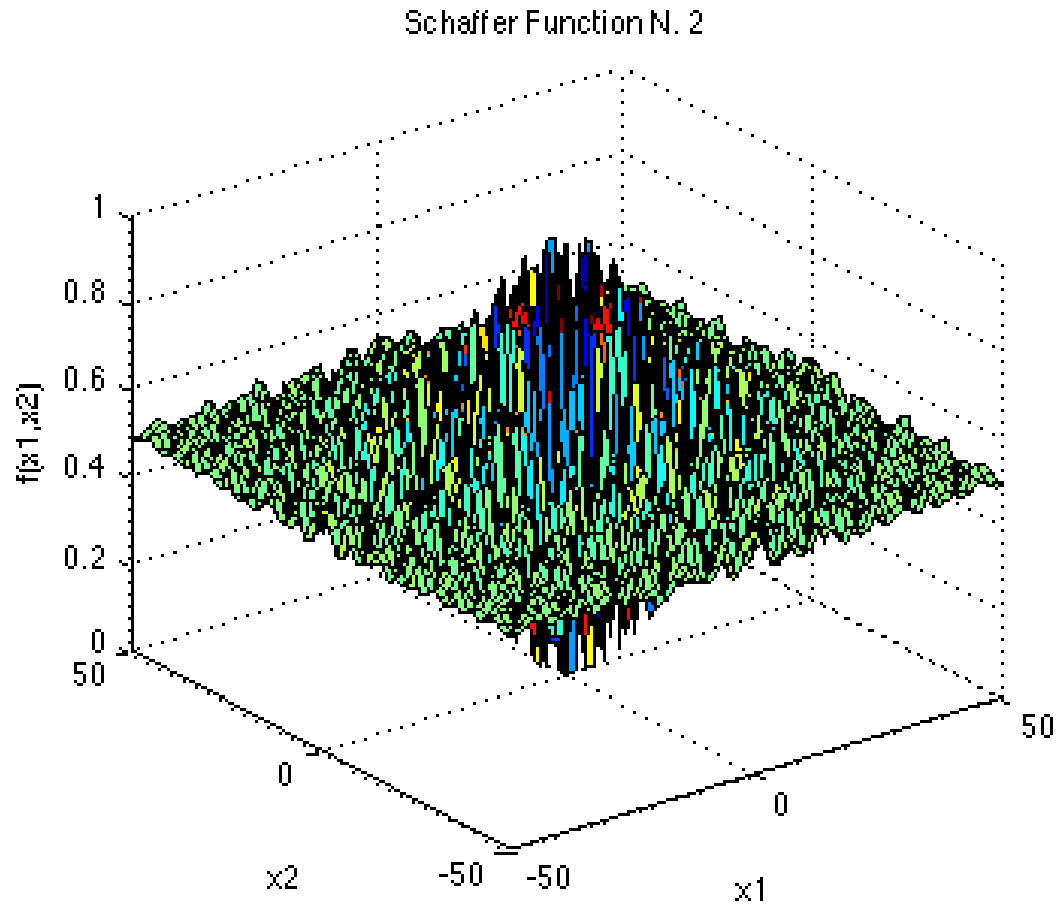
- **d** : la dimension du problème
- **a** : 20
- **b** : 0.2
- **c** : 2π
- **[-32.768, 32.768]** : le domaine des valeurs des solutions

✓ **Schwefel :**

$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|})$$

Avec :

- **d : la dimension du problème**
- **[-500, 500] : le domaine des valeurs des solutions**

