

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE



Département Hydraulique

Laboratoire Matériaux de Génie Civil Et Environnement

Mémoire de Master en Hydraulique

Etude des performances de l'algorithme Intelligent Mesh for Optimum Search et application au modèle GR2M

Présenté et soutenu publiquement le 19/09/2017

Par Mr **Rafik OULEBSIR**

Sous la direction du **Pr. Abdelmalek BERMAD**

Co-dirigé par M. **Mohamed AMIRECHE**

Composition du Jury :

Président	M. Abdelouaheb LEFKIR	MCA	ENSTP
Promoteurs :	M. Abdelmalek BERMAD	Professeur	ENP
	M. Mohamed AMIRECHE	MAA	U-OEB
Examineur	M. Khmissi HOUARI	MAA	UMBB

ENP 2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE



Département Hydraulique

Laboratoire Matériaux de Génie Civil Et Environnement

Mémoire de Master en Hydraulique

Etude des performances de l'algorithme Intelligent Mesh for Optimum Search et application au modèle GR2M

Présenté et soutenu publiquement le 19/09/2017

Par Mr **Rafik OULEBSIR**

Sous la direction du **Pr. Abdelmalek BERMAD**

Co-dirigé par M. **Mohamed AMIRECHE**

Composition du Jury :

<i>Président</i>	M. Abdelouaheb LEFKIR	MCA	ENSTP
<i>Promoteurs :</i>	M. Abdelmalek BERMAD	Professeur	ENP
	M. Mohamed AMIRECHE	MAA	U-OEB
<i>Examineur</i>	M. Khmissi HOUARI	MAA	UMBB

ENP 2017

وتلخص هذه الأطروحة سير عمل خوارزمية (إيموس) يحاول هذا التقنية الجمع بين مزايا خوارزميات التحسين العامة والمحلية للتعامل بشكل أفضل مع مشاكل النمذجة الهيدرولوجية وأيضاً مع مشاكل أخرى باستخدام خوارزميات التحسين. الكلمات المفتاحية : GR2M-

Abstract :

This thesis summarizes the functioning of the Intelligent Mesh Optimum Search (IMOS) algorithm, which attempts to combine the advantages of global and local optimization algorithms to better deal with hydrological modeling problems, and also to other problems using optimization algorithms.

Keywords: optimization-GR2M.

Résumé:

Cette thèse résume le fonctionnement de l'algorithme d'optimisation IMOS (Intelligent Mesh For Optimum Search), cette technique essaye d'allier les avantages des algorithmes d'optimisations globales et locales, et cela pour mieux faire face aux problèmes de modélisation hydrologiques, et aussi a d'autres problématiques faisant appel aux algorithmes d'optimisation.

Mots clés : optimisation- GR2M

Remerciements

Avant tout je remercie le bon dieu de m'avoir guidée et m'a permis de prospérer dans ce monde.

C'est avec beaucoup d'émotions que je voudrais remercier toute les personnes qui ont contribué de près ou de loin à la réalisation de ce travail, qui m'ont permis de pouvoir passer au travers des plusieurs difficultés que j'ai pu rencontrer, a tous ceux qui m'ont soutenus je voudrais dire merci.

Je remercie aussi mon encadreur Mr A.Bermed qui tout au long de mon cursus universitaire au sein de l'Ecole National Polytechnique a su m'apporter tout l'aide qui était en son possible, et qui m'appris beaucoup tant au niveau scientifique qu'au niveau humain, c'est avec lui que dans le présent travail j'ai pu dépasser mes limites et ainsi ressortie le potentiel que j'avais, et pour cela je ne pourrais le remercier assez.

Je remercie aussi le corps enseignants qui a pu nous transmettre un peu de leur connaissance et ainsi nous permettre de nous améliorer chaque jour passée dans l'Ecole National Polytechnique, mais aussi à l'Ecole National Préparatoire de Sciences et Technique de Tlemcen.

Je tiens aussi à remercier aussi mes camarades de classe avec qui j'ai pu passer trois années formidables, mais aussi mes amis d'enfance avec qui je partage jusqu'à maintenant des souvenirs formidables ; je remercie aussi les membres de ma famille qui ont pu me soutenir tout au long de ma vie, et qui ont pu me permettre d'arriver jusqu'ici, je tiens aussi à remercier mes parents pour tous les sacrifices qu'ils ont du faire pour que je puisse réussir dans ma vie.

Je remercie aussi les membres du jury qui ont m'ont fait l'honneur de bien vouloir juger mon travail, et me permettre de pouvoir m'améliorer et de pouvoir avancer dans ma vie.

Table des Matières :

Liste des Tableaux

Liste des Figures

Introduction Générale	7
Chapitre 1 : introduction à l'optimisation	9
1. Problème d'optimisation	10
2. Difficulté que représente l'optimisation des modèles pluie-débit	11
2.1. Interdépendance des paramètres	11
2.2. Indifférence de la fonction objective face aux paramètres	12
2.3. Direction du gradient	12
2.4. les optimums locaux	13
3. Les Algorithmes d'optimisation	13
3.1. Les méthodes déterministes	14
3.2. Les méthodes stochastiques non inspirées de la nature	15
3.3. Les algorithmes inspirés de la Nature	15
3.4. Les algorithmes hybrides ou mimétiques	17
4. Les algorithmes d'optimisation : de l'exploration à l'exploitation	18
5. Critères de Choix d'un algorithme d'optimisation	19
6. Conclusion	20
Chapitre 2 : Particle Swarm Optimization et Intelligent Mesh for Optimum Search	21
1. Particles swarm optimization (Optimisation par essaim de particules)	22
1.1. Caractéristiques de l'algorithme	24
2. Intelligent Mesh for Optimum Search (IMOS)	25
3. Conclusion	26
Chapitre 3 : Résultats et Observations	28
1. Matériels et méthodes	29
1.1. Langage de programmation	29
1.2. Matériel	29
1.3. Méthodes	30
2. Fonctions tests	30
2.1. La fonction Goldstein-Price	30
2.2. La Fonction Rosenbrock	30
2.3. La fonction six-hump Camel	31
2.4. La fonction de Rastrigin	32
2.5. La fonction Griewank	33
2.6. La fonction shekel	34
2.7. La fonction de Hartman 3D	34
3. Observation et discussion	35
4. Application au modèle hydrologique GR2M	36
4.1. Présentation du modèle GR2M	36
5. Application	38
6. Conclusion	39
Conclusion Finale	40
Bibliographie	42

Liste des tableaux :

Tableau 3-1: résumé de différentes caractéristiques des fonctions tests

Tableau 3-2: résumé de différents résultats pour les fonctions tests

Tableau 3-3: résumé de différentes caractéristiques des fonctions tests

Liste des figures :

Figure 1-1 : organigramme représentant les différentes techniques d'optimisation global

Figure 1-2 : schéma représentant un espace de recherche incluant des minimums et des maximums globaux et locaux

Figure 2-1 : organigramme de l'algorithme PSO

Figure 2-2 : schéma représentative de l'oscillation d'une particule lors l'optimisation avec l'algorithme PSO

Figure 2-3 : organigramme du fonctionnement de l'algorithme IMOS

Figure 3-1:performances de la machine de calcul

Figure 3-2: représentation de La fonction Goldstein-Price

Figure 3-3: représentation de La fonction Rosenbrock

Figure 3-4: représentation de La fonction six-hump Camel

Figure 3-5: représentation de La fonction Rastrigin

Figure 3-6: représentation de La fonction Griewank

Figure 3-7: schéma représentatif du fonctionnement du modèle GR2M

Introduction Générale

Introduction générale :

L'optimisation est une branche très importante en mathématique, cette branche a permis des avancées fulgurantes dans tous types de sciences, ses principes et les méthodes avec lesquelles elle travaillent sont assez différentes et pourtant ont servis à résoudre presque le même type de problème, prouvant ainsi que plusieurs approches peuvent permettre de résoudre le même type de problème.

Dans cette étude nous allons utiliser les avantages des algorithmes d'optimisation locale qui sont pour la plupart basée sur des techniques mathématiques analytiques, et aussi les avantages des méthodes d'optimisation globale qui pour les plus performant et les plus utilisées sont des algorithmes tirés de la nature. Dans notre cas nous avons pris du fonctionnement de l'algorithme Particle Swarm Optimization, et aussi des techniques d'optimisation non linéaire avec contraintes, l'algorithme tiré de la fusion de ces derniers sera testé avec des fonctions tests pour permettre d'évaluer ses performances, et ses défauts.

Chapitre I : Introduction à l'optimisation

1. Problème d'optimisation

L'optimisation en hydrologie est une technique adaptée pour effectuer le calage automatique. Ce calage automatique n'est d'autre que des algorithmes à appliquer aux critères de validation des modèles hydrologiques afin de trouver le meilleur jeu de paramètres permettant une simulation très proche de la réalité des débits. Dans le sens général, un problème d'optimisation est une situation qui nécessite une maximisation ou une minimisation d'une fonction objective représentant une information sur le phénomène étudié (Moulahoum, 2016)

Le problème d'optimisation en hydrologie peut être défini de la manière suivante :

Le problème de calage d'un modèle pluie-débit à une sortie (Débit) peut aussi être défini comme un problème d'optimisation multi-objective. Il s'agit alors de minimiser les distances entre les valeurs observées et simulées des débits considérant plusieurs pas de temps (horaire, journalier, mensuel,...). Ce problème se pose comme suit (Moulahoum, 2016)

$$Min \{D_i(S_{oi}, S_{ci}), i = 1..n\}$$

Avec

D_i : fonction calculant la distance entre les débits observés et simulés au niveau de pas du temps i ($i = 1..n$)

S_{oi} : le débit observé du modèle au pas de temps i

S_{ci} : le débit calculé du modèle au pas de temps i

n : nombre de pas de temps

Ce problème est souvent réduit à un problème mono-objectif, en choisissant à optimiser une fonction calculant la distance *globale* entre les deux séries observées et calculées appelée critère de performance du modèle ou fonction objective FO. Le problème d'optimisation s'écrit alors de la manière suivante :

$$Min \{FO(S_{oi}, S_{ci}), i = 1..n\}$$

Min FO (S_{oi}, S_{ci}), $i=1, n$

Avec

FO : fonction calculant la distance globale entre les débits observés et simulés

S_{oi} : le débit observé du modèle au pas de temps i ($i=1, n$)

S_{ci} : le débit calculé du modèle au pas de temps i

n : taille de l'échantillon

D'autre part, le problème de calage automatique peut se formuler comme un problème bi-objectif ou même à plusieurs objectifs en choisissant d'optimiser simultanément plusieurs critères de performances adoptants des distances différentes entre les séries observées et calculées (Moulaoui, 2016).

