

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

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département Hydraulique

Mémoire de *Master en Hydraulique*

**Etude de Performance de deux
Algorithmes d'Optimisation
Globale : Le Shuffled Complex
Evolution (SCE) et l'Algorithme
Génétique (GA)**

Présenté et soutenu publiquement le 13/06/2016

Par Mr **Ahmed Walid MOULAHOU**

Sous la direction du **Pr Abdelmalek BERMAD**

Composition du Jury :

<i>Président</i>	M. Amar BILEK	Docteur	ENP
<i>Promoteur</i>	M. Abdelmalek BERMAD	Professeur	ENP
<i>Examineur</i>	M. Abdelouaheb LEFKIR	Docteur	ENSTP
<i>Examineur</i>	M. Mohamed AMIRECHE	MAA	U-OEB

Remerciement

C'est avec une certaine émotion et beaucoup de sincérité que je voudrais remercier toutes les personnes ayant soutenu et apprécié mon travail. Il est l'occasion de se remémorer les différentes embûches qu'il a fallu surmonter mais surtout les personnes qui m'ont permis d'en arriver là.

Je tiens à remercier mon encadreur, mon professeur, mon collègue et mon Parent à l'Ecole Polytechnique M. Abdelmalek Bermad pour le soutien qu'il m'a offert et la confiance qu'il m'a témoignée tout au long de ces années, à toujours chercher à faire de mon mieux pour devenir et rester le meilleur. Son contact a d'ailleurs été très enrichissant tant au niveau humain qu'au niveau de mon travail. Il peut être assuré de mon sincère respect et de ma profonde gratitude. J'ai beaucoup appris à ses côtés et je lui adresse toute ma reconnaissance.

Je remerci les enseignants qui transpirent honorablement chaque jour afin de nous fournir les outils techniques et humains pour qu'on puisse reconstruire ce pays réduit en cendre. En particulier M. Berkani pour sa présence vocationnelle en salle de cours malgré sa maladie qui le torturait tous les jours sous nos yeux.

Je tiens également à remercier tous mes amis et collègues de l'équipe du CAP pour leur aide, leur soutien et leur gentillesse. Je souhaite également les remercier pour la confiance dont ils ont fait preuve à mon égard en me laissant m'impliquer au sein de la vie collective de l'équipe. Les rapports humains dont j'ai profité à leur côté ont fait naître de réels liens d'amitiés qui à mes yeux n'ont pas de prix. Qu'ils soient tous assurés de ma plus profonde gratitude et estime.

Je remercie les membres de jury d'avoir pris le temps et la patience d'examiner le présent travail et pour toutes leurs remarques qui aideront à améliorer ma modeste contribution à la bibliothèque scientifique de l'Ecole Nationale Polytechnique.

Je remercie enfin toutes les personnes intéressées par mon travail, en espérant qu'elles puissent trouver dans mon rapport des explications utiles pour leurs propres travaux.

Avant tout, je remercie le bon dieu de m'avoir toujours guidé et aidé à achever ce travail et à lutter pour prospérer dans ce monde.

Le meilleur reste à venir

Dédicaces

A mes très chers parents. Tout au long de mon cursus, ils m'ont toujours soutenu dans mes efforts. Ma mère, avec son amour m'a procuré un nid de sécurité et de sérénité dont j'avais besoin dans les moments de doute. Mon père, une vraie école de la vie, je ne cesse d'apprendre tous les jours avec lui. Il a fait de moi un homme de qualité, aux horizons illimités.

Ils ont su me donner toutes les chances pour réussir. C'est grâce à eux que je suis ce que je suis maintenant. Leur bonté et leur patience m'ont guidé tout au long de ma vie. Ils me redonnent mes ailes lorsque ma vie est souvent trop superficielle. Sans la spiritualité qu'ils m'ont apportée, je serais perdue, sans bouée, traumatisée par les limitations de l'être humain lorsqu'il véhicule la difficulté au quotidien. J'aurais pris le mauvais chemin s'ils ne m'avaient pas tenu par la main et, depuis, chaque jour je m'efforce de leur montrer pourquoi ils ont raison d'avoir foi en moi.

Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de ma plus affectueuse gratitude. Je ne pourrais les remercier assez...que Dieu les protège !

A mon frère, le génie de la famille. Celui qui a toujours répondu à mes demandes d'aide, et m'a toujours encouragé à essayer, m'aventurer et visiter le monde. Hichem, qui a toujours partagé avec moi tout ce qu'il possède. Une source de fierté et de respect.

A ma défunte grand-mère qui a tant voulu être présente pour vivre le jour de ma réussite académique.

A ma Tante Rachida et mes cousines et à toute ma famille que j'aime énormément !

A la personne qui m'a soutenu lorsque je n'avais plus personne à qui me tourner. La personne qui a toujours su dessiner le sourire sur mon visage quand la tristesse régnait sur mon esprit. Celle qui m'offrait des pommes lorsque le monde entier me jetait des pierres. La personne avec laquelle je me vois tracer un avenir certain et serein. A ELLE, l'Amour de ma vie !

A la famille Quist, ma deuxième famille aux Etats Unis, qui m'ont accueilli chez eux et traité comme leur fils. Je vous en suis reconnaissant.

A mes très chers amis : Aghiles Izem, Amine Azazga, Samira Samurai, Sabah « El Khir », Ghinwa G, Adem El Dey, Abdennour Nechnache et à Kevin Chuck.

A tous ceux que j'aime, j'apprécie et je respecte.

Ahmed Walid Moulahoum

ملخص

تستخدم أساليب التحسين ذو البحث الشامل مع نماذج المحاكاة على نطاق واسع في المعايرة الآلية وتخدم بوصفها أدوات مفيدة للبحث عن بدائل فعالة من حيث التكلفة للتدابير البيئية. ومع ذلك، فإن درجة الصعوبة في حل مشكلة التحسين ذو البحث الشامل هي بشكل عام تعتمد على أبعاد المشكلة وخصائص معينة للدالة الهدف. غي هذه الدراسة يتم عرض طريقتين للتحسين المثالي ذو البحث الشامل.

يتم اختبار هذه الخوارزميات عن طريق تشغيل 100 محاكات منية على العشوائية على مشكلات اختبارية مختلفة الصعوبة. وأخيراً، تتم مقارنة أداء الطريقتين التلثيتين مع معايرة تلقائية للنموذج المشترك IHACRES-GR4J لمدة عامين من المعطيات اليومية من الجريان السطحي لمستجمع المياه Canning في غرب أستراليا، وذلك باستخدام معيار Nash كدالة هدف. كان أداء SCE دائماً أفضل مقارنة بنتائج GA. فهو يوفر المزيد من الكفاءة والفعالية في الوصول إلى الحل الأمثل للمعالم الافتراضية للأسلوبين. لذلك يوصى ببرنامج SCE-UA للمعايرة التلقائية للنماذج الهيدرولوجية.

الكلمات الدالة: للمعايرة التلقائية، SCE-UA، GA، التحسين المثالي

Abstract

Global Optimization methods linked with simulation models are widely used for automated calibration and serve as useful tools to search for cost-effective alternatives for environmental management. However, the degree of difficulty in solving a global optimization problem is in general dependent on the dimensionality of the problem and certain characteristics of the objective function. Two global optimization methods are presented in this study. These algorithms are tested by running 100 randomly initiated trials on 6 test problems of differing difficulty. Finally, the performance of the two optimization methods was compared by automatically calibrating an IHACRES-GR4J combined model for two years of daily runoff from the Canning Watershed in West Australia, using the Nash criteria as the objective function. The SCE performance was always better compared to the GA results. It offers more efficiency and effectiveness in reaching the optimal solution for default methods parameters. The SCE-UA method is therefore recommended for automatic model calibration.

Key words: *Automatic calibration, Shuffled Complex Evolution, Genetic Algorithms, Optimization*

Résumé

Les méthodes d'optimisation globale liées aux modèles de simulations sont largement utilisées pour le calage automatique et constituent des outils utiles pour rechercher des solutions efficaces pour une bonne gestion environnementale. Cependant, le degré de difficulté à résoudre un problème d'optimisation globale est généralement dépendant de la dimensionnalité du problème et certaines caractéristiques de la fonction objective. Deux méthodes d'optimisation globale sont présentées dans cette étude. Ces algorithmes sont testés en exécutant 100 essais initiés aléatoirement sur 6 problèmes de test de difficultés différentes. Enfin, les performances des deux méthodes ont été comparées par le calage automatique d'un modèle combiné IHACRES-GR4J pour deux ans de ruissellement journalier du bassin versant de Canning en Australie Occidentale, en prenant le critère de Nash pour la fonction objectif. La performance du SCE a toujours été mieux par rapport aux résultats du GA. Le SCE offre une plus grande efficacité et efficacité pour atteindre la solution optimale et ce pour les paramètres par défaut des deux méthodes. Le programme SCE-UA est donc recommandé pour Le calage automatique du modèle.

Mots clés : *Calage automatique, Shuffled Complex Evolution, Algorithme Génétique, Optimisation*

Sommaire

Remerciement	02
Dédicace	03
Résumés	04
Sommaire	05
Liste des tableaux	07
Liste des figures	08
Introduction générale	10
Chapitre I:	13
Problèmes d'Optimisation et Auto-Calage des Modèles Hydrologique	
1. Problème d'optimisation	14
1.1. Difficultés que présente l'optimisation d'un modèle pluie-débit	15
1.2. Les algorithmes d'optimisation	18
1.2.1. Les méthodes déterministes	19
1.2.2. Les méthodes stochastiques non inspirées de la nature	19
1.2.3. Les algorithmes inspirés de la nature	20
1.2.4. Les algorithmes hybrides ou mimétiques	22
1.3. Les algorithmes d'optimisation : de l'exploration à l'exploitation	23
1.4. Critère de choix d'un algorithme d'optimisation	24
2. Les méthodes d'optimisation appliquées pour les modèles IHACRES et GR4J	25
2.1. Présentation du Shuffled Complex Evolution (SCE-UA)	25
2.2. Présentation des Algorithmes Génétiques	32
2.2.1. Les étapes de l'algorithme génétique	32
2.2.2. Algorithme génétique de base	36
2.3. Mise en évidence de la supériorité du SCE-UA	36
3. Calage des Modèles pluie-débit (Avantages et inconvénients)	38
3.1. Période de mise en route	38
3.2. Période de calibration	38
3.2.1. Le calage manuel	38
3.2.2. Le calage automatique	38
3.3. Période de validation	39
3.4. Difficultés	40
4. Conclusion	40

Chapitre II :	41
Matériels et Méthodes	
1. Matériels	43
1.1. Le langage de programmation R	43
1.1.1. Origines	43
1.1.2. Pourquoi utiliser R ?	43
1.1.3. R et les statistiques	44
1.1.4. R et les graphiques	44
1.1.5. R et l'Hydrologie	45
1.2. La machine de calcul	47
1.3. Données	47
1.3.1. Région de Canning, Australie	47
2. Méthodes	51
2.1. Conception de l'étude comparative de performance	51
2.2. Les fonctions tests	52
2.2.1. La fonction Goldsetein-Price	52
2.2.2. La fonction Six-hump Camelback	52
2.2.3. La fonction Rastrigin	53
2.2.4. La fonction de Ackley	54
2.2.5. La fonction de Hartmann 6D	55
2.2.6. La fonction Griewank	56
2.2.7. Modèle cwi-gr4jrouting	57
2.3. Calcul	58
Chapitre III :	61
Observation et Discussion	
1. Les résultats de l'étude comparative de performance	62
2. Discussion	70
3. Conclusion	71
Conclusion Générale	72
Références	74

Liste des Tableaux

Tableau 2.1 : Tableau résumant les statistiques simple des données de Canning, Australie

Tableau 2.2 : Les problèmes tests utilisés dans cette étude

Tableau 2.3: Résumé des paramètres par défauts des deux algorithmes SCE et GA

Tableau 3.a : Résultats pour la fonction Goldstein-Price

Tableau 3.b : Résultats pour la fonction Six-Hump Camelback

Tableau 3.c : Résultats pour la fonction Rastrigin

Tableau 3.d : Résultats pour la fonction de Ackley

Tableau 3.e : Résultats pour la fonction de Hartmann 6D

Tableau 3.f : Résultats pour la fonction Griewank

Tableau 3.g : Résultats pour le calage du modèle hybride $sma = cwi$, $routing = gr4jrouting$ sur la période 1978 des données de Canning, Australie.