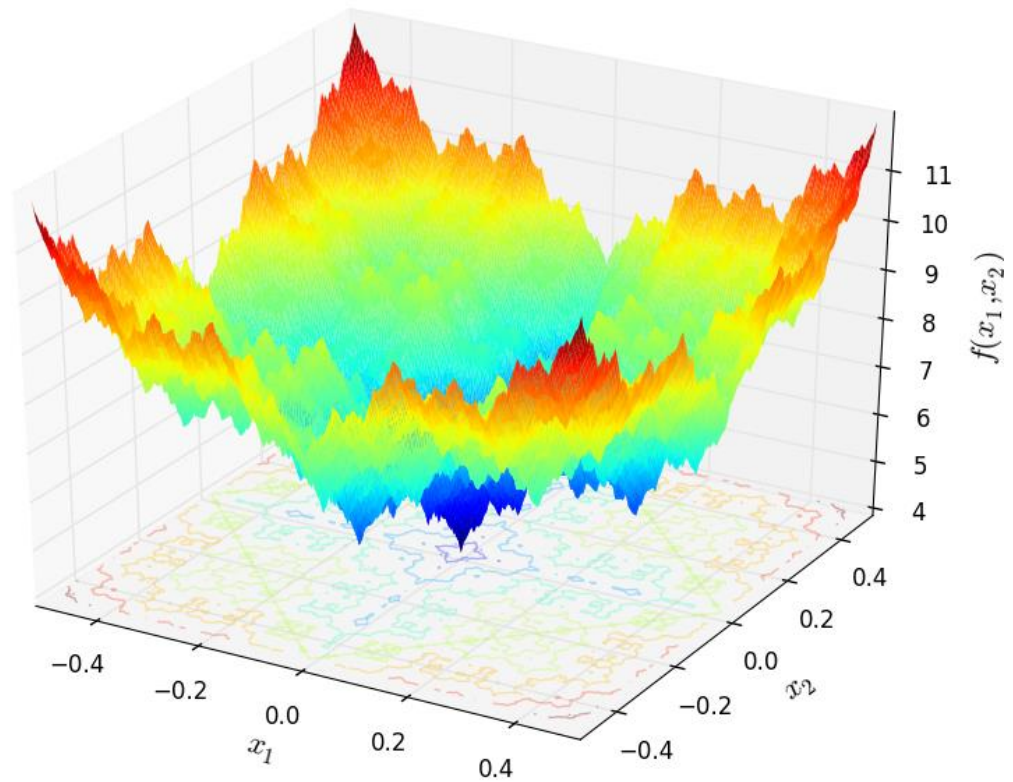
✓ **Schaffer :**

$$f(\mathbf{x}) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

Avec :

- la dimension du problème est 2
- [-100, 100] : le domaine des valeurs des solutions

✓ Weierstrass :



$$f_i(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)]$$

Avec :

- **D** : la dimension du problème
- **Kmax** : 20
- **a** : 0.5
- **b** : 3
- **[-5, 5]** : le domaine des valeurs des solutions

• Les classes :

Pour la réalisation de ce projet, on a cherché des codes déjà existant qui traitent notre algorithme sur différentes plateformes tel que Git Hub. On a ensuite choisi un des codes, et on a essayé de le modifier, de façon à ce qu'il respecte le squelette imposé par Mr. Idoumghar sur son site web.

On distingue 4 classes :

1. La classe Set Up Params :

Contient les paramètres de notre programme tel que :

- Independent Runs qui représente le nombre d'exécution de l'algorithme
- Number Evolution Steps qui est le nombre d'évolution que va subir notre vecteur de solutions à chaque itération d'Independent Runs
- HMCR qui est la probabilité de piocher une valeur déjà existante dans notre matrice
- ...

Tous les paramètres sont expliqués dans la section « Etapes de l'algorithme », paragraphe « Etape 1 : Initialisation des paramètres de l'algorithme HSA »

2. La classe Problème :

Contient les paramètres du problème :

- La dimension du problème
- L'index qui identifie le problème
- Lower Limit : la valeur minimale, ou encore la borne inférieure du domaine des valeurs possibles
- Upper Limit : la valeur maximale, ou encore la borne supérieure du domaine des valeurs possibles