La Fonction Objective peut être prise selon l'objectif de l'étude et l'ampleur de la simulation souhaitée. En principe, le calage automatique effectué par des méthodes d'optimisations est très souhaité car il permet de passer moins de temps pour arriver à l'ajustement idéal du modèle aux données d'entrée et de sortie. L'inconvénient de cette approche de calage est le jeu de paramètres nécessaire qui aboutit à des valeurs irraisonnable du fait que les algorithmes utilisés s'ils ne sont pas mis sous contraintes vont converger vers l'optimum global indépendamment de la signification physique des paramètres. Pour cette raison un calage automatique doit toujours être associé à une appréciation (Moulaoui, 2016).

2. Difficulté que représente l'optimisation des modèles pluie-débit :

Plusieurs difficultés se ressentissent lorsque on parle d'optimisation en hydrologie, et encore plus dans le calage des modèles pluie-débit, plusieurs algorithmes ont été testé, plusieurs algorithmes ont joué leur rôles mais il reste encore beaucoup de chemin à faire, les difficultés rencontré sont :

2.1. Interdépendance des paramètres :

Dans l'optimisation les relations mathématiques utilisées ne sont pas bijective, ce qui veut que pour un jeu de paramètres différents on peut avoir la même valeur de la fonction objective, donc on pourra un seul optimum mais pour différent jeu de paramètres, cela s'explique par le fait que les paramètres dépendent les uns des autres et que la variation d'un des paramètres peut être compensé par les autres.



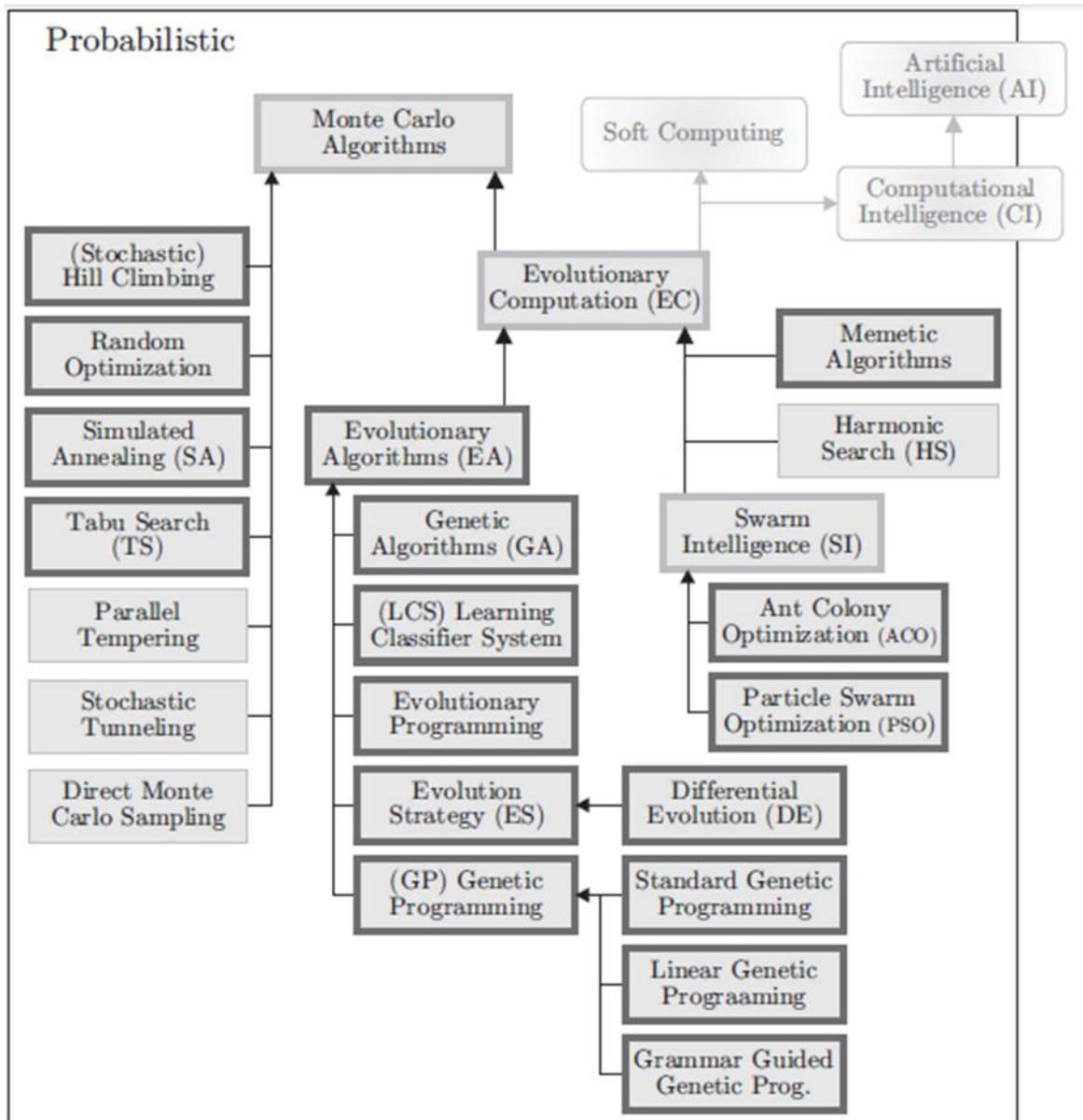


Figure 1-1 : organigramme représentant les différentes techniques d'optimisation global

2.2. Indifférence de la fonction objective face aux paramètres :

On peut parler de l'indifférence ou de l'instabilité de la fonction objectif quand la variation d'un des paramètres à optimiser n'influent pas sur la valeur de la fonction objective, ce phénomène se manifeste par des vallées de la fonction réponse de la fonction objective, cette vallée peut causer une difficulté aux algorithmes d'optimisation qui peuvent s'y trouver piégés. Si la vallée se situe au niveau de l'optimum, ceci entraîne une forte incertitude au niveau de l'estimation de l'optimum surtout dans la direction des paramètres insensibles (Moulahoum, 2016).

2.3. Direction du gradient :

Certains algorithmes utilisent le principe de la descente du gradient, et donc convergent que dans le cas où la fonction est dérivable et continue dans l'espace de recherche.

2.4. les optimums locaux :

Les optimums locaux représente des pics de la fonction objectives mais qui ne représentent pas la valeur optimal de tout l'espace de recherche, ces pics représente des pièges pour les algorithmes d'optimisation, surtout pour les algorithmes d'optimisation local, qui peuvent tomber très facilement dans ce genre de pièges.

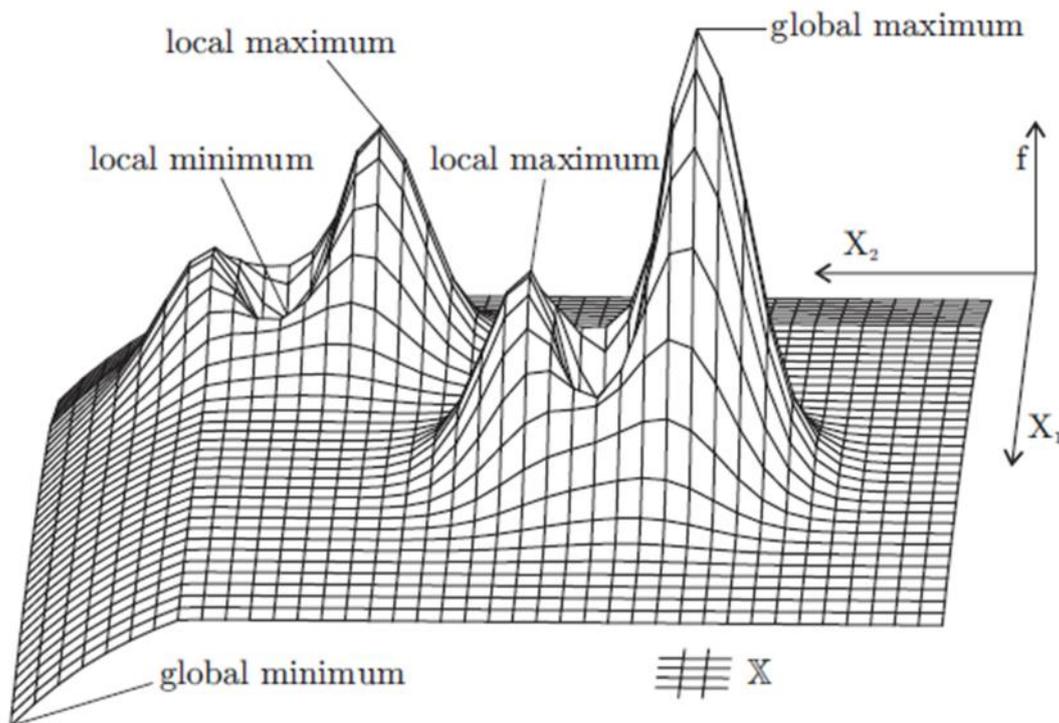


Figure 1-2 : schéma représentant un espace de recherche incluant des minimums et des maximums globaux et locaux

3. Les Algorithmes d'optimisation :

On peut distinguer différentes classifications des algorithmes de recherche de l'optimum. Les algorithmes d'optimisation peuvent être classés par exemple en algorithmes de recherche locale et algorithmes globaux : les premiers algorithmes constituent des suites qui convergent vers la solution optimale et qui n'acceptent que l'amélioration (et non la détérioration) de la fonction objective d'une itération à une autre. Les algorithmes de recherche globale acceptent la détérioration de la solution d'une itération à une autre, ceci dans le cadre d'une stratégie veillant à déjouer les optimums locaux. Les algorithmes d'optimisation globaux acceptent dans leur démarche de recherche du point le plus bas (optimum global) d'escalader des dunes dans le but de voir ce qu'il y a derrière (éventualité d'existence des optimums globaux au-delà). Les algorithmes de recherche locaux sont facilement piégés par les optimums locaux et les résultats

qu'ils donnent dépendent largement des points de départs, comme le cas de Downhill-Simplex (Moulahoum 2016).

Les algorithmes d'optimisation peuvent être classés aussi en tenant compte de la dimension de l'espace de recherche en méthodes unidirectionnelles et méthodes pluridimensionnelles.