Liste des Figures

- Figure 1.1** : Illustration de l'interdépendance des paramètres du modèle
- Figure 1.2** : Indifférence de la fonction $F(X)$ objective au paramètre X
- Figure 1.3** : Illustration du problème des optimum locaux d'une fonction à deux variables $F(x,y)$
- Figure 1.4** : Organigramme de l'algorithme Shuffled Complex Evolution (SCE-UA) (Duan, 1993)
- Figure 1.5** : Diagramme de la stratégie Compétitive Complex Evolution (CCE)
- Figure 1.6** : Illustration de la méthode Shuffled Complex Evolution (SCE-UA) en 2D
- Figure 1.7** : Illustration des étapes d'évolution de chaque complexe
- Figure 1.8** : Courbe du nombre minimal de complexes en fonction du nombre de paramètres à caler, résumant les travaux de Duan et al. (1994)
- Figure 1.9** : Structure générale d'un Algorithme génétique
- Figure 1.10** : Exemple d'un croisement (Le croisement génère des individus qui sont le résultat du mélange de plusieurs solutions)
- Figure 1.11** : Exemple d'une mutation (la mutation est une modification aléatoire d'un individu).
- Figure 1.12** : Schéma de la chronologie du calage (Berthier, 2005)
- Figure 2.1** : L'interface du langage de programmation R (RStudio)
- Figure 2.2** : Positionnement de la région de Perth en Australie Occidentale.
- Figure 2.3** : Bassin versant de la région Canning, Australie
- Figure 2.4** : Données hydrologique de la rivière de Canning, Australie
- Figure 2.5.a** : Histogramme des pluies journalières de la région de Canning, Australie
- Figure 2.5.b** : Histogramme des débits journaliers de la région de Canning, Australie
- Figure 2.6.a** : Représentation de la fonction Goldstein-Price sur son domaine de définition
- Figure 2.6.b** : Représentation de la fonction Six-Hump Camelback sur son domaine de définition
- Figure 2.6.c** : Représentation de la fonction Rastrigin sur son domaine de définition
- Figure 2.6.d** : Représentation de la fonction de Ackley sur son domaine de définition
- Figure 2.6.e** : Représentation de quelques parcelles zoomées de la fonction Girewank
- Figure 2.7** : Hydrogrammes observé et simulé sur la période 1978. (Canning, Australie)
- Figure 3.a** : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Goldstein-Price
- Figure 3.b** : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction fonction Six-Hump Camelback
- Figure 3.c** : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Rastrigin
- Figure 3.d** : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Ackley

Figure 3.e : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Hartmann 6D

Figure 3.f : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Griewank

Figure 3.g : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour le calage du modèle hybride sma = cwi, routing = gr4jrouting sur la période 1978 des données de Canning, Australie.

Introduction

Générale

Bien que très peu abondante dans l'atmosphère, l'eau joue un rôle fondamental dans le fonctionnement du système climatique. Elle est localement menacée ou très dégradée par la pollution et surexploitation. Aussi, elle n'est pas répartie uniformément sur la terre ni dans le temps. Il n'y a pas de problème de l'eau relatif à la globalité de la planète, mais plutôt une diversité de scénarios de disponibilité localisée. Cette ressource est qualitativement limitée, et localement quantitativement limitée.

Examiner la façon dont l'eau voyage au cours de son cycle nous permet de mieux comprendre ses interactions avec l'environnement et de mieux évaluer la quantité disponible pour la consommation de l'homme. Cette évaluation est utilisée pour assurer une meilleure gestion de la ressource étant donné qu'elle n'est pas répartie uniformément ni dans le temps ni dans l'espace.

Cependant, la première étape de gestion des ressources en eau consiste en leur estimation, et en la maîtrise de leur évolution dans le temps. La connaissance du débit d'un cours d'eau est devenue indispensable à la gestion des systèmes hydrologiques. Plusieurs axes de recherche se sont développés dans ce sens, qui visent la quantification de cette ressource en partant de son origine première: la pluie, jusqu'à arriver au débit du cours d'eau, d'où la modélisation pluie-débit.

On entend par modèles pluie-débit l'ensemble des équations représentant le bilan hydrique et les équations de transferts des flux hydriques mis en jeu au cours du cycle hydrologique à l'échelle d'un bassin versant. Ce sont des modèles prédictifs des écoulements dans les cours d'eau mais aussi des autres échanges sol-atmosphère comme l'évapotranspiration, l'infiltration et la percolation. Ces modèles apparaissent essentiels dans l'aide à la décision pour la gestion intégrée des ressources en eaux, et en particulier l'analyse du risque hydrologique et la prévision des crues. D'autre part, ces modèles permettent de simuler l'impact du changement climatique sur les composantes du cycle hydrologique.

Toutefois la performance des modèles pluie-débit dans la simulation des débits des cours d'eau et des autres flux du cycle hydrologique dépend du choix du modèle et certainement aussi de la qualité de l'estimation de leurs paramètres, elle est elle-même tributaire d'une stratégie de calage du modèle hydrologique adopté.

L'opération de calage consiste à choisir les paramètres du modèle de façon à obtenir la meilleure performance du modèle moyennant la confrontation des simulations avec des observations. Avec le développement de l'outil informatique, le recours au calage automatique, basé sur des algorithmes d'optimisation est devenu de plus en plus répandu. Cependant, les difficultés que présente un problème de calage au vue de la complexité du modèle pluie-débit et des critères d'évaluation nous mènent à poser les questions suivantes : Quelle est la fonction objective (critère d'évaluation) à optimiser ? Et quel est l'algorithme d'optimisation à choisir et sous quelles conditions ?

Notre travail s'inscrit dans l'optique de répondre à la deuxième question, et donc déterminer le meilleur algorithme.

Récemment, de nombreuses techniques d'optimisation sont largement utilisées dans des applications de modélisation informatique. Un avantage majeur de l'utilisation de techniques d'optimisation avec les modèles est d'obtenir de meilleurs résultats sans efforts considérables. Les techniques d'optimisation peuvent être divisées en deux groupes: les méthodes de recherche globale telle que l'algorithme génétique (GA) et Shuffled Complex Evolution (SCE-UA), et les méthodes de recherche locale, telles que la méthode de Downhill Simplex, la méthode de recherche des motifs et la méthode de la direction de rotation. De nombreuses études ont rapporté que les méthodes de recherche globale aboutissent à de meilleurs résultats que les méthodes de recherche locale.

Le coût du calage des modèles pluie-débit donc, estimé en termes de temps de calcul est un facteur important voire déterminant du choix et de la conception d'un algorithme d'optimisation. On définit l'efficacité d'un algorithme d'optimisation comme son aptitude à trouver la solution optimale. Elle peut être mesurée par exemple par le taux de réussite de tests répétés. On définit l'efficacité d'un algorithme d'optimisation comme sa vitesse de convergence. Elle peut être mesurée par le nombre d'évaluations de la fonction objective nécessaires pour atteindre la solution optimale.

Dans l'étude suivante, deux algorithmes d'optimisation les plus cités dans la littérature ont été choisis. Tandis que les algorithmes génétiques, sont les plus connus et les plus utilisés depuis leur création en 1989, en présentant une approche mimétique de l'évolution biologique pour choisir la meilleure espèce et survivre un environnement, le Shuffled Complex Evolution a été spécifiquement développé en 1993 à l'université d'Arizona pour le calage des modèles hydrologique, offrant une capacité d'atteindre l'optimum avec une efficacité et une efficacité élevées qui permettent aux modèles de s'adapter aux situations de prévision et d'alerte.

Pour répondre à la question posée, nous allons mesurer l'efficacité et l'efficacité des deux algorithmes à franchir des épreuves de difficultés croissantes illustrées par des fonctions tests spécifiquement choisies pour mettre à l'épreuve la performance de ces méthodes. A la fin, une comparaison des deux algorithmes va être effectuée sur une opération de calage sur les données de l'Australie du modèle IHAcres, jugé le plus performant à l'issue de l'étude de travail de l'ingénieur du même auteur.

Les méthodes vont être détaillées et les résultats analysés. Des conclusions et suggestions seront proposées à la fin de ce travail.

Chapitre I :

**Problèmes d'Optimisation
et Calage Automatique des
Modèles Hydrologiques**

I. 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 [6].

Le problème de calage d'un modèle pluie-débit à une sortie (Débit) peut 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 :

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

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 du temps i

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

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

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

Avec

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

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

S_{ci} : Le débit calculé du modèle au pas du 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.

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.

1.1. Difficultés que présente l'optimisation d'un modèle pluie-débit :

Le problème de calage automatique des modèles pluie-débit a été pendant ces dernières décennies l'un des problèmes d'optimisation les plus traités dans le domaine de l'hydrologie. Il est connu pour certaines difficultés spécifiques. Celles-ci ont été identifiées et traitées afin de pouvoir choisir les algorithmes d'optimisation pouvant y faire face voire même développer des algorithmes d'optimisation appropriés. Ces difficultés conduisent aussi bien à des problèmes à l'algorithme d'optimisation de détecter l'optimum qu'à la non unicité de la solution optimale (problème d'équifinalité). Les difficultés que présente le calage automatique d'un modèle hydrologique sont [7]:

a. Interdépendance des paramètres du modèle :

Une variation de la valeur d'un paramètre peut être compensée en termes de valeur de la fonction objective par un ou plusieurs autres paramètres. Le cas du problème d'optimisation à deux paramètres peut illustrer ce problème comme le montre la **figure 1.1**. De ce fait, un grand nombre de jeux de paramètres peut donner des résultats similaires. L'algorithme de recherche peut ainsi rencontrer des difficultés dans cette vallée parce qu'il n'est pas capable de progresser là où la fonction objective est stable [6]. Il s'agit autrement du problème d'équifinalité mentionné par Beven (2006) se manifestant par le fait que le modèle peut donner des performances similaires pour des jeux de paramètres différents [4].

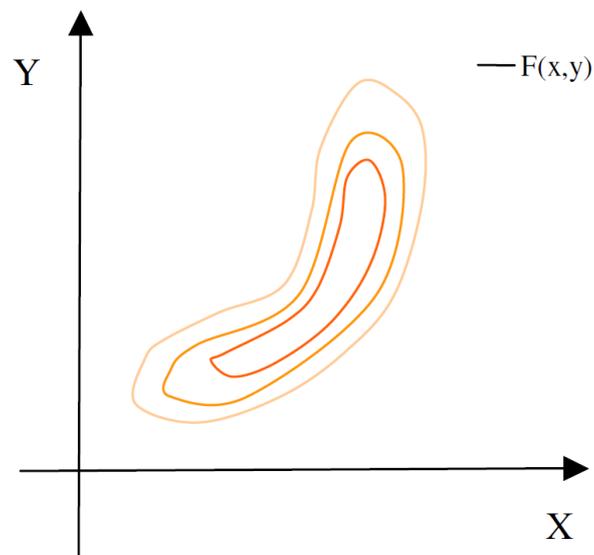


Figure 1.1 : Illustration de l'interdépendance des paramètres du modèle

b. Indifférence de la fonction objective aux paramètres :

On peut parler de l'indifférence ou de l'insensibilité de la fonction objective à un paramètre lorsque la variation de ce paramètre n'entraîne aucune variation ou une variation minime 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 dans l'estimation de l'optimum surtout dans la direction des paramètres insensibles [6]. En effet une large marge de paramètres peut former l'optimum en donnant la même performance.