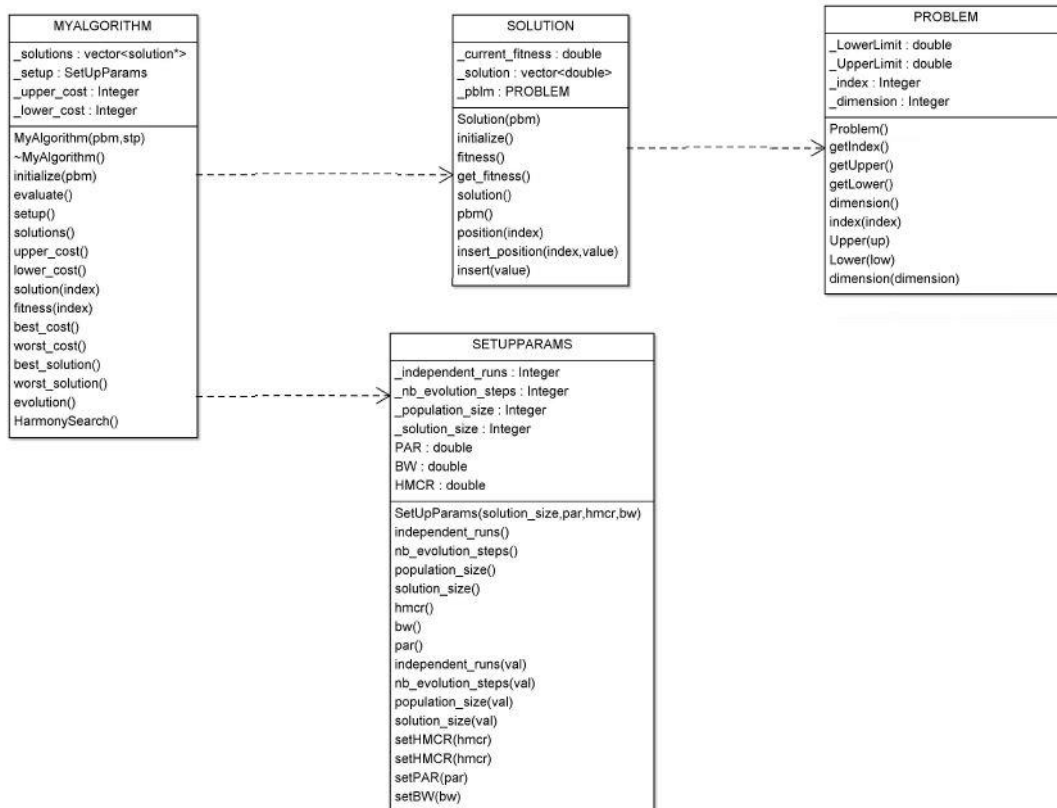
3. La classe Solution :

Contient un objet de type Problème, et un tableau qui est remplie avec de valeurs aléatoires entre le Lower Limit et le Upper Limit du problème. La taille de ce tableau est la dimension du problème. Elle contient aussi le paramètre Curent Fitness qui stock la fitness de son tableau.

4. La classe My Algorithm :

Contient un objet de type Set Up Params qui contient les paramètres du programme, et un vecteur de solutions. C'est au niveau de cette classe qu'on implémente notre algorithme.

On résume tout ceci dans le diagramme de classes suivant :



● Résultats de l'application :

Tous les tests ont été effectués avec les paramètres :

- HMCr : 0.95
- PAR : 0.9
- BW : 0.3

Et on trouve les résultats suivants :

✓ Rosenbrock :

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 0
independent run 0: la meilleur fitness est : 106.71
independent run 1: la meilleur fitness est : 92.2746
independent run 2: la meilleur fitness est : 90.3907
independent run 3: la meilleur fitness est : 77.6969
independent run 4: la meilleur fitness est : 56.5491
independent run 5: la meilleur fitness est : 56.5491
independent run 6: la meilleur fitness est : 56.5491
independent run 7: la meilleur fitness est : 56.5491
independent run 8: la meilleur fitness est : 56.5491
independent run 9: la meilleur fitness est : 56.5491
independent run 10: la meilleur fitness est : 56.5491
independent run 11: la meilleur fitness est : 56.5491
independent run 12: la meilleur fitness est : 56.5491
independent run 13: la meilleur fitness est : 56.5491
independent run 14: la meilleur fitness est : 56.5491
independent run 15: la meilleur fitness est : 56.5491
independent run 16: la meilleur fitness est : 56.5491
independent run 17: la meilleur fitness est : 56.5491
independent run 18: la meilleur fitness est : 56.5491
independent run 19: la meilleur fitness est : 56.5491
independent run 20: la meilleur fitness est : 56.5491
independent run 21: la meilleur fitness est : 56.5491
independent run 22: la meilleur fitness est : 56.5491
independent run 23: la meilleur fitness est : 56.5491
independent run 24: la meilleur fitness est : 56.5491
independent run 25: la meilleur fitness est : 56.5491
independent run 26: la meilleur fitness est : 56.5491
independent run 27: la meilleur fitness est : 56.5491
independent run 28: la meilleur fitness est : 56.5491
independent run 29: la meilleur fitness est : 56.5491