Tenant compte de l'usage ou non de simulations stochastiques dans leur recherche, les algorithmes d'optimisation peuvent être classés en méthodes déterministes et méthodes stochastiques (Moulahoum 2016).

Les méthodes locales partent d'un jeu initial de paramètres. Celui-ci constitue le point de départ de la méthode dans l'espace des paramètres. La méthode explore ensuite à chaque itération une direction de cet espace de façon à améliorer systématiquement la valeur de la fonction objectif. Lorsqu'un optimum de la fonction objectif est atteint, les itérations s'arrêtent. Le jeu de paramètres optimaux est alors celui de la dernière itération. Ces méthodes nécessitent un temps de calcul relativement faible comparé aux méthodes globales. L'inconvénient est que si la fonction objective présente plusieurs optima, ces méthodes sont susceptibles de converger vers un optimum local. (Moulahoum 2016).

Les méthodes globales permettent d'explorer une plus grande partie de l'espace des paramètres en l'échantillonnant de façon régulière ou aléatoire. Grâce à cet échantillonnage, elles ne dépendent plus du jeu de paramètres initialement choisi ce qui permet en théorie d'éviter les optima locaux. Des études réalisées avec des données synthétiques (où les données et la structure du modèle sont parfaites) semblent montrer que les méthodes globales sont supérieures aux méthodes locales. Les études réalisées avec des données réelles ne semblent pas confirmer cette supériorité. Les méthodes globales semblent aussi dépendantes de l'échantillonnage initial de l'espace des paramètres. Nécessitant un temps de calcul important, ces méthodes globales sont en pratique délaissées au profit de méthodes locales plus rapides. (Moulahoum 2016).

3.1. Les méthodes déterministes

- Les algorithmes de recherche directe : ce sont ceux qui ne nécessitent pas le calcul des dérivées de la fonction objective et se limitent au calcul de cette fonction, tels que la méthode du Downhill-Simplex (Nelder et Mead 1965).
- Les algorithmes de recherche analytique : ce sont ceux qui utilisent les dérivées de la fonction objective tels que la méthode de gradient, du Gradient conjugué.

Les algorithmes de recherche directe sont plus performants en présence de discontinuités de la fonction objective. Cependant les algorithmes de recherche analytique sont plus performants quand le point de départ est proche de l'optimum.

3.2. Les méthodes stochastiques non inspirées de la nature:

Ce sont des méthodes de recherches basées sur l'aléa, on peut citer :

(a) L'algorithme de recherche aléatoire:

Il est également connu sous le nom de Simulation Monte-Carlo. Il s'agit de générer aléatoirement des jeux de paramètres dans l'espace de recherche et de calculer la valeur de la fonction objective. La solution optimale est celle qui obtient la meilleure valeur de la fonction objective.

Cet algorithme peut être considéré comme l'algorithme d'optimisation le plus primitif. En effet il ne fait aucun usage de l'information acquise durant la génération de l'échantillon pour diriger la recherche. Il s'agit ici d'une simple exploration du domaine de recherche sans aucune exploitation de l'information explorée.

Cette recherche peut être faite aléatoirement en se basant sur la loi uniforme (Uniform Random Search), comme elle peut être faite en utilisant d'autres distributions statistiques (Loi Normale par exemple) donnant plus de chance à des sous-domaines déterminés de l'espace de recherche.

(b) Adaptive Random Search :

C'est une technique améliorée de recherche aléatoire qui consiste à guider la recherche dans un domaine cubique qui va en se rétrécissant sur l'optimum.

3.3. Les algorithmes inspirés de la Nature

La Nature sait bien optimiser. C'est en effet en observant la nature ou en retournant à des travaux de recherche sur certains processus naturels que certains chercheurs ont réussi à mettre en place des algorithmes d'optimisation inspirés de certains processus naturels d'optimisation: physique (recuit simulé), biologique (algorithmes évolutifs), ontologique (algorithmes de colonie de fourmis), etc.

(a) Le Recuit simulé :

Il s'agit d'un algorithme d'optimisation inspiré de la métallurgie qui a été développé initialement par Kirkpatrick et al. (1983) et Cerny (1985). Il s'agit d'une technique basée sur l'exploration aléatoire de l'espace d'état (fonction objective), tout en favorisant les descentes,

mais sans rejeter les remontées. Il s'agit autrement d'une chaîne de Markov dans l'espace d'état, les transitions sont acceptées ou rejetées avec une probabilité p selon le cas :

- $P = 1$ si la fonction d'état diminue,
- $P = \exp(-F/T)$ si la fonction d'état augmente.
- Avec T : un paramètre, appelé température
- F : variation de la fonction objective d'une itération par rapport à l'itération précédente

On peut noter ici que plus la température est grande plus les transitions ascendantes sont favorisées. Mais pour des températures nulles cet algorithme devient tout simplement un algorithme de descente. Une bonne pratique est de faire décroître la température ni trop vite (ce qui permet à l'algorithme de mieux explorer le domaine de recherche sans être piégé par un optimum local), ni trop lentement (pour ne pas altérer l'efficacité de l'algorithme).

(b) Les méthodes évolutives :

Ce sont des méthodes basées sur la théorie d'évolution de Darwin (1859). Dans la nature tout individu, faisant partie d'une population, qui s'adapte mieux à son environnement (qui inclut aussi les autres individus) peut vivre plus longtemps. Ainsi, il a plus de temps, autrement dit plus de chance, d'avoir des descendants et donc plus de chance de transmettre ses caractéristiques génétiques d'adaptation à ses descendants. Ceci veut dire aussi que les individus les moins adaptés ont moins de chance d'avoir des descendants vu que leur vie est courte, et donc moins de chances de transmettre leurs caractères (gènes ou chromosomes).

Ce principe de sélection naturelle favorisant le plus fort s'applique bien, aux descendants, d'une génération à une autre. Ainsi la population évolue dans le temps vers des individus de plus en plus adaptés (résistants, forts, compétitifs, ...).

De plus si le hasard joue un rôle en faveur d'un individu né avec un gène muté qui lui offre des nouvelles caractéristiques lui permettant de mieux s'adapter à son environnement, l'individu aura plus de chance de transmettre ces nouvelles caractéristiques aux descendants. Autrement, ce caractère aura plus de chance de se conserver et d'être transmis aux générations futures. Ces générations futures vont donc cumuler dans le temps de plus en plus de bons caractères, et vont être de plus en plus adaptées.

Il s'agit bien d'un processus d'optimisation naturelle, consistant à maximiser dans le temps le caractère d'adaptation d'une population en la faisant évoluer.

Certains chercheurs ont essayé de mettre en place des méthodes d'optimisation servant à résoudre de nombreux problèmes dans la pratique en s'inspirant de ce processus d'optimisation naturel. Il s'agit des méthodes dites évolutives.

Ces méthodes considèrent les points dans l'espace des paramètres à explorer comme des « individus », la fonction objective à optimiser comme « le caractère d'adaptation », en partant d'une population initiale et en la faisant évoluer en donnant plus de chance aux points ayant les meilleures valeurs de la fonction objective, (plus résistants dans le concept naturel) à contribuer dans la création des nouveaux points.

Parmi ces méthodes on peut citer les algorithmes génétiques qui, en plus du principe d'évolution, empruntent à la nature et plus spécifiquement à la génétique certains concepts surtout pour la formation des nouveaux individus et de caractérisation (la mutation, les gènes, les chromosomes, etc.). Les algorithmes génétiques ont réussi à percer beaucoup de problèmes d'optimisation reliés à des domaines divers (industrie, électronique, etc.) et à être parmi les méthodes d'optimisation les plus utilisées. Ils ont trouvé leur chemin pour l'application au calage des modèles pluie-débit.

(i) Les algorithmes Génétiques

Les Algorithmes Génétiques (Goldberg, 1989) sont des méthodes d'optimisation basées sur les principes de la sélection naturelle par analogie avec la théorie darwinienne. Les algorithmes génétiques opèrent sur une population d'individus qui évoluent au cours des générations, grâce à des opérateurs génétiques (croisement, mutation,..) vers un individu optimal, autrement vers la solution du problème d'optimisation. Les individus (autrement jeu de variables, point dans l'espace de recherche...) peuvent aussi être appelés des chromosomes et peuvent aussi être codés en chaînes binaires. Ces algorithmes sont très robustes.

Certains chercheurs ont essayé d'améliorer la performance des algorithmes génétiques en s'inspirant encore de la nature et plus précisément des niches écologiques ou îles. En effet dans la nature les niches écologiques assez isolées évoluent d'une manière assez différente des autres parties du monde ce qui favorise l'apparition d'espèces assez originales. La méthode de niche étend les algorithmes génétiques de base en encourageant la formation de sous-populations stables dans le voisinage des solutions optimales. Elles permettent aux algorithmes génétiques de rechercher plusieurs pics en parallèle et donc d'éviter d'être piégés par les optimums locaux (Parent, 2007).

3.4. Les algorithmes hybrides ou mimétiques:

Les algorithmes mimétiques ou hybrides sont des méthodes qui combinent plusieurs techniques d'optimisation à la fois.

Il y a plusieurs exemples qu'on peut citer :

- Les algorithmes génétiques peuvent être hybridés par des algorithmes de recherche locale. Ceci consiste à remplacer un des mécanismes aléatoires de l'algorithme génétique par une recherche locale (déterministe) afin d'accélérer la convergence. En effet, connus par la lenteur de leur convergence, les algorithmes génétiques peuvent être dopés par des méthodes de recherche locale pour les accélérer, ces algorithmes servent à créer des individus avec des performances meilleures pour les intégrer dans la population afin d'accélérer la convergence. Une approche courante est de coupler un algorithme génétique avec un algorithme de recherche locale. Cette approche a été appliquée en modélisation pluie-débit. En effet la solution optimale trouvée par un algorithme génétique agissant dans un domaine discrétisé est ensuite raffinée avec une méthode d'optimisation locale.

- Le Recuit Simulé Simplex: cet algorithme combine le recuit simulé avec le Downhill-Simplex. Le recuit simulé connu comme algorithme de recherche globale contribue à mieux explorer l'espace de recherche et le simplex connu comme un algorithme de recherche locale contribue à mieux raffiner la solution et à faciliter la convergence.