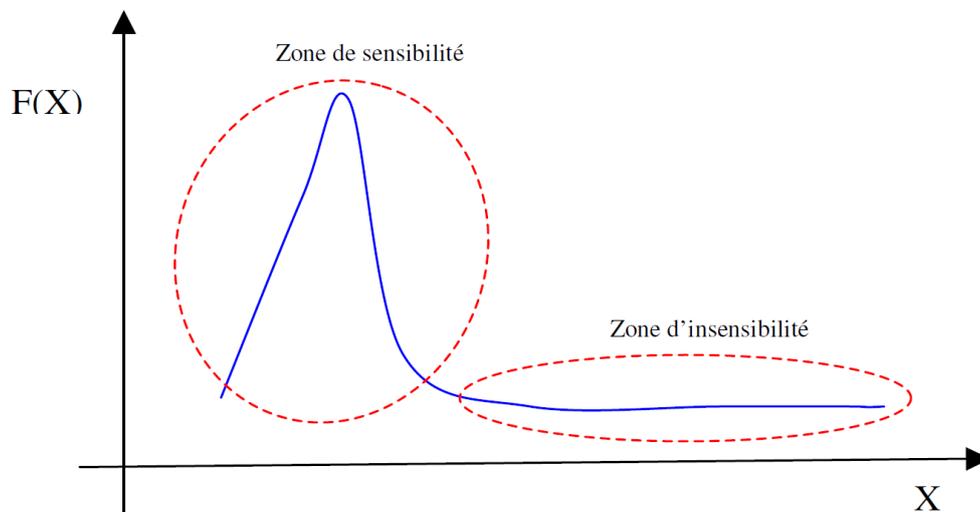


Figure 1.2 : Indifférence de la fonction $F(X)$ objective au paramètre X

Il est à noter que l'indifférence de la fonction objective vis-à-vis de certains paramètres représente un effet joint du paramètre et de la fonction objective. En effet certaines fonctions objectives sont insensibles à certains paramètres et sensibles pour d'autres paramètres. On peut citer comme exemple les indices de performances de bilan qui sont sensibles aux paramètres de bilan, et peuvent être insensibles aux paramètres de forme.

c. Définition de la direction du gradient :

Pour les méthodes indirectes, qui se basent sur l'estimation des dérivés de la fonction objective ou autrement le gradient, une difficulté additionnelle peut se poser. C'est celle d'estimer le gradient en raison de la discontinuité ou de la non dérivabilité de la fonction objective [6-7].

d. Les optimums locaux :

Les optimums locaux se manifestent par des pics de la fonction objective dans l'espace des paramètres autres que l'optimum global. Ces pics peuvent constituer des pièges pour les algorithmes d'optimisation en les faisant converger vers un optimum local et non global [6].

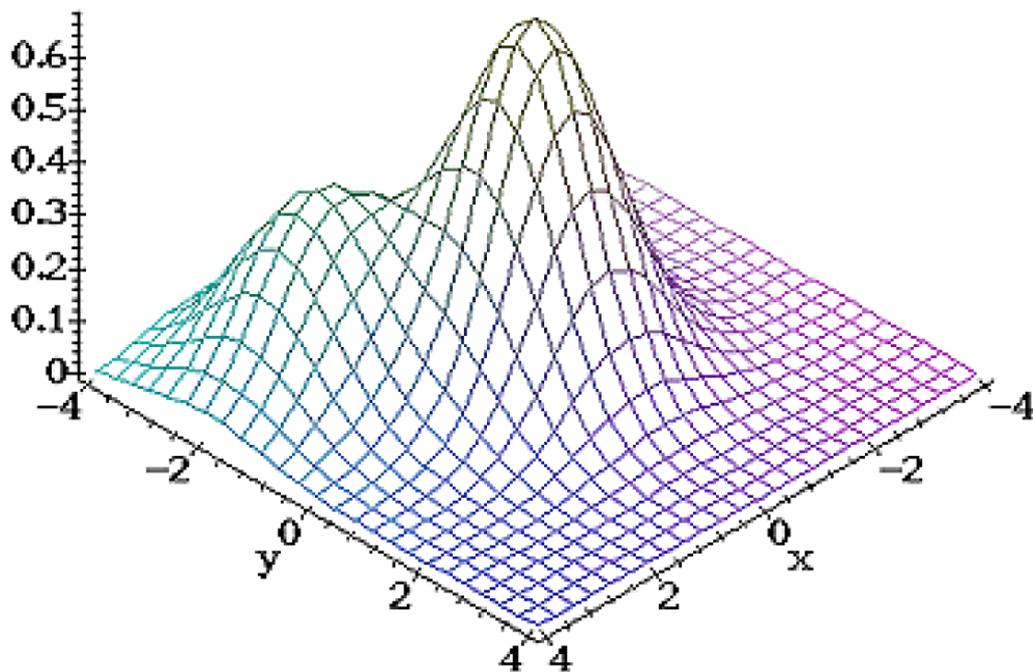


Figure 1.3 : Illustration du problème des optimum locaux d'une fonction à deux variables $F(x,y)$

e. Problème d'échelle des paramètres :

Certains algorithmes d'optimisation ne tolèrent pas de différence importante entre les intervalles de variation des paramètres à optimiser. A titre d'exemple, pour le modèle GR4J, le paramètre X1 varie entre 100 à 1200 et le paramètre X4 varie seulement entre 1,1 et 2,9. Cette différence dans les échelles peut perturber le fonctionnement de l'algorithme qui trouve une difficulté à converger vers l'optimum. Un bon remède à ce problème est d'effectuer une transformation de l'espace des paramètres afin de les ramener à la même plage de variation. Mais le grand problème est de trouver la transformation adéquate [6-7].

1.2. 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 départ, comme le cas de Downhill-Simplex.

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.

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 [6-8].

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.

1.2.1. 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.

1.2.2. 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.

1.2.3. 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,
- $P = \exp(-\Delta F/T)$ si la fonction d'état augmente.
- Avec T : un paramètre, appelé température
- ΔF : 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) [6].

(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.

1.2.4. 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.
- L'algorithme Recuit Simplex Evolutif: cet algorithme combine les trois techniques : algorithme évolutionnaire, Recuit Simulé et le Downhill-Simplex.
- Le Shuffled Complex Evolution: il combine les algorithmes évolutifs et la méthode de recherche directe Simplex. Cet algorithme a été développé au sein de l'université d'Arizona, cette fois par des hydrologues pour résoudre le problème de calage d'un modèle pluie-débit [7-8]. Le SCE-UA a été adapté dans cette étude. Il sera ainsi présenté dans une section à part.

1.3. 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 [6].

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. Le SCE-UA combine ces deux dernières techniques d'exploitation.

Dans le cas de l'algorithme de Simulation de Monte-Carlo, le mécanisme d'exploitation est absent. En effet cet algorithme génère aléatoirement des jeux de paramètres dans l'espace de recherche continuellement d'une itération à une autre sans aucun mécanisme d'exploitation des informations acquises par les itérations précédentes pour diriger la recherche, C'est d'ailleurs cette défaillance qui rend cette technique peu efficace en tant qu'algorithme d'optimisation.

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). Le Downhill-Simplex limite la mémoire à $(n+1)$ points, le SCE-UA à $(2n+1)*p$ points, avec n correspond à la dimension du problème d'optimisation et p le nombre de complexes [6-7-8]. Quant aux algorithmes génétiques, ils se limitent à la taille de la population de recherche comme mémoire, et le Simulated Annealing à un seul point.

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.

1.4. 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 [7-8].

- **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 [6-7].
- **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'efficience [6].
- **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].

2. Les méthodes d'optimisations appliquées pour les modèles

IHACRES et GR4J :

Perrin (2000) propose de caler le modèle GR4J par un algorithme de recherche locale : la méthode pas à pas consistant à effectuer une série de déplacements unidirectionnels du vecteur de paramètres jusqu'à convergence [6-15]. Perrin (2000) justifie le choix de cette méthode de recherche locale et dit qu'il n'est pas nécessaire d'avoir recours à des techniques d'optimisation sophistiquées de type recherche globale [15]. En effet ces techniques prennent leur force de « l'excès de paramétrisation » des modèles, ce qui n'est pas le cas pour GR4J. Dans la présente étude nous allons tout de même caler le modèle GR4J avec une méthode d'optimisation globale (SCE-UA) afin d'étudier son efficacité par rapport aux nombres de paramètres à optimiser.

Etant donné sa structure simplifiée, le modèle IHACRES, comme le modèle GR4J, ne forme pas un vrai défi face aux algorithmes d'optimisation. La méthode de calage adoptée dans cet outil est le SCE-UA [7-8].

Après avoir présenté la conceptualisation des modèles hydrologiques en étude dans le chapitre précédent. La suite de ce chapitre va porter sur la présentation brève de l'algorithme utilisé (Shuffled Complex Evolution).

2.1. Présentation de l'algorithme SCE-UA :

Selon Duan et al. (1992), cette méthode est efficace, robuste, flexible et efficiente. Elle prend sa force du fait qu'elle est fondée sur la synthèse de quatre concepts [7]:

- Combinaison des approches déterministes et probabilistes
- L'évolution systématique des points (sommets du complexe) dans la direction d'amélioration globale de la FO,
- l'évolution compétitive
- L'évolution indépendante des complexes avec un «shuffling» périodique ou autrement un brassage des complexes, qui est assez proche de la méthode des niches des algorithmes génétiques présentés dans la section précédente.

Les trois premiers concepts sont issus de la littérature, et le dernier a été développé par les auteurs de l'algorithme [6-7].

Une description générale des différentes étapes de la méthode est présentée comme suit, pour le cas de recherche du minimum d'une FO :

- (1) Générer aléatoirement la population aléatoire: générer un échantillon des points aléatoirement dans l'espace faisable des paramètres, et calculer la valeur de la fonction objective au niveau de chaque point. En l'absence d'une information a priori de la localisation approximative de l'optimum on utilise une distribution uniforme pour générer l'échantillon,
- (2) Classer les points par ordre croissant de la FO de telle sorte que le premier point donne la plus petite valeur de la fonction objective et le dernier représente la plus large valeur.