la meilleur fitness est : 56.5491
la moyenne est : 61.245
l'ecart type est : 12.5478
-----
Process exited after 53.09 seconds with return value 0
```

✓ Rastrigin :

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 1
independent run 0: la meilleur fitness est : 136.527
independent run 1: la meilleur fitness est : 127.251
independent run 2: la meilleur fitness est : 123.324
independent run 3: la meilleur fitness est : 123.324
independent run 4: la meilleur fitness est : 123.324
independent run 5: la meilleur fitness est : 123.324
independent run 6: la meilleur fitness est : 123.324
independent run 7: la meilleur fitness est : 115.441
independent run 8: la meilleur fitness est : 115.441
independent run 9: la meilleur fitness est : 115.441
independent run 10: la meilleur fitness est : 109.019
independent run 11: la meilleur fitness est : 109.019
independent run 12: la meilleur fitness est : 109.019
independent run 13: la meilleur fitness est : 109.019
independent run 14: la meilleur fitness est : 109.019
independent run 15: la meilleur fitness est : 109.019
independent run 16: la meilleur fitness est : 109.019
independent run 17: la meilleur fitness est : 109.019
independent run 18: la meilleur fitness est : 109.019
independent run 19: la meilleur fitness est : 109.019
independent run 20: la meilleur fitness est : 109.019
independent run 21: la meilleur fitness est : 109.019
independent run 22: la meilleur fitness est : 109.019
independent run 23: la meilleur fitness est : 109.019
independent run 24: la meilleur fitness est : 109.019
independent run 25: la meilleur fitness est : 109.019
independent run 26: la meilleur fitness est : 109.019
independent run 27: la meilleur fitness est : 109.019
independent run 28: la meilleur fitness est : 109.019
independent run 29: la meilleur fitness est : 109.019

la meilleur fitness est : 109.019
la moyenne est : 113.57
l'ecart type est : 7.33612
-----
Process exited after 164.5 seconds with return value 0
```

✓ Ackley:

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 2
independent run 0: la meilleur fitness est : 1.22882
independent run 1: la meilleur fitness est : 1.0746
independent run 2: la meilleur fitness est : 1.0746
independent run 3: la meilleur fitness est : 1.0746
independent run 4: la meilleur fitness est : 1.0746
independent run 5: la meilleur fitness est : 1.0746
independent run 6: la meilleur fitness est : 1.0746
independent run 7: la meilleur fitness est : 1.0746
independent run 8: la meilleur fitness est : 1.0746
independent run 9: la meilleur fitness est : 1.0746
independent run 10: la meilleur fitness est : 1.0746
independent run 11: la meilleur fitness est : 1.0746
independent run 12: la meilleur fitness est : 1.0746
independent run 13: la meilleur fitness est : 1.0746
independent run 14: la meilleur fitness est : 1.0746
independent run 15: la meilleur fitness est : 1.0746
independent run 16: la meilleur fitness est : 1.0746
independent run 17: la meilleur fitness est : 1.0746
independent run 18: la meilleur fitness est : 1.0746
independent run 19: la meilleur fitness est : 1.0746
independent run 20: la meilleur fitness est : 1.0746
independent run 21: la meilleur fitness est : 1.0746
independent run 22: la meilleur fitness est : 1.0746
independent run 23: la meilleur fitness est : 1.0746
independent run 24: la meilleur fitness est : 1.0746
independent run 25: la meilleur fitness est : 1.0746
independent run 26: la meilleur fitness est : 1.0746
independent run 27: la meilleur fitness est : 1.0746
independent run 28: la meilleur fitness est : 1.0746
independent run 29: la meilleur fitness est : 1.0746

la meilleur fitness est : 1.0746
la moyenne est : 1.07974
l'ecart type est : 0.0276841
-----
Process exited after 177.6 seconds with return value 0
```

✓ Schwefel :

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 3
independent run 0: la meilleur fitness est : 10.9633
independent run 1: la meilleur fitness est : 0.121585
independent run 2: la meilleur fitness est : 0.0889989
independent run 3: la meilleur fitness est : 0.0889989
independent run 4: la meilleur fitness est : 0.0888647
independent run 5: la meilleur fitness est : 0.0888647
independent run 6: la meilleur fitness est : 0.0888647
independent run 7: la meilleur fitness est : 0.0604839
independent run 8: la meilleur fitness est : 0.0604839
independent run 9: la meilleur fitness est : 0.0604839
independent run 10: la meilleur fitness est : 0.0604839
independent run 11: la meilleur fitness est : 0.0604839
independent run 12: la meilleur fitness est : 0.0604839
independent run 13: la meilleur fitness est : 0.0604839
independent run 14: la meilleur fitness est : 0.0604839
independent run 15: la meilleur fitness est : 0.0604839
independent run 16: la meilleur fitness est : 0.0604839
independent run 17: la meilleur fitness est : 0.0604839
independent run 18: la meilleur fitness est : 0.0604839
independent run 19: la meilleur fitness est : 0.0604839
independent run 20: la meilleur fitness est : 0.0604839
independent run 21: la meilleur fitness est : 0.0604839
independent run 22: la meilleur fitness est : 0.0604839
independent run 23: la meilleur fitness est : 0.0604839
independent run 24: la meilleur fitness est : 0.0604839
independent run 25: la meilleur fitness est : 0.0604839
independent run 26: la meilleur fitness est : 0.0604839
independent run 27: la meilleur fitness est : 0.0604839
independent run 28: la meilleur fitness est : 0.0604839
independent run 29: la meilleur fitness est : 0.0604839

la meilleur fitness est : 0.0604839
la moyenne est : 0.430687
l'ecart type est : 1.95591
-----
Process exited after 189.7 seconds with return value 0
```