4. Les algorithmes d'optimisation : de l'exploration à l'exploitation:

Les algorithmes d'optimisation disposent de deux mécanismes principaux servant à la recherche de l'optimum, il s'agit de l'exploration et de l'exploitation.

L'exploration : c'est l'exploration du domaine de recherche. C'est ce mécanisme qui permet à l'algorithme d'optimisation de découvrir l'espace de recherche. Cette tâche est effectuée de manière différente d'un algorithme à un autre, elle est faite d'une manière géométrique pour le cas de Downhill-Simplex (réflexion, contraction,..) ou stochastique pour le cas de Simulated Annealing. C'est d'ailleurs grâce à cette exploration que l'échantillon des points déjà explorés par l'algorithme d'optimisation peut servir comme une base de données pour l'analyse de sensibilité des paramètres du modèle.

L'exploitation: c'est l'exploitation de l'information déjà explorée sur le domaine de recherche afin de choisir une direction de recherche (l'exploitation). C'est ce mécanisme qui rend les algorithmes d'optimisation plus efficaces. Le simplex utilise une technique d'exploitation basée sur la discrimination des jeux de paramètres explorés selon leur valeur de fonction objective en favorisant de se débarrasser des mauvais points (ayant la plus grande valeur de la fonction objective, dans le cas de problème de minimisation) et en orientant la recherche au voisinage

des zones supposées prometteuses (loin du mauvais point). Les algorithmes génétiques utilisent une technique d'exploitation basée sur le principe d'évolution de Darwin favorisant la recherche au voisinage des individus mieux adaptés.

Il est à noter ici que pour pouvoir explorer l'information sur les parties explorées les algorithmes d'optimisation ne gardent en mémoire, au cours de leurs déroulements, qu'une partie limitée de l'information déjà explorée et en perdent les autres parties. La taille de la population de recherche reflète la mémoire de l'algorithme d'optimisation.

L'algorithme d'optimisation restreint généralement cette mémoire à un nombre limité de points (jeux de paramètres avec les valeurs correspondantes de la fonction objective).

L'algorithme d'optimisation n'exploite ainsi, à une itération donnée, qu'un nombre limité de jeu de paramètre exploré, ceci malgré le nombre important de points explorés lors de son déroulement, qui est tellement important qu'il peut même servir à une analyse de sensibilité des paramètres du modèle.

5. Critères de Choix d'un algorithme d'optimisation :

Le choix d'un algorithme d'optimisation pour la résolution d'un problème donné se base notamment sur l'évaluation de sa performance. Ici il faut noter qu'on parle de la performance de l'algorithme d'optimisation et pas de la performance du modèle pluie débit, et ces deux notions sont différentes. Plusieurs auteurs parlent de deux critères déterminants d'évaluation d'un algorithme d'optimisation : l'efficacité et l'efficience, auxquelles on ajoute la robustesse [Duan].

- **Efficacité** : ou « effectiveness » c'est l'aptitude de l'algorithme de trouver la solution optimale, ou autrement son aptitude de déjouer les difficultés citées dans le paragraphe précédent. Notamment, il s'agit de ne pas tomber sur des optimums locaux, de ne pas être piégé par une zone d'insensibilité de la fonction objective, etc.

- **Efficience** : ou « efficiency » c'est la rapidité de l'algorithme de trouver la solution optimale ou dans un autre sens le coût de trouver la solution optimale en terme de temps de calcul. Une mesure de l'efficience peut être faite par le temps CPU d'une machine nécessaire pour trouver la solution optimale. Mais puisque les machines diffèrent de leurs performances, et étant donné que l'évaluation de la fonction objective en un point donné dans l'espace des variables est la partie qui consomme la majorité du temps de calcul dans un algorithme d'optimisation, le

nombre d'évaluations de la fonction objective nécessaire pour atteindre l'optimum peut être considéré comme un indicateur standard de l'efficacité.

- **Robustesse** : la robustesse est une partie de l'efficacité, elle s'illustre par l'aptitude de l'algorithme d'optimisation de trouver la même solution (optimale) en changeant les points de départ ou autrement son insensibilité aux conditions initiales.

6. Conclusion :

Dans le présent chapitre nous avons fait un brève résumé des techniques d'optimisation, leur caractéristiques et leur classification, ceux-ci nous aidera à mieux comprendre ses techniques et de mieux appréhender leur fonctionnement

Chapitre 2 :
Particle Swarm
Optimization et Intelligent
Mesh for Optimum Search

Dans le présent chapitre nous allons présenter l'algorithme Particle Swarm Optimization (PSO), dont on s'est inspiré pour concevoir l'algorithme Intelligent Mesh for Optimum Search (IMOS), utilisant ainsi les mêmes principes de recherche que le PSO mais avec une vision différente, et en l'améliorant sa capacité en le fusionnant avec un algorithmes de recherche local.

1. Particles swarm optimization (Optimisation par essaim de particules) :

Faisant partie d'une branche de l'optimisation connue sous le nom de swarm intelligence, et constituant le premier algorithme conçu, cette algorithme a prouvé son efficacité quand a la recherche de l'optimum globale ; elle est Basée sur le principe de la descente stochastique tout comme les algorithmes génétiques.

Cette méthode est inspiré de la nature, plus précisément du comportement des oiseaux dans la période migratoire, qui doivent parcourir de longues distances et donc doivent optimiser leur déplacement en termes d'énergie dépensée.

Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum, Chacune de ces particules est dotée (bombrun) :

- D'une position, c'est-à-dire ses coordonnées dans l'ensemble de définition.
- D'une vitesse qui permet à la particule de se déplacer. De cette façon, au cours des itérations, chaque particule change de position. Elle évolue en fonction de son meilleur voisin, de sa meilleure position, et de sa position précédente. C'est cette évolution qui permet de tomber sur une particule optimale.
- D'un voisinage, c'est-à-dire un ensemble de particules qui interagissent directement sur la particule, en particulier celle qui a le meilleur critère.

A tout instant, chaque particule connaît :

- Sa meilleure position visitée. On retient essentiellement la valeur du critère calculée ainsi que ses coordonnées.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnancement optimal.
- La valeur qu'elle donne à la fonction objectif car à chaque itération il faut une comparaison entre la valeur du critère donnée par la particule courante et la valeur optimale.

On se rend compte que l'évolution d'une particule est finalement une combinaison de trois types de comportements : égoïste (suivre sa voie suivant sa vitesse actuelle), conservateur (revenir en arrière en prenant en compte sa meilleure performance) et panurgien (suivre aveuglement le meilleur de tous en considérant sa performance). On voit alors que la bio-inspiration à l'origine de l'optimisation par essaim particulaire ressort dans l'algorithme sous la forme d'une intelligence collective : coordination du groupe, instinct individuel et interaction locale entre les individus (grognements, phéromones...)(bombrun).

On observe donc un compromis psycho-social entre confiance en soi et influence des relations sociales (bombrun).

Nous allons présenter le fonctionnement de l'algorithme de base :

- a. Initialisation des paramètres de chaque particules, position des particules dans l'espace de recherche, vitesse d'avancement, et on prendra comme meilleur position connue sa position initiale et évaluation de la FO pour chaque particules.
- b. Pour chaque particule, on tira aléatoirement deux paramètres c_2 et c_3 et on met à jour la vitesse de la particule grâce à la formule suivante :

$$V_{k+1} = c_1 * V_k + c_2 * (best_{particule} - position_{particule}) + c_3 * (best_{voisin} - position_{particule})$$

Puis on met à jour la position de chaque particule grâce à la formule suivante :

$$X_{k+1} = X_k + V_{k+1}$$

- c. On compare les valeurs de FO de chaque particules leur voisinage et on procède à la mise à jour de la façon suivante :
 - Si la valeur de la FO de la particule est meilleure que celle de son voisinage alors elle est choisie, sinon on prendra celle de son voisinage.
 - On compare la valeur retenue précédemment avec la meilleure position enregistrée dans son historique par la particule de la même manière que précédemment, et on retient la position qui a la valeur de FO comme étant la meilleure position enregistrée par la particule.
- d. Si les critères d'arrêt sont vérifié alors refait les étapes (b) et (c) sinon arrêter l'algorithme.

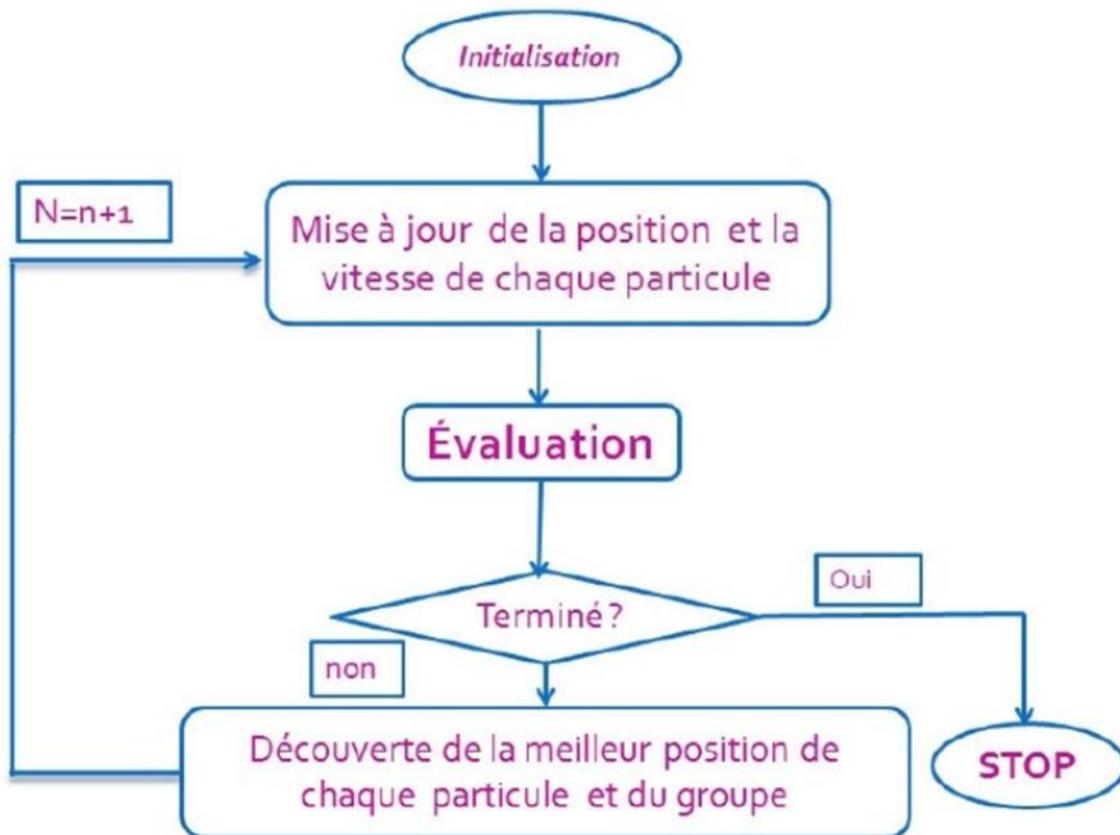


Figure 2-1 : organigramme de l'algorithme PSO

1.1. Caractéristiques de l'algorithme :

1.1.1. Topologie :

Par topologie on parle du groupe de particules qui représente le voisinage d'une particule, le choix de la topologie influe grandement sur le partage de l'information, c'est ce mécanisme qui permet d'éviter de tomber dans des optimums locaux. Plusieurs types de topologies sont existants mais la plus connue et la plus simple est la topologie en cercle, dans ce type de voisinage la particule partage l'information avec les deux particules les plus proche d'elles.