- (3) Répartition des individus sur p complexes, cette répartition fait en sorte de répartir équitablement les bons (petite valeur de FO) et mauvais individus (grande valeur de FO) sur les différents complexes. Par exemple on met le premier point (selon le classement fait dans l'étape 2) au premier complexe, le deuxième au second complexe et le p ème individu au p ème complexe ; ensuite le $p+1$ ème au premier complexe et ainsi de suite.
- (4) Evoluer indépendamment chaque complexe selon la méthode d'évolution compétitive des complexes basée sur la méthode Simplex.
- (5) Mélanger les résultats des complexes et les réarranger de nouveau sur p complexes (comme dans 2 et 3)
- (6) Vérifier les critères de convergence sinon répéter 3

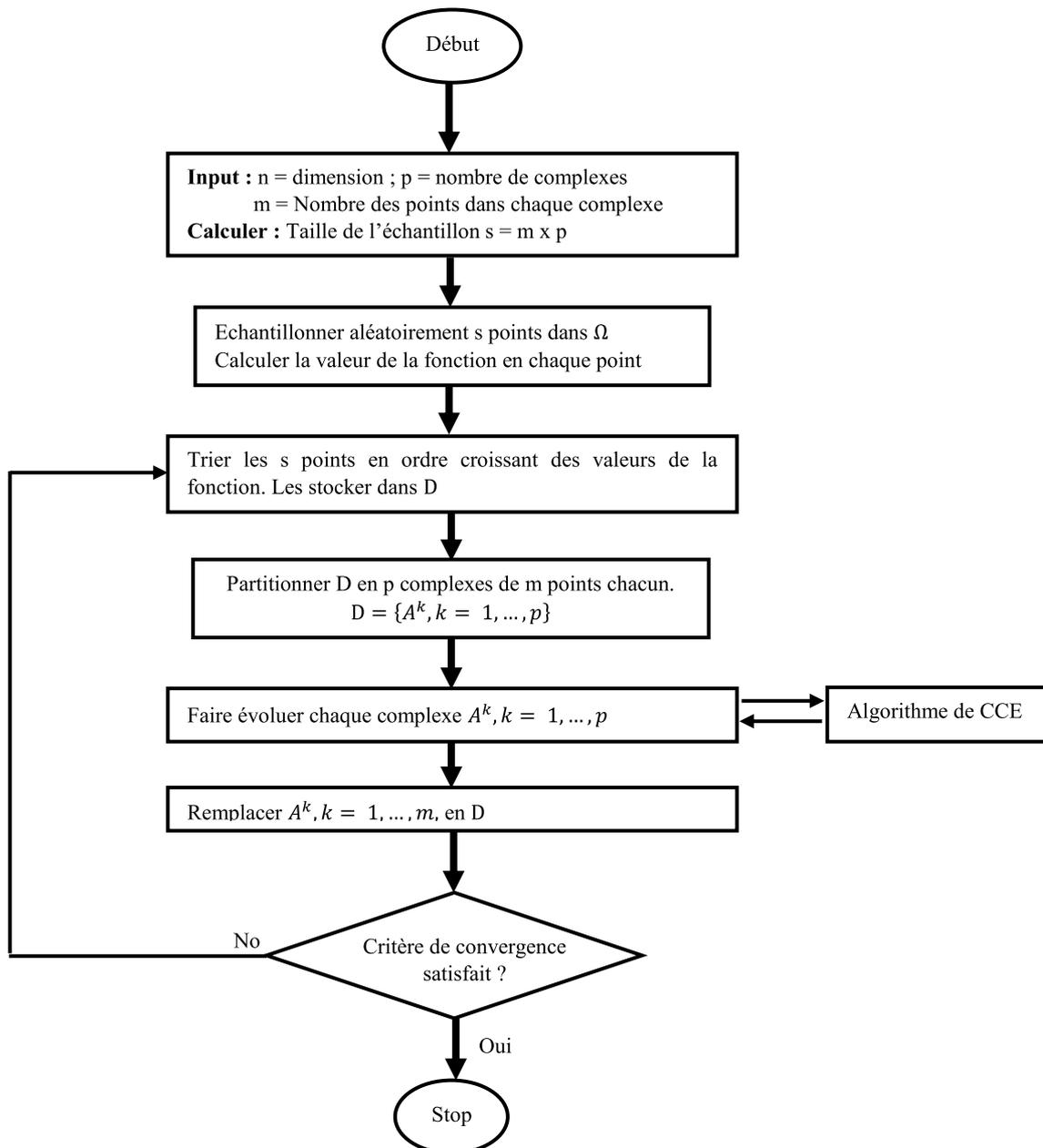


Figure 1.4 : Organigramme de l'algorithme Shuffled Complex Evolution (SCE-UA) (Duan, 1993)

Le mélange des complexes ou "Shuffling" permet aux complexes de partager l'information entre eux. En effet si l'un des complexes converge vers un optimum local il va découvrir en partageant l'information avec les autres complexes qui se sont rapprochés plus de l'optimum global qu'il ne s'agit pas de la bonne direction de recherche et donc il va changer de cap. Duan et al. (1992) insistent sur le fait que répartir la population sur des sous population qui évoluent indépendamment tout en partageant périodiquement l'information (grâce au mélange) donne un algorithme plus efficace et plus efficient que si on considère des populations de même taille travaillant indépendamment jusqu'à la fin [7-8]. Ces auteurs font une analogie avec la recherche scientifique où les groupes de recherche travaillent indépendamment puis partagent l'information à travers les colloques, les stages, les post-doctorants, etc. Ceci paraît plus efficient que si les groupes de recherches travaillent isolés les uns des autres continuellement [6].

L'algorithme d'Evolution Compétitive du complexe utilisé pour faire évoluer chaque complexe indépendamment comme l'indique l'étape 5 du SCE-UA est décrit comme suit:

- 1) affecter des poids selon une probabilité trapézoïdale aux individus constituant le complexe de façon à donner le poids le plus grand à l'individu de meilleure performance et le poids le plus petit à l'individu de mauvaise performance.
- 2) choix des parents (q sommets du simplexe) par tirage aléatoire de q individus du complexe et en suivant la distribution citée dans (1)
- 3) générer les descendants selon les trois procédures suivantes:
 - (a) réflexion du faible sommet du simplexe à travers le centroïde des autres sommets du sous-complexe, si le point réfléchi est dans les limites de l'espace de recherche, aller à (d) sinon générer aléatoirement un individu dans l'hypercube contenant le sous-complexe
 - (b) si la fonction objective s'améliore par réflexion, le point réfléchi remplace le point faible du sous-complexe sinon contracter le point faible vers le centroïde des autres points
 - (c) Si la contraction améliore la fonction objective du mauvais sommet, le point contracté remplace le mauvais point sinon générer aléatoirement un individu dans l'hypercube contenant le sous-complexe
 - (d) Répéter les étapes (a)-(c) α fois avec α un paramètre au choix de l'opérateur.
- 4) Remplacer les parents par leurs descendants dans le complexe et classer les individus du complexe par ordre croissant de fonction objective.
- 5) Répéter les étapes 1-4 b fois avec b un paramètre algorithmique à fixer par l'utilisateur et qui détermine combien de descendants sont générés.

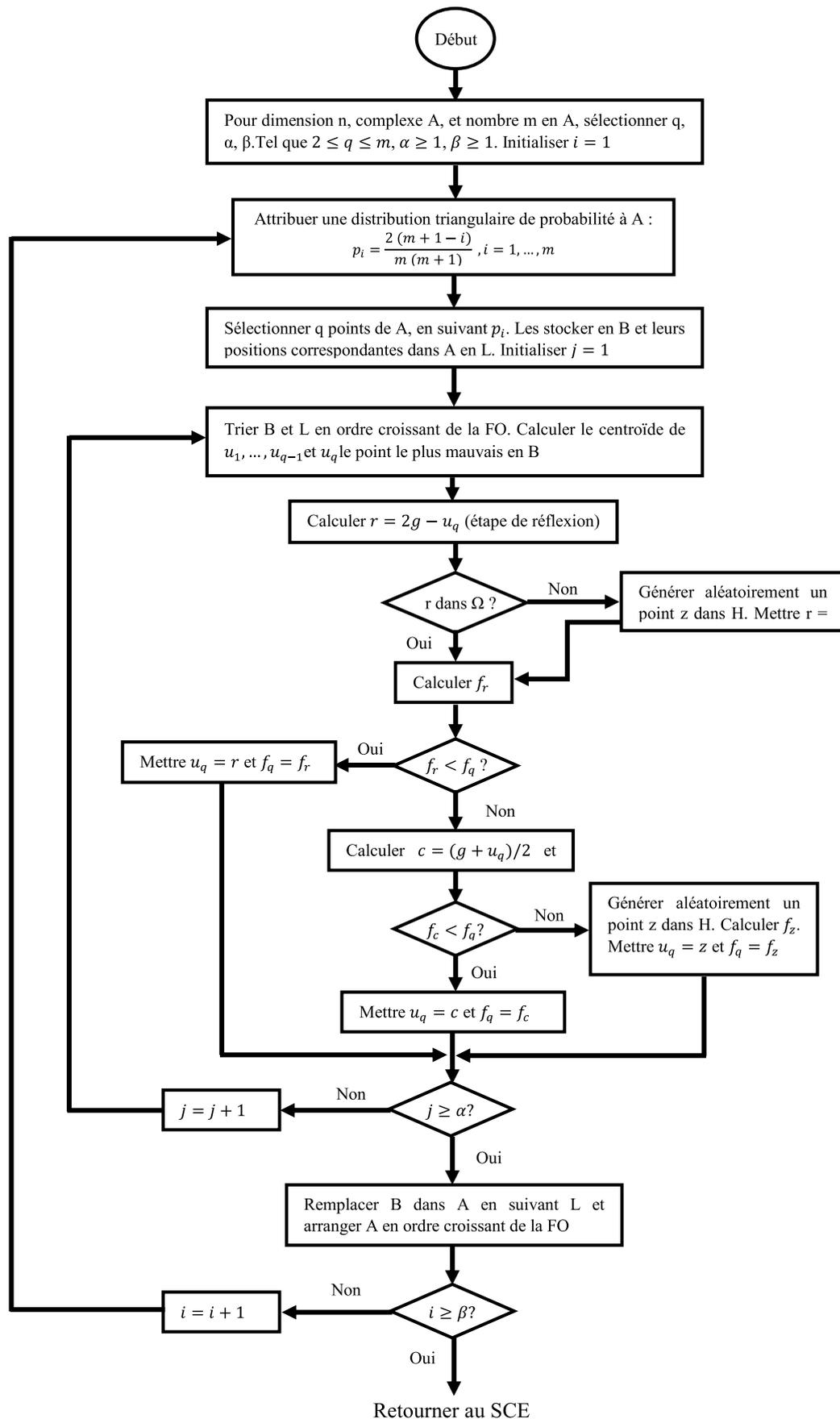


Figure 1.5 : Diagramme de la stratégie Compétitive Complexe Evolution (CCE)

Dans cette description on voit bien le caractère évolutif de l'algorithme qui se manifeste dans la partie qui concerne le choix des parents. En effet ce choix se fait de manière à donner plus de chance aux individus plus adaptés à contribuer dans la formation des générations futures (choix aléatoire selon une fonction de distribution trapézoïdale) [6].