✓ Schaffer :

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 4
independent run 0: la meilleur fitness est : 1.46477e-009
independent run 1: la meilleur fitness est : 1.27142e-009
independent run 2: la meilleur fitness est : 1.27142e-009
independent run 3: la meilleur fitness est : 7.02789e-010
independent run 4: la meilleur fitness est : 1.13337e-010
independent run 5: la meilleur fitness est : 1.13337e-010
independent run 6: la meilleur fitness est : 1.13337e-010
independent run 7: la meilleur fitness est : 1.13337e-010
independent run 8: la meilleur fitness est : 1.13337e-010
independent run 9: la meilleur fitness est : 1.13337e-010
independent run 10: la meilleur fitness est : 1.13337e-010
independent run 11: la meilleur fitness est : 1.13337e-010
independent run 12: la meilleur fitness est : 1.13337e-010
independent run 13: la meilleur fitness est : 1.13337e-010
independent run 14: la meilleur fitness est : 1.13337e-010
independent run 15: la meilleur fitness est : 1.13337e-010
independent run 16: la meilleur fitness est : 1.13337e-010
independent run 17: la meilleur fitness est : 1.13337e-010
independent run 18: la meilleur fitness est : 1.04701e-010
independent run 19: la meilleur fitness est : 1.04701e-010
independent run 20: la meilleur fitness est : 1.04701e-010
independent run 21: la meilleur fitness est : 1.04701e-010
independent run 22: la meilleur fitness est : 1.04701e-010
independent run 23: la meilleur fitness est : 1.04701e-010
independent run 24: la meilleur fitness est : 1.04701e-010
independent run 25: la meilleur fitness est : 1.04701e-010
independent run 26: la meilleur fitness est : 1.04701e-010
independent run 27: la meilleur fitness est : 1.04701e-010
independent run 28: la meilleur fitness est : 1.04701e-010
independent run 29: la meilleur fitness est : 1.04701e-010

la meilleur fitness est : 1.04701e-010
la moyenne est : 2.51784e-010
l'ecart type est : 3.77801e-010
-----
Process exited after 32.56 seconds with return value 0
```

✓ Weierstrass :

```
Rosenbrock : 0
Rastrigin : 1
Ackley : 2
Schwefel : 3
Schaffer : 4
Weierstrass : 5

Votre choix : 5
independent run 0: la meilleur fitness est : 29.2924
independent run 1: la meilleur fitness est : 29.2924
independent run 2: la meilleur fitness est : 29.2924
independent run 3: la meilleur fitness est : 29.2924
independent run 4: la meilleur fitness est : 27.7391
independent run 5: la meilleur fitness est : 27.574
independent run 6: la meilleur fitness est : 27.574
independent run 7: la meilleur fitness est : 26.907
independent run 8: la meilleur fitness est : 26.907
independent run 9: la meilleur fitness est : 26.907
independent run 10: la meilleur fitness est : 26.907
independent run 11: la meilleur fitness est : 26.7543
independent run 12: la meilleur fitness est : 26.2455
independent run 13: la meilleur fitness est : 26.2455
independent run 14: la meilleur fitness est : 26.2455
independent run 15: la meilleur fitness est : 26.2455
independent run 16: la meilleur fitness est : 26.2455
independent run 17: la meilleur fitness est : 26.0765
independent run 18: la meilleur fitness est : 26.0765
independent run 19: la meilleur fitness est : 26.0765
independent run 20: la meilleur fitness est : 26.0765
independent run 21: la meilleur fitness est : 26.0765
independent run 22: la meilleur fitness est : 26.0765
independent run 23: la meilleur fitness est : 23.9389
independent run 24: la meilleur fitness est : 23.9389
independent run 25: la meilleur fitness est : 23.9389
independent run 26: la meilleur fitness est : 23.9389
independent run 27: la meilleur fitness est : 23.9389
independent run 28: la meilleur fitness est : 23.9389
independent run 29: la meilleur fitness est : 23.9389