1.1.2. Choix des paramètres :

Le choix des paramètres c_1, c_2 et c_3 sont très important dans la convergence de l'algorithme, si les paramètres sont mal choisis il peuvent conduire à une divergence de l'algorithme ou bien conduire l'algorithme à une convergence très rapide qui peut le conduire à tomber dans le piège des minima locaux. Le choix de ces paramètres font jusqu'à jour l'objet de recherche.

Une solution proposé pour augmenter l'efficacité de ces paramètres est de procéder à une méta-optimisation, c'est-à-dire optimiser les paramètres de l'algorithme grâce un autre algorithme d'optimisation.

1.1.3. Phénomènes d'oscillations :

Plus communément appelée « two step forwards, one step back », est un phénomène qui conduit l'algorithme à ne pas converger, cela s'explique par le fait que dans la formule de mise à jour de la vitesse, celle-ci se calcul par une sommation algébrique de distance, donc si dans une étape j la distance entre la meilleur valeur enregistré dans l'historique et la position actuelle est supérieur à celle entre la position actuelle et la meilleur valeur dans le voisinage alors il s'approchera plus de la meilleur valeur enregistré, mais si dans l'itération $j+1$ c'est le contraire qui se produit alors la particule va se diriger vers le meilleur valeur de son voisinage, ce qui conduit la particules à faire des va et viens et donc à osciller.

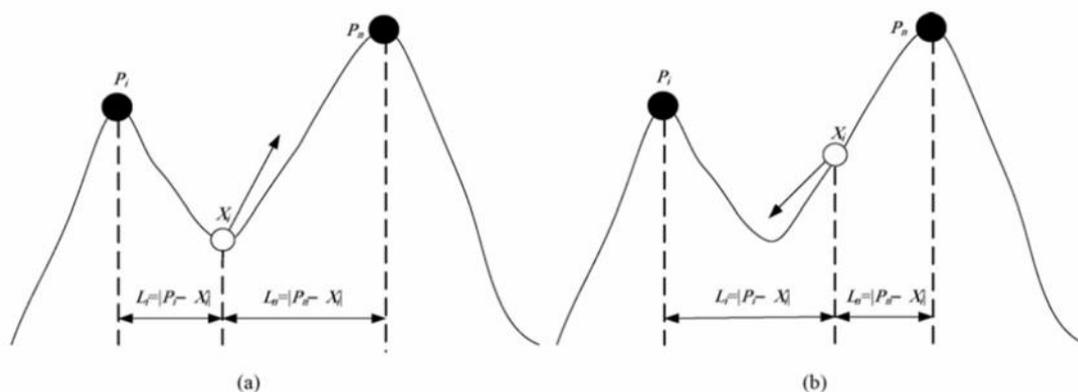


Figure 2-2 : schéma représentative de l'oscillation d'une particule lors l'optimisation avec l'algorithme PSO

2. Intelligent Mesh for Optimum Search (IMOS) :

L'idée de base de cet algorithme est de pouvoir considéré les particules non pas comme des entité ponctuelles, c'est à dire des points dans un espace de recherches mais comme des cellules qui délimite une partie de l'espace de recherche, utilisant le principe de partage de l'information et la communication des cellules entre elles comme dans le PSO, cet algorithme permet ainsi d'augmenter les chances de trouver l'optimum globale.

Chaque cellule ne peut trouver l'optimum qui est contenu dans son propre espace délimité, alors on utilisera ainsi un algorithme de recherche local, jugé ainsi très rapide, le temps de calcul est amoindrie, mais rien n'interdit l'utilisation d'algorithme d'optimisation globale.

La convergence se fait par partage de l'information entre chaque cellule, chacune des cellules ayant ainsi un optimum dans leur espace de recherche, on pourra ainsi comparer les différents optimums trouvés et permettre la convergence vers l'optimum globale.

- 1) Dans notre cas on va discrétiser notre espace de recherche de manière uniforme avec des pas constant pour chaque dimension de l'espace de recherche, on notera ce pas d_i , le nombre de cellules est calculé de la façon suivante :

$$C = div^N$$

C : nombre de cellules,

div : nombre de division,

N : nombre de dimensions de l'espace de recherche,

d_i : pas de chaque dimensions,

i = identifiant de la dimension allant de 1,.....,N

on pourra identifier l'espace de recherche en sachant les limites maximums et les limites minimums de chaque cellule c_i notée [max_i , min_i], ces limites sont par la suite utilisées par l'algorithme de recherche local.

- 2) On procédera par une initialisation des valeurs initiale pour l'algorithme d'optimisation utilisées pour trouver l'optimum local de chaque cellule.
- 3) Après avoir trouvée l'optimum de chaque cellule, on comparera ceux-ci en utilisant un voisinage en anneau, ou on comparera la cellule c_i avec c_{i+1} et c_{i-1} , et on pourra mettre à jour les valeurs des optimums de chaque cellule.
- 4) Après on fera une mise à jour des limites de chaque cellule, et cela grâce à la formule suivante :

$$V_i = c_2 * (bestlim_{i\ ceullule(min\ ou\ max)} - lim_{i\ ceullule(min\ ou\ max)}) + c_3 * (bestlim_{i\ voisinage(min\ ou\ max)} - lim_{i\ ceullule(min\ ou\ max)})$$

$bestlim_{i\ ceullule(min\ ou\ max)}$: Meilleur position des limites enregistrées par la particule.

$lim_{i\ ceullule(min\ ou\ max)}$: Position actuelle des limites de la cellule.

$bestlim_i$ voisinage (min ou max) : Meilleure position des limites du voisinage de la cellule.

- 5) On procédera à la mise jour des positions limite des cellules grâce à la formule suivante :

$$c_i = c_i + V_i$$

- 6) On répétera ces étapes jusqu'à atteindre le nombre d'itérations maximum, ou bien vérifier la condition : $|optimum_t - optimum_{t+1}| < \varepsilon$.

Dans notre cas nous avons pris comme valeur des paramètres $c_2 = c_3 = 0.5$, ces valeurs peuvent être modifiées et en augmentant ces valeurs tout en gardant $c_2 = c_3$ la vitesse de convergence augmente mais leur augmentation peut conduire à tomber dans un optimum local, ce qui n'est pas le but de l'algorithme.

3. Conclusion :

L'algorithme IMOS représente une amélioration de l'algorithme PSO, et aussi une nouvelle vision de l'algorithme, gardant les aspects essentiels de l'algorithme de base avec des modifications qui ont pu diminuer ainsi les défauts de l'algorithme de base.

Cet algorithme prouvera son efficacité dans le prochain chapitre avec quelques fonctions test.