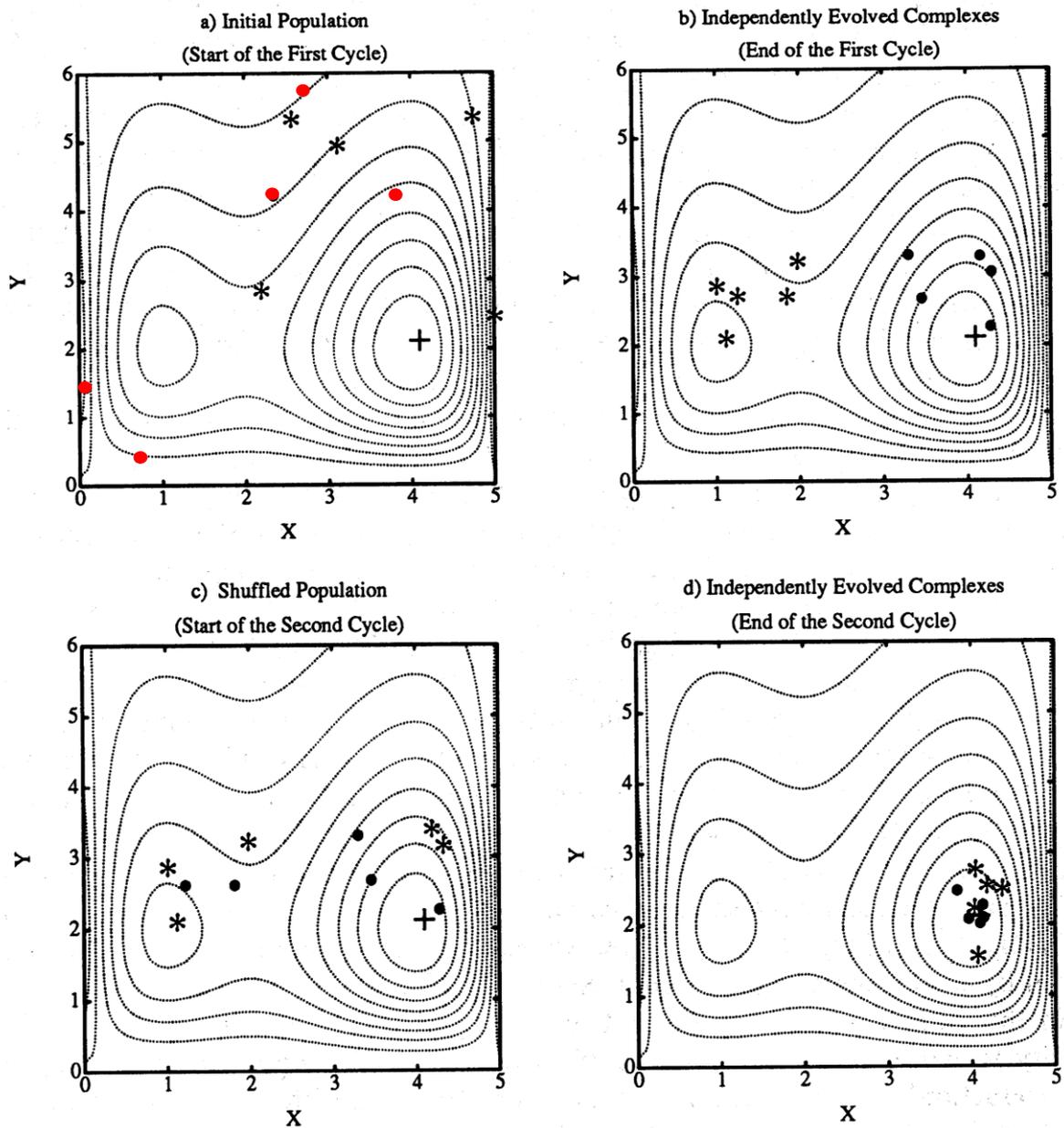


Figure 1.6 : Illustration de la méthode Shuffled Complex Evolution (SCE-UA) en 2D

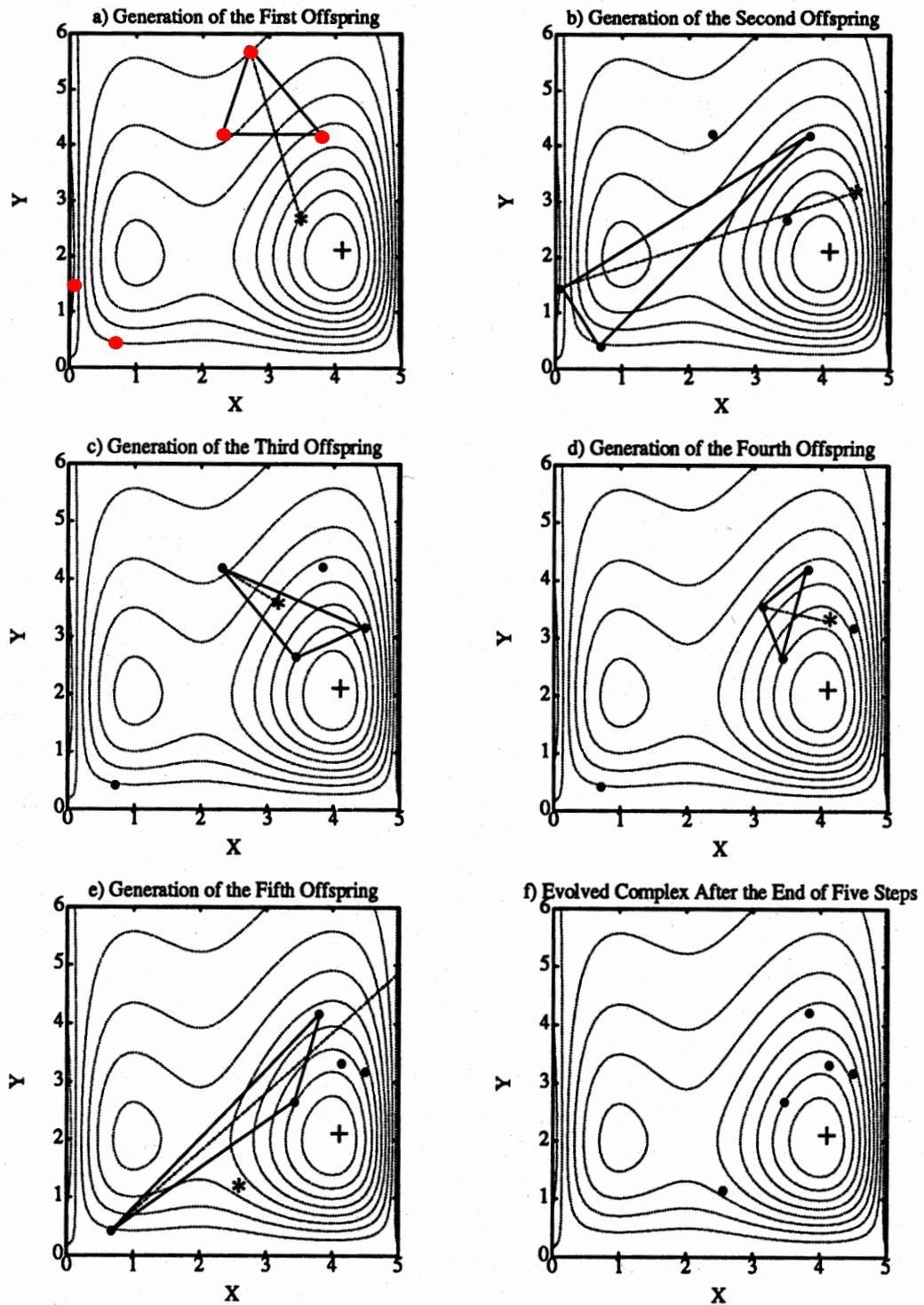


Figure 1.7 : Illustration des étapes d'évolution de chaque complexe

Les auteurs de SCE-UA (Duan et al. 1993) suggèrent de fixer les paramètres algorithmiques du SCE-UA comme suit [8]:

- taille du complexe $m = 2n + 1$, avec n la dimension du problème (nombre d'inconnus ou paramètres à caler)
- taille du sous-complexe $q = n + 1$
- $a = 1$
- $b = 2n + 1$

Le seul paramètre qui reste à fixer par l'utilisateur est le nombre de complexes p . Un choix d'un nombre de complexes grand améliore l'efficacité de SCE-UA mais en dépit de son efficacité. Au contraire un nombre petit de complexes donne un algorithme très rapide en temps de calcul mais qui peut être facilement piégé par un optimum local. Il s'agit là de choisir une valeur optimale pour p qui assure l'efficacité de l'algorithme avec une efficacité acceptable.

Duan et al. (1994), dans leur travail portant sur l'usage optimal de SCE-UA, ont essayé de proposer des valeurs de p liées à la dimension du problème. Leurs expériences ont été menées sur le modèle NWSRFS-SMA (National Weather Service River Forecast System-Soil Moisture Accounting) [6]. Dans la **figure 1.8** on résume les résultats de ces travaux en dessinant la courbe de nombre minimal de complexes en fonction du nombre de paramètres à caler. Il faudrait préciser le modèle utilisé et le nombre de paramètres maximal ; On voit l'allure exponentielle de cette relation. Une bonne mesure de sécurité consiste à choisir une valeur de p égale à la dimension du problème à optimiser.

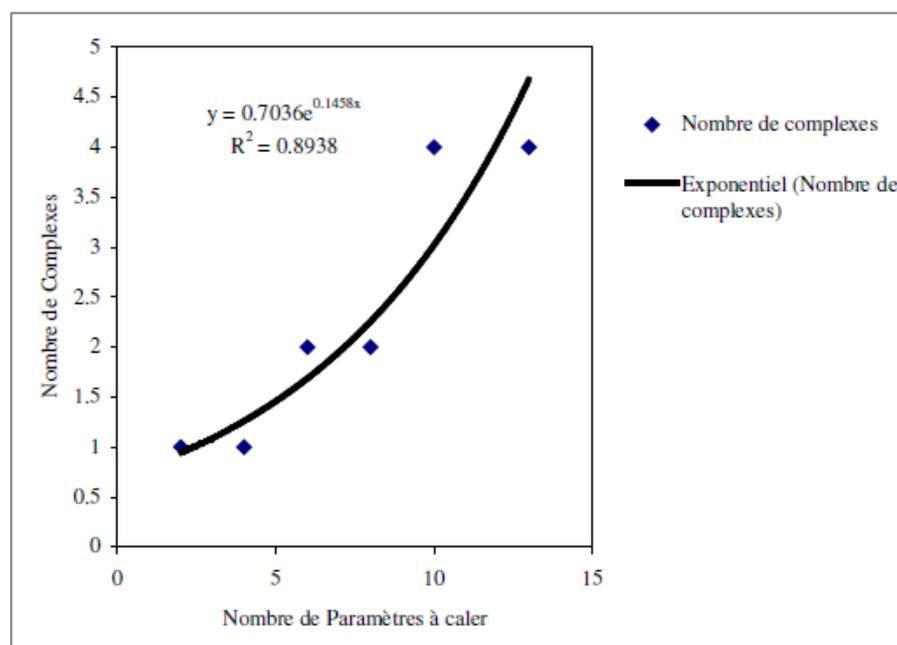


Figure 1.8 : Courbe du nombre minimal de complexes en fonction du nombre de paramètres à caler, résumant les travaux de Duan et al. (1994)

2.2. Présentation des Algorithmes Génétiques :

Depuis l'idée originale proposée par Holland (1975), le développement des algorithmes génétiques dans les trois dernières décennies a connu un essor considérable. Les algorithmes génétiques sont des techniques de recherche basées sur les mécanismes de la sélection naturelle et de la génétique. Ils se sont avérés très efficaces dans la résolution des problèmes complexes d'optimisation, tels que le problème du voyageur de commerce ou les problèmes d'ordonnement.

Pour mettre en œuvre un algorithme génétique, on doit disposer des cinq éléments suivants :

- Un mécanisme de génération de la population initiale : Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.
- Un principe de codage de l'élément de population, c'est-à-dire un codage de solutions utilisé sous la forme de chromosomes (La qualité du codage des données conditionne le succès des algorithmes génétiques).
- Une fonction à optimiser : L'application de cette fonction à un élément de la population donne sa fitness, une fonction qui permet d'évaluer l'adaptation d'un chromosome à son environnement, ce qui offre la possibilité de comparer des individus.
- La sélection est basée sur la reproduction et sur le codage génétique, qui stocke les informations décrivant l'individu sous forme de gènes. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.
- L'algorithme utilise des paramètres de dimensionnement : taille de la population, probabilité de croisement et de mutation, nombre total de générations.

2.2.1. Les étapes de l'algorithme génétique

Le schéma de la **Figure 1.9**, illustre la structure générale d'un AG. L'algorithme commence par générer une population initiale qui se compose d'un nombre déterminé d'individus. La meilleure solution peut ne pas se trouver dans cette population initiale. À chaque génération, une succession d'opérations de sélection, de croisement, de mutation, d'évaluation et de remplacement est appliquée aux individus de la population, afin de produire la génération suivante. Lorsque le critère d'arrêt est atteint, la meilleure solution trouvée est sélectionnée et l'algorithme s'arrête.