la meilleur fitness est : 23.9389
la moyenne est : 26.3233
l'ecart type est : 1.65466
-----
Process exited after 5710 seconds with return value 0
```


On regroupe les résultats dans un tableau :

Benchmark	Best Fitness	Moyenne	Ecart Type
Rosenbrock	56.5491	61.245	12.5478
Rastrigin	109.019	113.57	7.33612
Ackley	1.0746	1.07974	0.0276841
Schwefel	0.0604839	0.430687	1.95591
Schaffer	$1.04701 \cdot 10^{-10}$	$2.51784 \cdot 10^{-10}$	$3.77801 \cdot 10^{-10}$
Weierstrass	23.9389	26.3233	1.65466

En comparant avec un algorithme de la littérature :

Benchmark	Minimum Global	Nos Fitness	Fitness littérature
Rosenbrock	0	56.5491	24.58
Rastrigin	0	109.019	1.390625
Ackley	0	1.0746	0.723
Schwefel	0	0.0604839	$4.117 \cdot 10^{-6}$
Schaffer	0	$1.04701 \cdot 10^{-10}$	$6.84132 \cdot 10^{-3}$
Weierstrass	0	23.9389	0.826

• Conclusion :

Nous nous sommes intéressés dans ce projet à la résolution de problèmes d'optimisation académiques difficiles à travers l'algorithme HSA.

Notre objectif à été de proposer un code pour la résolution de ce type de problèmes. Les Benchmarks qui ont été proposé par Mr. Idoumghar Lhassane, et que nous traitons dans ce rapport sont : Rosenbrock, Rastrigin, Ackley, Schwefel, Schaffer et Weierstrass.

Afin de réaliser notre but, nous avons organisé notre travail en deux phases.

Dans la première phase, nous nous sommes concentrés sur la recherche d'articles dans la littérature expliquant le fonctionnement de l'algorithme qui nous a été confié. Nous avons essayé de récolter des notions de bases en optimisation, comprendre le rôle de la théorie, et chercher des informations sur les problèmes.

En ce qui concerne la seconde phase, nous avons ensuite cherché des codes déjà existant qui traite notre algorithme, et nous avons essayé de les traduire pour qu'ils correspondent aux différentes classes et méthodes donné par Mr. Idoumghar.

Nous Pensons que les résultats qu'on trouve à travers notre code sont assez satisfaisants, par rapport à la durée du projet.

• Références :

- <http://www.sciencedirect.com/science/article/pii/S2090447913000610>
- <https://www.lri.fr/~hansen/Tech-Report-May-30-05.pdf>
- https://books.google.fr/books?id=gwUwIEPqk30C&pg=PA657&lpg=PA657&dq=weierstrass+benchmark&source=bl&ots=GKr-BtPae6&sig=Oh2VITH_mYwx5REBCEk2wfvpylg&hl=fr&sa=X&sqi=2&ved=0ahUKEwjSmO6bxaHRAhWCvBoKHd1nCA5Q6AEISDAF#v=onepage&q=weierstrass%20benchmark&f=false
- <https://books.google.fr/books?id=TflUCgAAQBAJ&pg=PA163&lpg=PA163&dq=schaffer+benchmark+harmony+search+algorithm&source=bl&ots=XTz5mmVXmH&sig=NCDwOvoSuoJmCvqi3P3ZGiXva2o&hl=fr&sa=X&ved=0ahUKEwjGgLf e0KHRAhWFExoKHVwXCKEQ6AEIKTAB#v=onepage&q=schaffer%20benchmark%20harmony%20search%20algorithm&f=false>
- <https://www.sfu.ca/~ssurjano/optimization.html>
- <http://mistis.inrialpes.fr/people/people/forbes/PAPERS/fp801-qin.pdf>
- <https://github.com/antogian/HarmonySearch>
- <http://www.sciencedirect.com/science/article/pii/S0898122112004907>