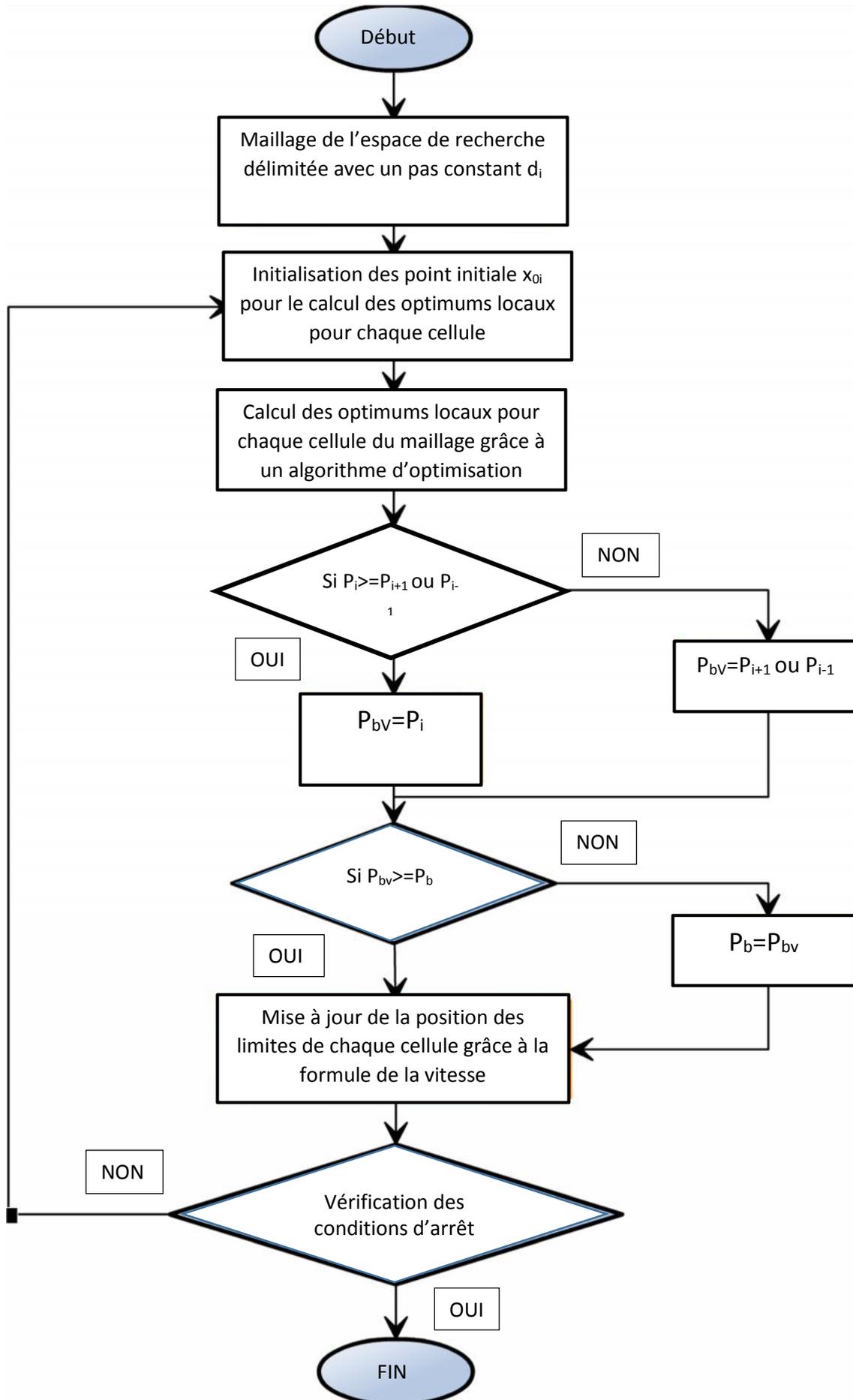


Figure 2-3 : organigramme du fonctionnement de l'algorithme IMOS

Chapitre 3 :
Résultats et
Observations

Dans le présent chapitre nous allons pouvoir tester les capacités de l'algorithme IMOS, et pour cela nous allons utiliser des fonctions test habituellement utilisée pour tester les capacités des algorithmes d'optimisation global, celle présenté ici ont été utilisées pour tester l'algorithme Shuffled Complex Evolution- University of Arizona (SCE-UA), il existe ainsi plusieurs fonctions test mais nous ne allons pas toutes les testées.

1. Matériels et méthodes :

1.1. Langage de programmation :

Matlab est à la fois un logiciel de calcul et un langage de programmation haut niveau. C'est un logiciel payant, dont il existe deux équivalents gratuits

– Octave est un logiciel qui utilise le langage de Matlab et peut donc utiliser les fonctions écrites en Matlab.

-Scilab est développé par l'INRIA et la syntaxe diffère un peu de celle de Matlab mais l'esprit est le même.

Le nom Matlab vient de Matrix Laboratory. En Matlab les objets sont tous par défaut des matrices. Une variable réelle est donc vue par Matlab comme une matrice 1×1 . Le produit est donc par défaut un produit matriciel.

Nous avons choisi le langage MATLAB pour sa large panoplie de programme prêt à l'emploi, nous avons ainsi pu utiliser plusieurs fonctions préprogrammer qui ont pu faciliter l'écriture du programme.

On a notamment utilisée la fonction *fmincon*, qui est une fonction d'optimisation local non linéaire avec ou sans contraintes, celle-ci utilisées pour trouver les optimum locaux propre à chaque cellule.

1.2. Matériel :

Dans notre cas nous avons utilisées une machine assez peu puissante, ses performances sont décrites dans l'image ci-dessous :

Système	
Évaluation :	 4,6 Indice de performance Windows
Processeur :	Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz 2.40 GHz
Mémoire installée (RAM) :	6,00 Go
Type du système :	Système d'exploitation 64 bits
Stylet et fonction tactile :	La fonctionnalité de saisie tactile ou avec un stylet n'est pas disponible sur cet écran
Paramètres de nom d'ordinateur, de domaine et de groupe de travail	
Nom de l'ordinateur :	rafik-PC
Nom complet :	rafik-PC
Description de l'ordinateur :	
Groupe de travail :	WORKGROUP

Figure 3-1:performances de la machine de calcul

1.3. Méthodes :

Pour pouvoir tester si la méthode IMOS est bien assez puissante, on la confrontera à des fonctions tests, et on verra son comportement au fil des calculs.

2. Fonctions tests :

2.1. La fonction Goldstein-Price :

$$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

Fonction a 2 dimensions, elle comporte plusieurs minima locaux, et elle est usuellement testé sur un espace de $[-2, 2]$ pour $i=1,2$.

Le minimum global est $f(x)=3$ a $x=(0,-1)$.

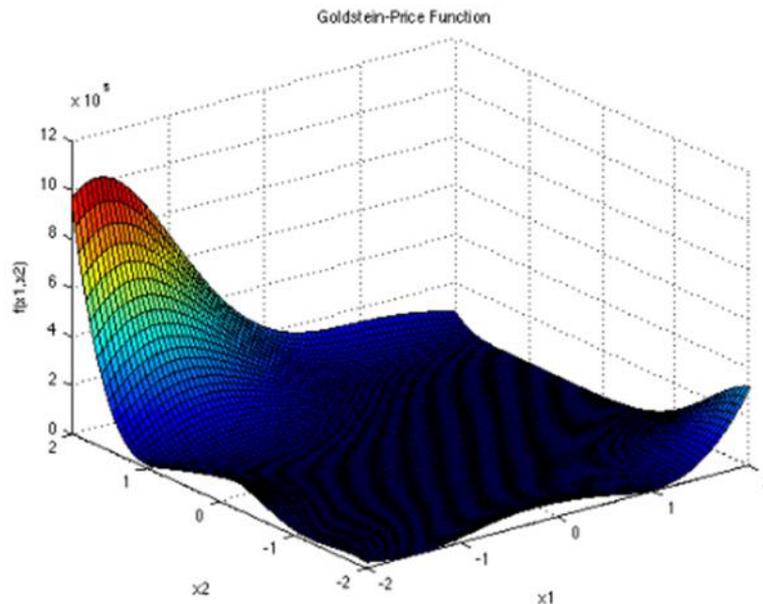


Figure 3-2: représentation de La fonction Goldstein-Price

2.2. La Fonction Rosenbrock :

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

La fonction Rosenbrock, est un problème de test populaire pour les algorithmes d'optimisation basés sur le gradient. Il est montré dans l'image ci-dessous sous sa forme bidimensionnelle.

La fonction est unimodale, et le minimum global réside dans une vallée étroite et parabolique. Cependant, même si cette vallée est facile à trouver, la convergence au minimum est difficile.

La fonction est habituellement évalué sur l'espace $x_i \in [-5 \ 10]$ pour $i=1 \dots d$.

Le minimum globale est de $f(x)=0$ quand $x=(1 \dots 1)$.

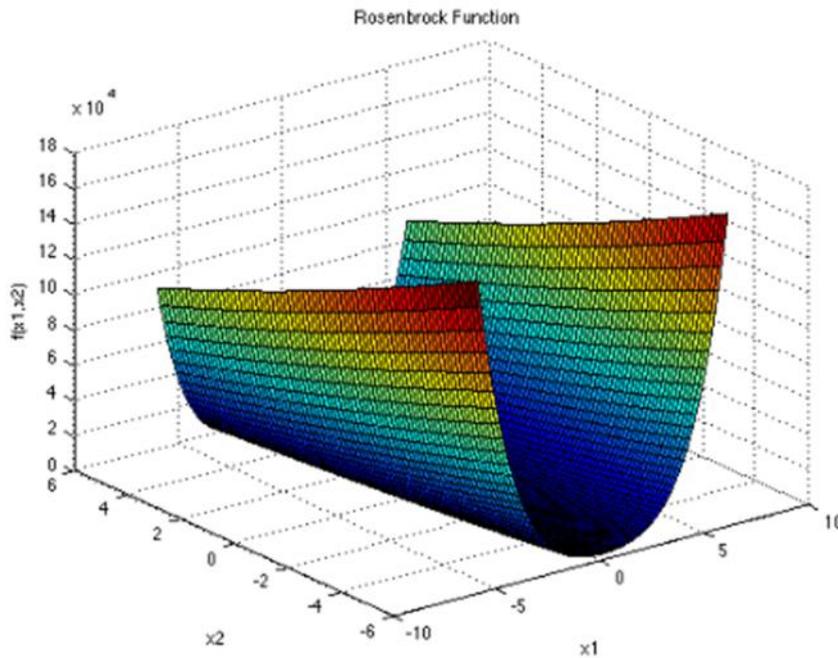


Figure 3-3: représentation de La fonction Rosenbrock

2.3. La fonction six-hump Camel :

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

L'image à gauche montre la fonction Camel à six bosses sur son domaine d'entrée recommandé, et l'image à droite ne montre qu'une partie de ce domaine, afin de faciliter la visualisation des caractéristiques clés de la fonction. La fonction a six minima locaux, dont deux sont globaux.

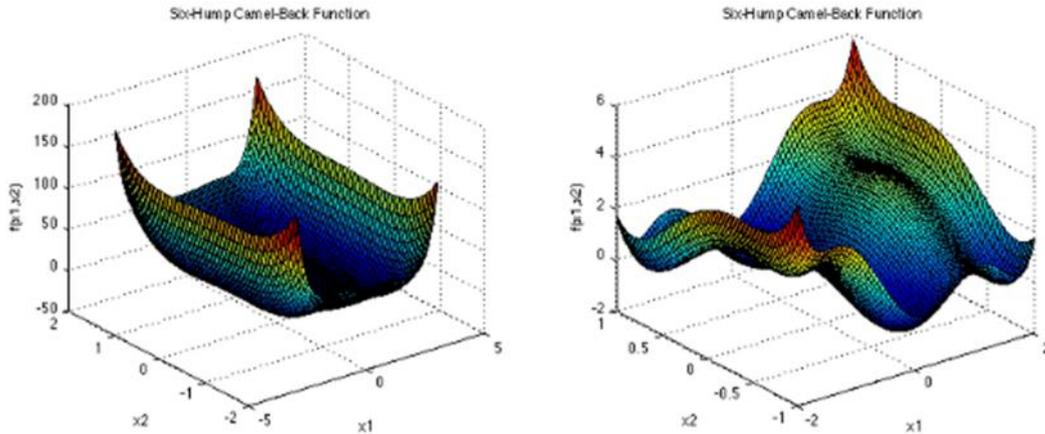


Figure 3-4: représentation de La fonction six-hump Camel

La fonction est habituellement évalué sur l'espace $x_1 \in [-3, 3]$ $x_2 \in [-2, 2]$, et son minima globale est $f(x)=-1.0316$ a $x=(0.0898, -0.7126)$ ou $x=(-0.0898, 0.7126)$.

2.4. La fonction de Rastrigin :

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

De dimension d , La fonction Rastrigin a plusieurs minima locaux. elle est multimodal, mais les emplacements des minima sont régulièrement distribués. Il est montré dans l'image ci-dessus sous sa forme bidimensionnelle.