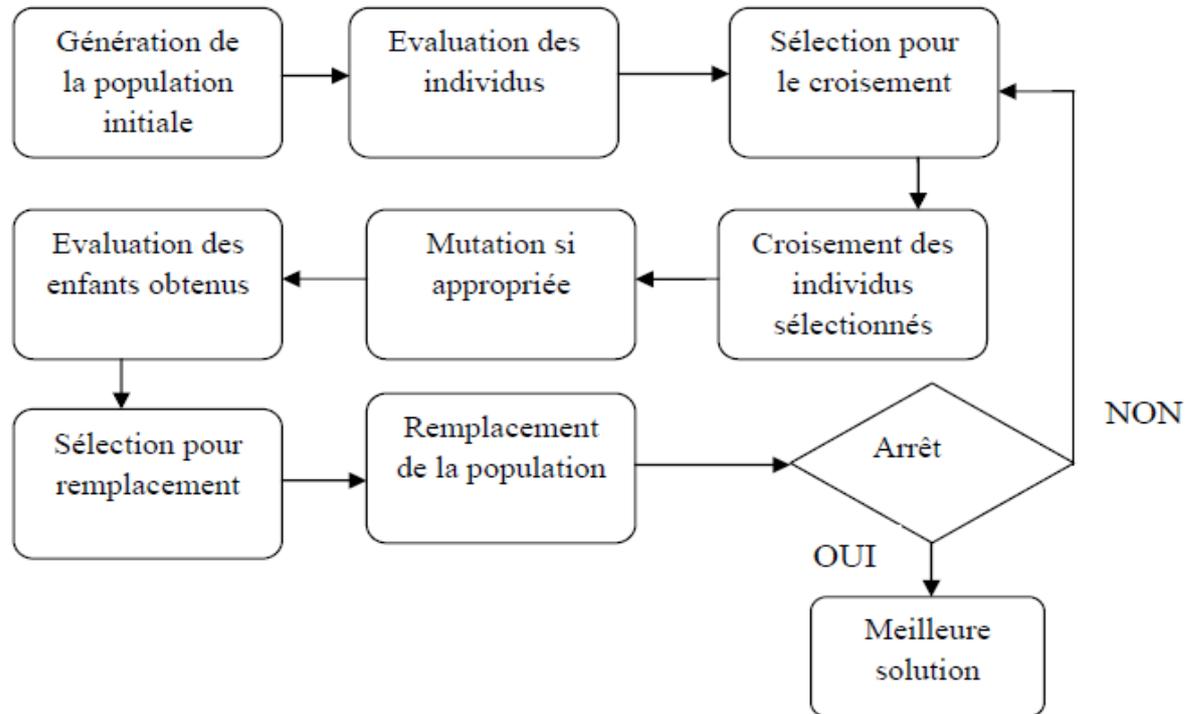


Figure 1.9 : Structure générale d'un Algorithme génétique

a. Codage

Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux. Chaque individu de la population est codé par un chromosome. Une population est donc un ensemble de chromosomes. Généralement, un chromosome est une chaîne ou un ensemble de chaînes, souvent de bit ou d'entiers. Chaque chromosome code un point de l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome.

b. Evaluation (fitness)

Les performances de chaque individu sont évaluées en utilisant une fonction d'évaluation. Cette fonction permet d'évaluer la capacité d'un individu à survivre en lui affectant un poids appelé fitness. Cette fitness est calculée afin que les plus forts soient retenus dans la phase de sélection, puis modifiés dans la phase de croisement et mutation.

c. La sélection

Cet opérateur est chargé de favoriser les meilleurs individus. La sélection joue un rôle très important dans les algorithmes génétiques: d'une part, pour diriger les recherches vers les meilleurs individus et d'autre part, pour maintenir la diversité des individus dans la population. Elle est liée au compromis entre la vitesse de convergence élevée et une forte

probabilité de trouver un optimum global dans le cas d'un problème d'optimisation. Si la sélection choisit seulement le meilleur individu, la population convergera rapidement vers cet individu.

d. Croisement

L'opérateur de croisement, appelé aussi recombinaison, est l'instrument majeur des innovations dans l'algorithme génétique. Il a pour objectif d'enrichir la diversité de la population en manipulant la combinaison des chromosomes. Généralement, le croisement se fait en combinant deux parents et génère deux enfants qui vont hériter certaines caractéristiques de leurs parents et qui peuvent se rapprocher de la solution optimale. Le croisement peut être simple, à un seul point, ou multiple, à deux ou plusieurs points. Dans le premier cas, se croisent en un seul point (figure 2.4). Dans le deuxième cas, les chromosomes se croisent en deux ou plusieurs points, et s'échangent de plusieurs portions.

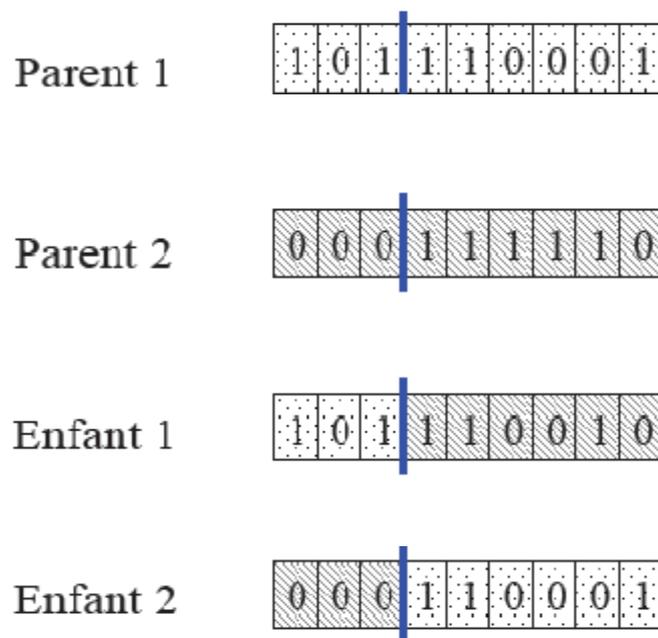


Figure 1.10 : Exemple d'un croisement (Le croisement génère des individus qui sont le résultat du mélange de plusieurs solutions)

e. Mutation

Cet opérateur a pour but de diversifier la population afin d'examiner d'autres parties de l'espace de recherche. Il consiste à remplacer, de façon aléatoire, un gène au sein d'un chromosome par un autre, c'est donc une modification légère dans le chromosome muté. La mutation a pour rôle d'éviter la convergence trop rapide de l'algorithme vers un minimum local. La figure 2.5 montre un exemple de mutation.

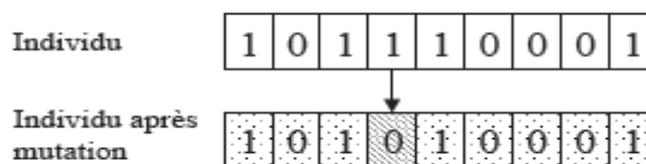


Figure 1.11 : Exemple d'une mutation (la mutation est une modification aléatoire d'un individu).

f. Phase de remplacement

Une fois qu'on a généré de nouveaux individus par croisement et par mutation, il faut décider de la constitution de la nouvelle population, le remplacement décide quels individus conserver et insérer dans la nouvelle population.

g. Critère d'arrêt :

L'arrêt de l'évolution d'un algorithme génétique est l'une des difficultés majeures car il est souvent difficile de savoir si la meilleure valeur a été atteinte ou pas. Actuellement les critères les plus utilisés sont les suivants :

- Arrêt de l'algorithme après un certain nombre de générations, fixé au départ.
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement, c'est à dire les meilleurs individus ne s'améliorent plus depuis un certain nombre de générations.

h. Valeurs des paramètres :

Les paramètres qui conditionnent la convergence d'un algorithme génétique sont :

- La taille de la population d'individus.
- Le nombre maximal de générations.
- La probabilité de croisement.
- La probabilité de mutation.

Les valeurs de tels paramètres dépendent fortement de la problématique étudiée. Ainsi il n'existe pas de paramètres qui soient adaptés à la résolution de tous les problèmes qui peuvent être posés à un algorithme génétique. Cependant, certaines valeurs sont souvent utilisées (définies dans la littérature) et peuvent être de bons points de départ pour démarrer une recherche de solutions à l'aide d'un AG.

La probabilité de croisement est choisie dans l'intervalle $[0.7, 0.99]$;

La probabilité de mutation est choisie dans l'intervalle $[0.001, 0.01]$.

Trouver de bonnes valeurs à ces paramètres est donc un problème parfois délicat.

2.2.2. Algorithme génétique de base :

Data : Problème d'optimisation difficile.

Résultat : Solution de bonne qualité.

- 1- Générer une population aléatoire de **n** chromosomes;
- 2 - Evaluer la fitness pour chaque chromosome **x**;
- 3 - **while** (non-critère d'arrêt) **do**
- 4 - Sélection des parents;
- 5 - Appliquer le croisement avec une probabilité **Pc**;
- 6 - Appliquer la mutation avec une probabilité **Pm**;
- 7 - Ajouter les enfants à la population;
- 8 - Evaluation;
- 9 - Remplacement;
- 10 - **end**

2.3. Mise en évidence de la supériorité de SCE-UA

Beaucoup de travaux de recherche ont fait le point sur le test d'algorithmes au calage des modèles hydrologiques et à la comparaison de leurs performances. Les algorithmes d'optimisation stochastiques globaux, comme les algorithmes génétiques, le SCE-UA et le Simplex Simulated Annealing ont prouvé leur aptitude de trouver la solution optimale avec un moindre coût. Le SCE-UA s'est bien distingué de ces algorithmes évolutifs en prouvant sa haute performance surtout pour les problèmes les plus compliqués, comme le montrent les études comparatives présentées dans la suite.

Duan et al. (1992) ont comparé les algorithmes Adaptive Random Search, Multistart Simplex, Random Generation et SCE-UA pour le calage du modèle pluie-débit expérimental Sixpar, une version simplifiée de Sacramento Soil Moisture Accounting (SAC-SMA) développé spécialement pour tester SCE-UA, pour un cas synthétique sans erreurs sur les débits où la solution optimale est connue à l'avance. Ils ont pu constater la supériorité nette de SCE-UA en termes d'efficacité et d'efficience. En effet ils trouvent pour 100 tests indépendants de chacun des algorithmes d'optimisation testés, que le SCE-UA ne nécessite en moyenne que 3000 évaluations de la fonction objective pour atteindre la solution optimale avec un taux de succès de 100%. Le Multistart Simplex, nécessite 11.000 évaluations de la fonction objective pour se rapprocher de ce taux de réussite. Quant à l'ARS, il nécessite en moyenne 25.000 évaluations de la fonction objective pour un taux de succès ne dépassant pas les 30%. Pour

L'ARS couplé avec Simplex, ça nécessite en moyenne 5000 évaluations de la fonction objective pour un taux de succès de 60% [7].

Duan et al (1993) ont comparé le SCE-UA à l'algorithme Multi-Start-Simplex pour le cas synthétique et réel de calage du modèle hydrologique SAC-SMA (Sacramento soil moisture accounting). Ils ont trouvé une remarquable supériorité du SCE-UA en termes d'efficacité et d'efficacités. En effet pour le cas synthétique, et pour 100 tests indépendants des deux algorithmes d'optimisation, la totalité des tests de SCE-UA arrivent à trouver la solution optimale, mais aucun des tests de Multi-Start-Simplex n'arrive à converger vers la solution optimale, cet algorithme se trouve toujours attiré vers les optimums locaux. Pour le réel, le SCE-UA arrive à trouver des solutions avec une erreur quadratique toujours plus petite, et avec en moyenne seulement 10.500 évaluations de la fonction objective contre 18.000 pour le Multi-Start-Simplex [6-7].