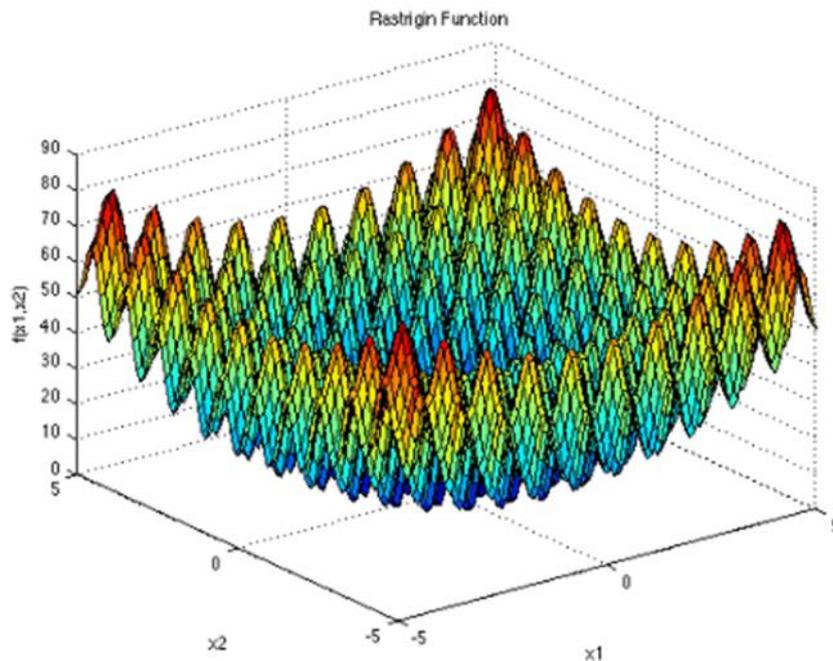


Figure 3-5: représentation de La fonction Rastrigin

Elle est habituellement évalué sur l'espace $x_i \in [-5.12 \ 5.12]$ avec $i=1 \dots d$.

Le minima globale de la fonction est $f(x)=0$ pour $x=(0 \dots 0)$.

2.5. La fonction Griewank :

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

De dimension d, La fonction Griewank a de nombreux minima locaux répandus, qui sont régulièrement distribués. La complexité est affichée sur les images ci-dessous.

La fonction est habituellement évalué sur l'hypercube $x_i \in [-600 \ 600]$ pour $i=1 \dots d$.

Son optimum global est $f(x)=0$ quand $x=(0 \dots 0)$.

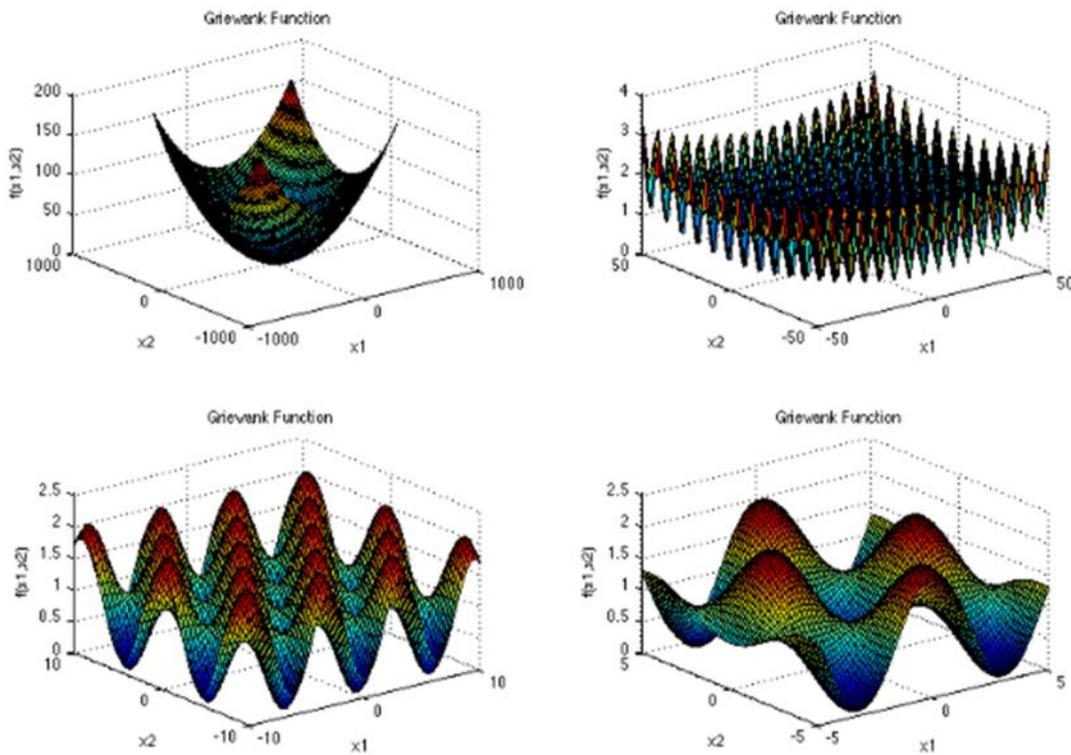


Figure 3-6: représentation de La fonction Griewank

2.6. La fonction shekel :

$$f(\mathbf{x}) = - \sum_{i=1}^m \left(\sum_{j=1}^4 (x_j - C_{ji})^2 + \beta_i \right)^{-1}, \text{ where}$$

$$m = 10$$

$$\beta = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)^T$$

$$\mathbf{C} = \begin{pmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{pmatrix}$$

De dimension 4, elle a 'm' nombre de minima locaux, est un vecteur de taille m , et C est une matrice de taille 4 x m .

La fonction est généralement évalué sur le domaine $x_i \in [0 10]$ avec $i=1,2,3,4$.

Le minimum global varie selon la valeur de m :

Pour $m=4$ $f(x)=-10.1532$ pour $x=(4,4,4,4)$

Pour $m=7$ $f(x)=-10.4029$ pour $x=(4,4,4,4)$

Pour $m=10$ $f(x)=-10.5364$ pour $x=(4,4,4,4)$.

2.7. La fonction de Hartman 3D :

$$f(\mathbf{x}) = - \sum_{i=1}^4 \alpha_i \exp \left(- \sum_{j=1}^3 A_{ij} (x_j - P_{ij})^2 \right), \text{ where}$$

$$\alpha = (1.0, 1.2, 3.0, 3.2)^T$$

$$\mathbf{A} = \begin{pmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}$$

$$\mathbf{P} = 10^{-4} \begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix}$$

Cette fonction peut se définir sur plusieurs dimensions, nous avons opté pour la fonction Hartman de dimension 3 pour moins consommer en temps de calcul, cette fonction contient 4 minima locaux.

Cette fonction est habituellement évaluée sur le domaine $x_i \in [0, 1]$ pour $i=1,2,3$.

Le minimum globale est $f(x)=-3.68278$ quand $x=(0.114614,0.555649,0.852547)$.

Toutes ces fonctions présentées vont permettre de tester le programme et voir s’il pourra ainsi détecter tous les optima globaux, le tableau récapitulatif ci-dessous résume les principales caractéristiques des fonctions tests.

Tableau 3-1: résumé de différentes caractéristiques des fonctions tests

nom	dimension	domaine	minimum globale	fming
Goldstein-Price		2 [-2 2]x[-2 2]	(0 -1)	3
Rosenbrock		2 [-5 10]x[-5 10]	(1 1)	0
six-hump camel		2 [-3 3]x[-2 2]	(0.0898 -0.7126) ou (-0.0898 0.7126)	-1,0316
Rastrigin		2 [-5.12 5.12] x[-5.12 5.12]	(0 0)	0
Griewank		2 [-600 600]x[-600 600]	(0 0)	0
Shekel		2 [0 10]x[0 10]	(4 4)	-11,0298
Hartman-3D		3 [0 1]x[0 1]x[0 1]	(0.114614,0.555649,0.852547).	-3,68278

3. Observation et discussion :

Nous avons mis comme paramètre du programme le nombre de subdivision de chaque domaine et aussi le nombre d’itérations maximal, nous avons pris comme nombre de subdivision étant égale à 10 et un nombre d’itération maximal de 10 pour les toutes les fonctions sauf pour la fonction de Griewank qui est de 20, cela pour observer si il n’y a pas un phénomène de divergence.

Tableau 3-2: résumé de différents résultats pour les fonctions tests

fonction	optimums trouvé	nombre d'itération	minimum globale
Goldstein-Price	(0 -1)		1 (0 -1)
Rosenbrock	(1 1)		1 (1 1)
six-hump camel	(-0,0898 0.7125)		1 (0.0898 -0.7126) ou (-0.0898 0.7126)
Rastrigin	(0 0)		1 (0 0)
Griewank	(0 0)		7 (0 0)
Shekel	(4,0027 4,0021)		1 (4 4)
Hartman-3D	(0.1146 0.5556 0.8525)		3 (0.114614,0.555649,0.852547).

On remarque que l’algorithme arrive très facilement à trouver tous les minima globaux de toutes les fonctions, le nombre maximal d’itération ne dépassant pas 7, et donc sa vitesse de convergence pour ses fonctions est assez élevée.

On remarque aussi une augmentation assez considérable du temps de calcul quand on augmente d’une dimension, le nombre de cellule ayant augmenté de 100 à 1000 cellules, et cela reste un désavantage pour cet algorithme dans le cas d’optimisation de fonction à plusieurs paramètres.

On pourra aussi changer l’algorithme d’optimisation pour trouver chaque optimum de chaque cellule pour voir le meilleur algorithme, cette technique reste assez maniable et pourra faire l’objet de plusieurs changements.

L'algorithme a aussi plus de chance de trouver l'optimum plus rapidement si le maillage est plus fin, mais en contre parti augmentera le temps de calcul.

4. Application au modèle hydrologique GR2M :

4.1. Présentation du modèle GR2M :

Le modèle GR2M (modèle du Génie Rural à 2 paramètres Mensuel) est un modèle pluie-débit global à deux paramètres. Son développement a été initié au Cemagref à la fin des années 1980, avec des objectifs d'applications dans le domaine des ressources en eau et des étiages(Perrin).

Sa structure, bien qu'empirique, l'apparente à des modèles conceptuels à réservoirs, avec une procédure de suivi de l'état d'humidité du bassin qui semble être le meilleur moyen de tenir compte des conditions antérieures et d'assurer un fonctionnement en continu du modèle. Sa structure associe un réservoir de production et un réservoir de routage ainsi qu'une ouverture sur l'extérieur autre que le milieu atmosphérique. Ces trois fonctions permettent de simuler le comportement hydrologique du bassin(Perrin).