Franchini et al. (1998) ont effectué une étude comparative de 3 méthodes d'optimisation pour des cas théoriques et réels de calage du modèle ADM (Franchini , 1996) pour des cas globaux et semi-distribués, avec la somme des erreurs quadratiques comme fonction objective. Les méthodes mises à l'essai sont :

- SCE-UA : Shuffled Complex Evolution de Duan et al. (1992)
- PS-SQP : le Patern Search (PS) couplé avec la programmation quadratique séquentielle (SQP)
- L'algorithme génétique (GA) selon un schéma proposé par Franchini et Geleati (1997) couplé avec la programmation quadratique séquentielle (SQP)

Ils ont testé ces méthodes pour le calage du modèle ADM pour : (i) des cas théoriques et réels de modèle global avec 11 paramètres mis en jeu lors de l'optimisation et (ii) des cas théoriques et réels de modèle semi distribué (3 sous-bassins), avec au total 37 paramètres mis en jeu (11 pour chaque sous bassins, et 4 pour la fonction de transfert dans le réseau hydrographique). La fonction objective utilisée est l'erreur quadratique. Ils ont conclu que les 3 méthodes donnent des résultats similaires pour le cas théorique global. Cependant, le SCE-UA se distingue des autres méthodes pour le problème d'optimisation le plus compliqué (réel semi distribué). En effet cette méthode est plus efficace et efficiente. Le SCE-UA donne pour les cas réels des résultats plus cohérents (les jeux de paramètres obtenus pour les différents tests indépendants sont assez similaires) avec une erreur quadratique sur les débits plus petite [6].

De même en appliquant l'algorithme au calage du modèle pluie-débit SFB, il a été constaté que cet algorithme a des performances supérieures à celles des algorithmes génétiques.

3. Calage des Modèles pluie-débit : (Avantages et inconvénients)

Le calage consiste à sélectionner le jeu de paramètres d'un modèle pluie-débit de façon à ce que celui-ci simule le comportement hydrologique du bassin versant de la meilleure façon possible [3]. Ce jeu de paramètres doit être adéquat offrant la meilleure reconstitution des sorties en particulier les débits [6]. Pour cela, on a recours à faire une confrontation des débits simulés à ceux observés par calage manuel, qui consiste à modifier les paramètres jusqu'à obtenir la meilleure adéquation entre les simulations et les observations sur la période d'étude. Autrement, on a recours au calage automatique qui vise à optimiser une fonction d'ajustement minimisant l'erreur de modélisation sur la période considérée [3-6]. Un bon calage d'un modèle pluie-débit se fait en principe sur trois périodes, mise en route, calibration et validation pour la vérification.

3.1. Période de mise en route :

Cette période est très importante pour l'initiation du système afin que le modèle soit indépendant de l'état initial [3]. En général, cette période est estimée à une année, cependant cela peut ne pas être très favorable si la série de donnée n'est pas assez longue et qu'il est préférable d'exploiter cette période pour le calage du modèle [21].

3.2. Période de Calibration:

C'est la période d'observation utilisée pour faire l'apprentissage du modèle et déterminer le jeu de paramètres adéquat pour reproduire des débits simulés très proches aux observés. Généralement, pour caler un modèle pluie-débit, une durée d'observation de cinq ans au minimum est requise [3-6]. On utilise deux méthodes de calage pour obtenir le meilleur jeu de paramètres qui nous permettent d'ajuster au mieux le modèle aux valeurs observées : le calage manuel et le calage automatique.

3.2.1. *Le calage manuel :*

Pour cette méthode de calage, c'est l'utilisateur du modèle qui recherche par tâtonnement la valeur des paramètres qu'il va utiliser pour faire tourner le modèle [3]. Cette pratique nous permet de comprendre le fonctionnement du modèle et quel est l'effet de chaque paramètre sur le comportement du modèle, mais en pratique, cette opération est longue et fastidieuse [6-21]. Ce calage, fondé sur l'expertise de l'opérateur, est généralement basé sur le diagnostic visuel fait par l'opérateur des hydrogrammes des débits (observés et calculés) ainsi que sur l'appréciation de certains indices de performance du modèle [6].

3.2.2. *Le calage automatique :*

Les chercheurs ont bénéficié de l'essor de l'outil informatique lors de ces dernières décennies pour développer des outils de calage automatique qui peuvent remédier au défaut majeur du calage manuel [6]. Dans ce cas, les données simulées sont ajustées aux données

observées, au moyen d'un programme informatique qui opte pour minimiser une fonction objective qui représente l'erreur du modèle à travers un ajustement itératif représenté par un algorithme d'optimisation [12], cet algorithme est plus avantageux que le calage manuel [6-12]:

- il nécessite moins de temps de calcul,
- il n'est pas tributaire de l'expertise de l'opérateur,
- il est automatique, autrement, l'intervention de l'opérateur est minimisée.

Cependant, il présente l'inconvénient d'être influencé par certaines contraintes [12-21] :

- Le choix de la fonction objective est subjectif, et les paramètres calibrés dépendent de ce choix.
- L'autocorrélation des paramètres.
- La sensibilité de la fonction objective.
- La présence d'optima locaux, et plusieurs zones de convergences.
- Les paramètres du modèle peuvent être calibrés à des valeurs qui n'ont pas un sens physique.

Ces inconvénients, nécessitent que la calibration automatique doive toujours être accompagnée par une appréciation manuelle de l'opérateur du modèle.

3.3. Période de Validation:

Avoir un bon ajustement entre les simulations du modèle et les observations du terrain sur la période de calibration est un test nécessaire pour tester l'applicabilité du modèle à un BV ; cependant, cette étape seule, reste insuffisante parce qu'elle ne garantit en aucun cas que ce modèle va bien simuler le débit pour les périodes non calibrées [12]. La validation, une période souvent confondue avec l'étape de calage, consiste à s'assurer que le modèle, une fois calibré, réagit correctement à des variations des données d'entrée qui n'ont pas servi à sa calibration [5-21]. Il faudrait s'assurer que les paramètres calibrés, permettent de s'assurer que les écarts entre les résultats obtenus et observés dans la période de validation sont raisonnables [21]. Cette mesure doit être vérifiée en utilisant le même critère choisi adéquat pour la mesure du bon ajustement dans la période de calibrage [5].

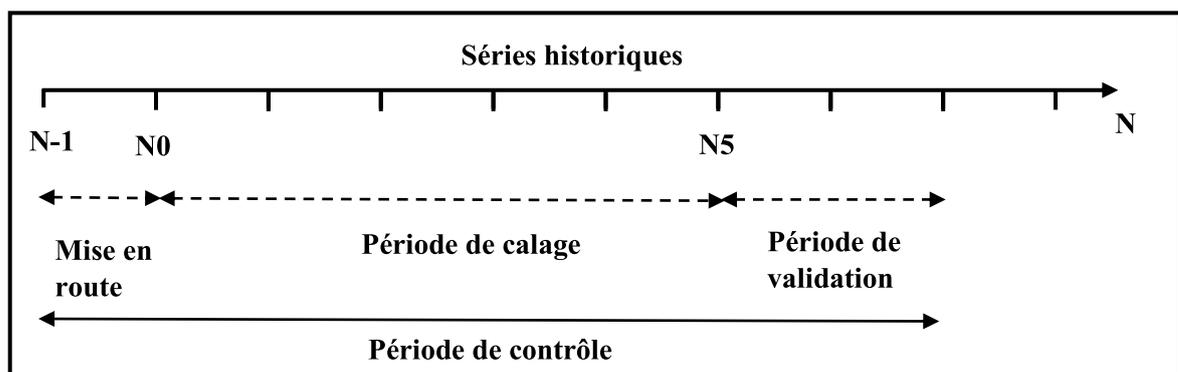


Figure 1.12 : Schéma de la chronologie du calage (Berthier, 2005)

3.4. Difficultés:

Il existe énormément de contraintes qui font face au modélisateur afin qu'il puisse assurer sa tâche de calibration et validation du modèle.

- Il faudrait s'assurer que les mesures de pluie, débit ou évapotranspiration sont de bonne qualité, sinon le comportement du modèle suivra une logique complètement erroné, ce qui conduit à des outputs erronés ou même à des difficultés liées à la calibration.
- Plus cette période représente une large gamme d'événements différents, plus le modèle sera représentatif du fonctionnement du bassin versant [5].
- La répartition de l'historique sur les périodes, mise en route, calibration et validation est cruciale, en particulier dans l'insuffisance de la longueur de la série historique [5].
- Les paramètres calibrés sont dépendants de la période choisie, n'offrant pas un jeu de paramètre unique représentatif des caractéristiques du BV.
- Le choix du type de calage du modèle (manuel ou automatique).
- Le choix adéquat de la fonction objective qui résume l'information de l'erreur entre les simulations et les observations.

4. Conclusion

Dans ce chapitre les techniques de calage des modèles pluie-débit qui sont soit manuelles, automatiques ou semi-automatiques ont été passées en revue. Le calage automatique se ramène généralement à un problème d'optimisation mathématique.

Les critères de performance sur lesquels repose le calage des modèles pluie-débit ont des structures différentes basées généralement sur le calcul de l'écart quadratique, sur le rapport ou l'écart absolu entre les observations et les simulations. Une corrélation entre les indices est mise en évidence. Les indices de performance n'arrivent à voir qu'un "côté" de la performance du modèle. Le calage automatique est basé sur les algorithmes d'optimisation. Ces algorithmes peuvent être classés en stochastique /déterministe, locaux/globaux, etc. Les algorithmes automatiques de calage à savoir le Shuffled Complex Evolution (SCE-UA) et l'Algorithme Génétique qui se sont montrés parmi les plus efficaces et efficients ont été passés en revue.

Chapitre II :

Matériels et Méthodes

Dans ce chapitre, l'objectif est de présenter de façon générale les outils utilisés pour effectuer le travail, entre autre, le logiciel d'étude, le langage de programmation, les données utilisées ainsi que l'approche méthodologique générale que nous avons choisie, en particulier pour tester et comparer les deux algorithmes d'optimisation globale (SCE et GA) sur des fonctions tests puis sur le meilleur modèle hydrologique qui s'adaptent à nos données et dont les résultats sont explicités dans le manuscrit du PFE de l'ingénieur. La particularité de cette étude réside premièrement dans le langage de programmation utilisé, le Langage R, qui est mis en œuvre pour la première fois à l'Ecole Nationale Polytechnique et à notre connaissance en Algérie. Ce langage de programmation simple mais profond commence à gagner en territoire en hydrologie au vue de la facilité de la manipulation des séries historiques et l'immense librairie de packages développés gratuitement en open source par les autres utilisateurs un peu partout dans le monde. La seconde particularité se résume dans la disponibilité du Package hydromad, destiné à la modélisation hydrologique en intégrant un nombre important de modèles globaux de pluie-débit. Les modèles intégrés dans le Package hydromad offrent une facilité de travail au vue du nombre minimes des variables de forçage qui se résument en la pluviométrie, et l'évapotranspiration. D'autre part, hydromad possède une panoplie de méthodes de d'optimisation locales et globales intégrées pour le calage automatique, entre autre la méthode du Shuffled Complex Evolution, Downhill-Simplex, et les algorithmes génétiques, qui forment l'objet de notre étude. En résumé, nous allons exposer le langage de programmation utilisé, le Package hydromad, les données ainsi que les caractéristiques de chaque série de données et le climat associé, la procédure de travail et la problématique envisagée. Le contenu des chapitres 1 servira comme support théorique sur tous ce qui va être présenté dans ce chapitre.

1. Matériels :

1.1. Le langage de programmation R

1.1.1. Origines :

Le logiciel R est un logiciel de statistique créé par Ross Ihaka & Robert Gentleman. Il est à la fois un langage informatique et un environnement de travail : les commandes sont exécutées grâce à des instructions codées dans un langage relativement simple, les résultats sont affichés sous forme de texte et les graphiques sont visualisés directement dans une fenêtre qui leur est propre. C'est un clone du logiciel S-plus qui est fondé sur le langage de programmation orienté objet S, développé par AT&T Bell Laboratories en 1988 [4]. Ce logiciel sert à manipuler des données, à tracer des graphiques et à faire des analyses statistiques sur ces données.