Les équations qui régissent le modèle sont les suivantes:

- Production :

La fonction de production du modèle repose sur un réservoir de suivi d'humidité du sol. Une partie P_s de la pluie P_k va être ajoutée au contenu S dans le réservoir en début de pas de temps :

$$P_s = \frac{X_1 \left(1 - \left(\frac{S_k}{X_1} \right)^2 \right) \cdot \tanh \left(\frac{P_k}{X_1} \right)}{1 + \frac{S}{X_1} \cdot \tanh \left(\frac{P_k}{X_1} \right)}$$

Le paramètre X_1 , capacité du réservoir, est positif et exprimé en mm. La pluie en excès, P_1 est donnée par :

$$P_1 = P - P_s'$$

Et le contenu du réservoir est actualisé :

$$S' = S_k + P_s$$

Du fait de l'évapotranspiration, une quantité E_s est prélevée du réservoir :

$$E_s = \frac{S' \left(2 - \frac{S'}{X_1} \right) \cdot \tanh \left(\frac{E}{X_1} \right)}{1 + \left(1 - \frac{S'}{X_1} \right) \cdot \tanh \left(\frac{E}{X_1} \right)}$$

E est l'évapotranspiration potentielle moyenne du mois calendaire considéré. Le niveau S' devient S'' :

$$S'' = S' - E_s$$

• **Percolation :**

Le réservoir de suivi d'humidité du sol se vidange ensuite selon une percolation P2 :

$$P_2 = S'' \left\{ 1 - \left[1 + \left(\frac{S''}{X_1} \right)^3 \right]^{-1/3} \right\}$$

Et son niveau Sk+1, prêt pour les calculs du mois suivant, est alors donné par :

$$S_{k+1} = S'' - P_2$$

• **Routage et échange avec l'extérieur non atmosphérique**

La quantité d'eau totale P3 qui atteint le réservoir de routage est donnée par :

$$P_3 = P_1 + P_2$$

Le niveau Rk dans le réservoir devient alors R' :

$$R' = R_k + P_3$$

Un terme d'échange en eau souterrain F a été imposé par les données des nombreux bassins utilisés. Ignorer cette ouverture sur l'extérieur non atmosphérique conduit à une baisse considérable de l'efficacité du modèle. F est alors calculé par :

$$F = (X_2 - 1).R'$$

Le paramètre X2 est positif et adimensionnel. Le niveau dans le réservoir devient :

$$R'' = X_2.R'$$

Selon l'équation suivante:

$$Q_k = \frac{R''^2}{R'' + 60}$$

Le contenu du réservoir est enfin actualisé par :

$$R_{k+1} = R'' - Q_k$$

• **Paramètres :**

Le modèle a deux paramètres optimisables :

X1: capacité du réservoir de production (mm)

X2: coefficient d'échanges souterrains (-)

Sur un large échantillon de bassins versants, on obtient les valeurs données dans le Tableau ci-dessous :

Tableau 3-3: résumé de différentes caractéristiques des fonctions tests

Paramètre	Médiane	Intervalle de confiance à 90%
X1 (mm)	380	140 -- 2640
X2 (-)	0.92	0.21 -- 1.31

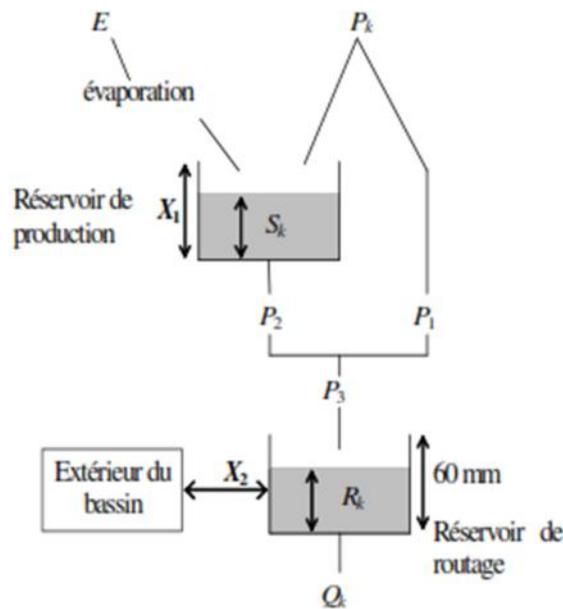


Figure 3-7: schéma représentatif du fonctionnement du modèle GR2M

5. Application :

Dans notre cas nous allons essayer de trouver les deux paramètres du modèle GR2M, en utilisant comme fonction objectif le critère de Nash-Sutcliff décrit ci-dessous :

$$E = 1 - \frac{\sum_{t=1}^T (Q_m^t - Q_o^t)^2}{\sum_{t=1}^T (\overline{Q_o} - Q_o^t)^2}$$

Q_m : débit simulés ; Q_o : débit observés ; $\overline{Q_o}$: moyenne des débit observées

Les données utilisées sont ceux du centre de recherche CEMAGREF, nous allons les utiliser comme étalon et voir si notre programme pourra trouver l'optimum.

Nous avons donné les valeurs suivantes au modèle et à l'algorithme :

S_{ini} : 325.35 mm

R_{ini} : 30 mm

nombre d'itération: 20/

nombre de division de l'espace: 10.

Les résultats trouvés sont les suivants :

```

ptopt =
    640.0000    0.7999
    640.0000    0.7999
    640.0000    0.7999
    749.5445    0.8145
    737.6820    0.8131
    717.4988    0.8105
    640.0000    0.7999
    726.9512    0.8118
    694.6476    0.8075
    721.0792    0.8110
    737.2037    0.8131
    705.2921    0.8089
    702.7004    0.8086
    764.9997    0.8164
    749.3748    0.8146
    724.3789    0.8114
    726.9013    0.8118
    725.3910    0.8116
    702.5000    0.8086
    726.9140    0.8118

Fopt =

Columns 1 through 9
    -0.7741    -0.7741    -0.7741    -0.7811    -0.7817    -0.7821    -0.7741    -0.7820    -0.7813

Columns 10 through 18
    -0.7821    -0.7817    -0.7818    -0.7817    -0.7798    -0.7811    -0.7820    -0.7820    -0.7820

Columns 19 through 20
    -0.7817    -0.7820

```

On voit bien qu'il atteint une valeur assez élevée et assez proche de celle trouvée par les chercheurs qui est de : $E = 0.83$

Nous avons utilisées une autre version améliorée du PSO pour pouvoir voir si l'algorithme IMOS aurais pu tomber dans un optimum local, les résultats reste les mêmes avec l'autre algorithme et confirme que c'est bien l'optimum local dans cet espace de recherche.

6. Conclusion :

L'algorithme IMOS a pu ainsi passé les tests régulièrement données pour pouvoir tester les algorithme d'optimisation globale, mais reste un algorithme que l'on devra compléter et d'essayer ainsi de corriger ses défauts apparent comme son temps de calcul assez élevée dans le cas où on augmente la dimension de l'espace de recherche, mais cet algorithme a la particularité d'allier les avantages des algorithmes d'optimisation local et globale, dont des recherche pour départager et voir qui des deux types d'algorithme était le plus efficace n'ont aboutie a aucune réponse concrète.

Conclusion
Finale

Conclusion finale :

L'algorithme d'optimisation rassemble les avantages des algorithmes d'optimisation globale et local, et reste très maniable, on pourra facilement tester d'autres algorithmes d'optimisations local ou globale pour voir l'efficacité de la technique, le principe le plus important dans cette technique étant le partage de l'information entre les cellules, ce principe est fondamentale dans la recherche de l'optimum globale dans un espace délimitée, ce principe pourra être utiliser a d'autre taches que celle de l'optimisation.

Cette technique n'est qu'un début et contient d'assez grands défauts qu'il faudra au fur et mesure éliminée ou bien diminuer l'impact de celle-ci, tel que le temps de calcul assez important qui croit exponentiellement avec l'augmentation de la dimension de l'espace de recherche, et aussi une étude sur l'impact des valeurs des paramètres de la vitesse de convergence sur la performance de l'algorithme.

Mais malgré ces défauts il a pu trouver les optimums des différentes fonctions tests, et ainsi prouver son efficacité dans la recherche d'optimums globaux, mais que dans le cas de variables indépendantes, ce qui n'est pas du tout le cas dans le domaine de l'hydrologie, ou les différents paramètres des modèles sont bien souvent interdépendants, et cela cause des problèmes assez fréquemment aux algorithmes d'optimisation.

Enfin cette technique est beaucoup plus intéressante dans le cas de problèmes d'optimisation de type spatiale, et pourra se montrer utile dans le cas de problèmes d'optimisation avec des systèmes d'information géographiques.

Références Bibliographiques :

1. Andréassian V., Michel C., Perrin C., *Modèles hydrologiques du Génie Rural (GR)*, Cemagref, 2007.
2. Perrin C., *Vers une amélioration d'un modèle global pluie-débit au travers d'une approche comparative*, Thèse de doctorat, de l'INPG, Institut National Polytechnique de Grenoble, Octobre 2000.
3. Perrin C., Michel C., Andréassian V., *Modèles hydrologiques du Génie Rural (GR)*, Cemagref, Juin 2007.
4. Duan Q.Y., Gupta V.K., and Sorooshian S., « *Shuffled Complex Evolution Approach for Effective and Efficient Global Minimization* », *Journal of Optimization Theory and Applications*: Vol. 76, No. 3, Mars 1993.
5. Duan Q.Y., Gupta V.K., Sorooshian S., & *Optimal use of the SCE-UA global optimization method for calibrating watershed models*, *Journal of Hydrology*, Vol. 158, pp. 265-284.
6. M.Moulahoum, 2016, *projet de fin d'étude pour l'obtention du diplôme d'ingénieur : Influence des critères d'évaluation sur le calage automatique des modèles conceptuels globaux par le shuffled complexe evolution*.
7. Bombrun, 2011, *Rapport d'ingénieur Projet de 2ème année : L'optimisation par essaim particulière pour des problèmes d'ordonnancement*.
8. AMIRECHE et al 2017- ASCMCES 17-Sharjah-EUA. *Comparative assessment between GR model and tank model for rainfall-runoff analysis using Kalman filter-application to Algerian basins*.