1.1.2. Pourquoi utiliser R ?

Tout d'abord R est un logiciel gratuit et à code source ouvert (open source). Il fonctionne sous UNIX (et Linux), Windows et Macintosh. C'est donc un logiciel multi-plates-formes. Il est développé dans la mouvance des logiciels libres par une communauté sans cesse plus vaste de bénévoles motivés.

Tout le monde peut d'ailleurs contribuer à son amélioration en y intégrant de nouvelles fonctionnalités ou méthodes d'analyse non encore implémentées. Cela en fait donc un logiciel en rapide et constante évolution.

C'est aussi un outil très puissant et très complet, particulièrement bien adapté pour la mise en œuvre informatique de méthodes statistiques. Il est plus difficile d'accès que certains autres logiciels du marché (comme SPSS ou Minitab par exemple), car il n'est pas conçu pour être utilisé à l'aide de « clics » de souris dans des menus. L'avantage en est toutefois double :

- l'approche est pédagogique puisqu'il faut maîtriser les méthodes statistiques pour parvenir à les mettre en œuvre ;
- l'outil est très efficace lorsque l'on domine le langage R puisque l'on devient alors capable de créer ses propres outils, ce qui permet ainsi d'opérer des analyses très sophistiquées sur les données

Le logiciel R est particulièrement performant pour la manipulation de données, le calcul et l'affichage de graphiques. Il possède, entre autres choses :

- un système de documentation intégré très bien conçu (en anglais) ;
- des procédures efficaces de traitement des données et des capacités de stockage de ces données ;

- une suite d’opérateurs pour des calculs sur des tableaux et en particulier sur des matrices ;
- une vaste et cohérente collection de procédures statistiques pour l’analyse de données;
- des capacités graphiques ’évoluées ;
- un langage de programmation simple et efficace intégrant les conditions, les boucles, la récursivité, et des possibilités d’entrée-sortie.

1.1.3. R et les statistiques

R est un logiciel dans lequel de nombreuses techniques statistiques modernes et classiques ont été implémentées. Les méthodes les plus courantes permettant de réaliser une analyse statistique telles que :

- statistique descriptive ;
- tests d’hypothèses ;
- analyse de la variance ;
- m’ méthodes de régression linéaire (simple et multiple) ;
- etc.

Sont enchâssées directement dans le cœur du système. Notez également que la plupart des méthodes avancées de statistique sont aussi disponibles au travers de modules externes appelés packages. Ceux-ci sont faciles à installer directement à partir d’un menu du logiciel. Ils sont tous regroupés sur le site internet du Comprehensive R Archive Network (CRAN) (<http://cran.r-project.org>) sur lequel vous pouvez les consulter. Ce site fournit aussi, pour certains grands domaines d’étude, une liste commentée des packages associés à ces thèmes (appelée Task View), ce qui facilite ainsi la recherche d’une méthode statistique particulière. Par ailleurs, une documentation détaillée en anglais de chaque package est disponible sur le CRAN.

Il est par ailleurs utile de noter que les méthodes statistiques les plus récentes y sont régulièrement ajoutées par la communauté statistique elle-même.

1.1.4. R et les graphiques

Une des grandes forces de R réside dans ses capacités, bien supérieures à celles des autres logiciels courants du marché, à combiner un langage de programmation avec la possibilité de réaliser des graphiques de qualité. Les graphiques usuels s’obtiennent aisément au moyen de fonctions prédéfinies. Ces dernières possèdent de très nombreux paramètres permettant par exemple d’ajouter des titres, des légendes, des couleurs, etc. Mais il est ’également possible d’effectuer des graphiques plus sophistiqués permettant de représenter des données complexes

telles que des courbes de surface ou de niveau, des volumes affichés avec un effet 3D, des courbes de densité, et bien d'autres choses encore. Il vous est également possible d'y ajouter des formules mathématiques. Vous pouvez aussi agencer ou superposer plusieurs graphiques sur une même fenêtre, et utiliser de nombreuses palettes de couleur.

1.1.5. R et l'Hydrologie

L'environnement gratuit pour l'analyse statistique et graphique qu'offre le logiciel R a été développé et est maintenu par des programmeurs statisticiens, avec le soutien d'une communauté croissante des utilisateurs avec de nombreux horizons différents, ce qui permet d'accéder à des techniques expérimentales et bien établies.

Les praticiens de la modélisation hydrologique passent énormément de temps dans la phase du prétraitement et le post traitement des données et résultats avec les instruments traditionnelles. Le R avec quelques bibliothèques associées sont présentés comme des outils puissants pour explorer et extraire des modèles à partir de l'information brute, pour prétraiter les données d'entrée des modèles hydrologiques et de post traiter leurs résultats.

Un Hydrologue devrait investir son temps à essayer R pour les raisons suivantes :

- Une panoplie d'algorithmes prêts à l'emploi
- Les fonctions, graphiques et les packages existant peuvent être facilement adaptés aux besoins particuliers.
- langage R est un nouvel environnement où la recherche et le développement sont le but principal
- Communauté d'utilisateurs importante et très active.
- R est principalement développé par ses utilisateurs, experts dans le domaine, et non par des programmeurs sans des connaissances en hydrologie (ce qui implique l'engagement et une participation à long terme, cependant des codes désordonnés)
- Ecrire une fois, utiliser plusieurs fois
- La documentation est disponible en plusieurs langues
- Multiplateforme (GNU/Linux, MacIS, Windows)
- Open Source
- Gratuit

Les avantages du langage R en hydrologie se distinguent dans les techniques suivantes :

- Analyse statistique et graphique
- Cadre adéquat et reproductible de modélisation
- Manipulation basique des données (importation, l'exportation, la sélection, joindre des tables)

- Interpolation spatiale
- Le calibrage et la simulation
- Analyse de l'incertitude
- Capacités géospaciales pour l'interrogation, la mise à jour, le partage et la visualisation des données, méthodes et résultats.
- Outils généraux que le soutien à l'analyse hydrologique
- Sorties MCMC
- Visualisation de la qualité d'impression
- Des séries chronologiques
- Données spatiales et spatio-temporelle
- Géostatistique, downscaling, upscaling,
- Agrégation et la désagrégation

R est un environnement précieux pour soutenir les étudiants du premier cycle et des niveaux supérieurs en hydrologie, puisqu'il est utile pour capturer les principales caractéristiques des données. C'est un langage de programmation flexible et entièrement fonctionnel, capable de faire interface des codes Fortran et C existants, et il est bien adapté aux exigences toujours croissantes de l'analyse hydrologique. Enfin, parce qu'il présente un environnement prometteur pour le suivi de la plupart des problèmes pratiques que pose la réalité pour le modélisateur hydrologique.

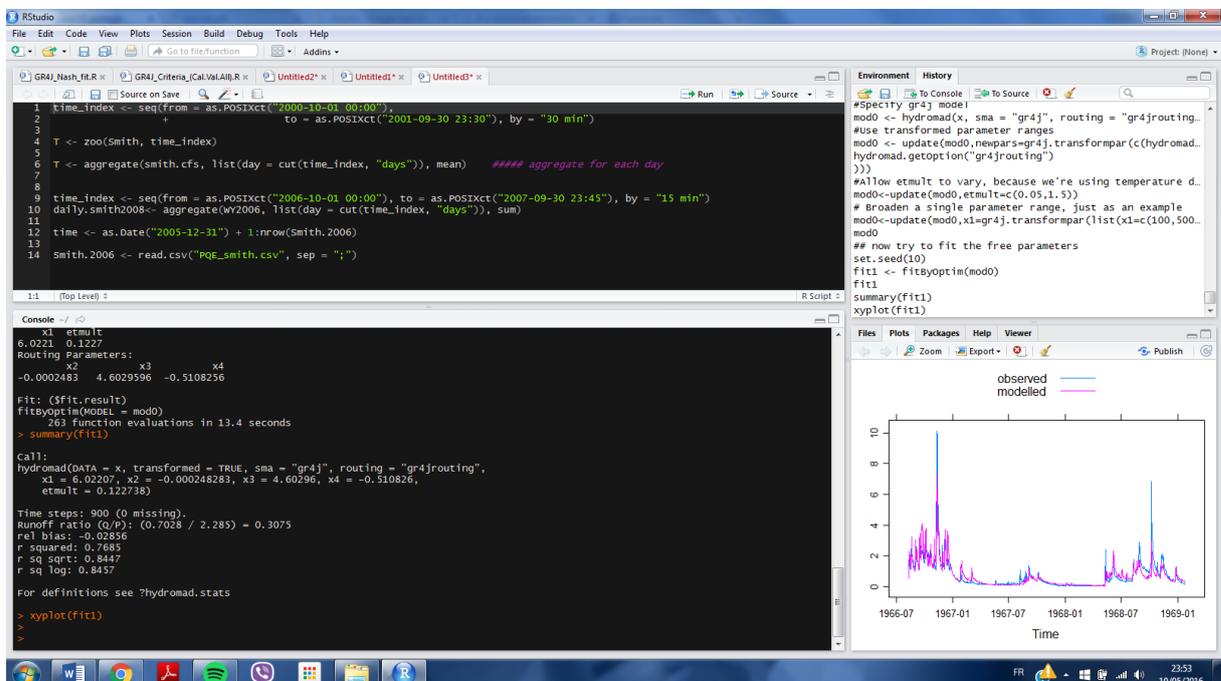


Figure 2.1 : L'interface du langage de programmation R (RStudio)

1.2. Machine de calcul

La machine utilisée pour effectuer les calculs de calage du modèle par l'algorithme du Shuffled Complex Evolution, est un PC personnel, un laptop de marque HP Pavilion g7, possédant un processeur Intel® Core™ i5-2450M CPU @2.50GHz, et une mémoire installée (RAM) de 4Go. La machine fonctionne sous le système d'exploitation Windows 7 (Edition Familiale Premium) avec un service pack 1 à 64 bits. La machine est opérationnelle depuis 2013, et l'appréciation de son état est moyenne. En générale, les performances de la machines n'ont pas autant d'importance que l'ennui dans le temps passé sur une seule opération de calage. Pour mesurer la performance de l'algorithme on observe généralement le nombre d'évaluation de la fonction objective afin de s'éloigner de cette dépendance de la machine.

1.3. Donnée :

1.3.1. Région de Canning, Australie

Les données dont on dispose représentent trois séries historiques de précipitations journalières, débits et évapotranspiration potentielle pour la rivière de Canning située à Scenic Drive en amont du barrage de Canning (Australie-Occidentale). La série historique s'étend sur 11 ans, à partir de 1977-01-01 jusqu'au 1987-12-31. La surface du bassin versant est de 517 kilomètres carrés. La rivière Canning est un affluent de la rivière Swan, et est située au Sud-Est de Perth, Australie Occidentale. Climatiquement, la région reçoit environ 900 millimètres de pluie par an, avec la plupart de ce qui tombe entre Mai et Septembre. Il existe une variabilité généralisée des précipitations dans ce bassin versant entre 700 et 1300 millimètres par an [10].

Les pluies, débits et ETP sont rapportés en millimètres (mm) au vue de l'homogénéisation et la bonne comparaison des grandeurs. Le débit journalier initialement en m³/s a été remis à l'échelle de hauteur en divisant par la surface du bassin versant. La qualité des données est indiscutable. Sur 11 ans, aucune lacune, ou aberration n'est enregistrée. Cette série accompagne le package hydromad afin de faciliter son apprentissage et sa manipulation [10].



Figure 2.2 : Positionnement de la région de Perth en Australie Occidentale.

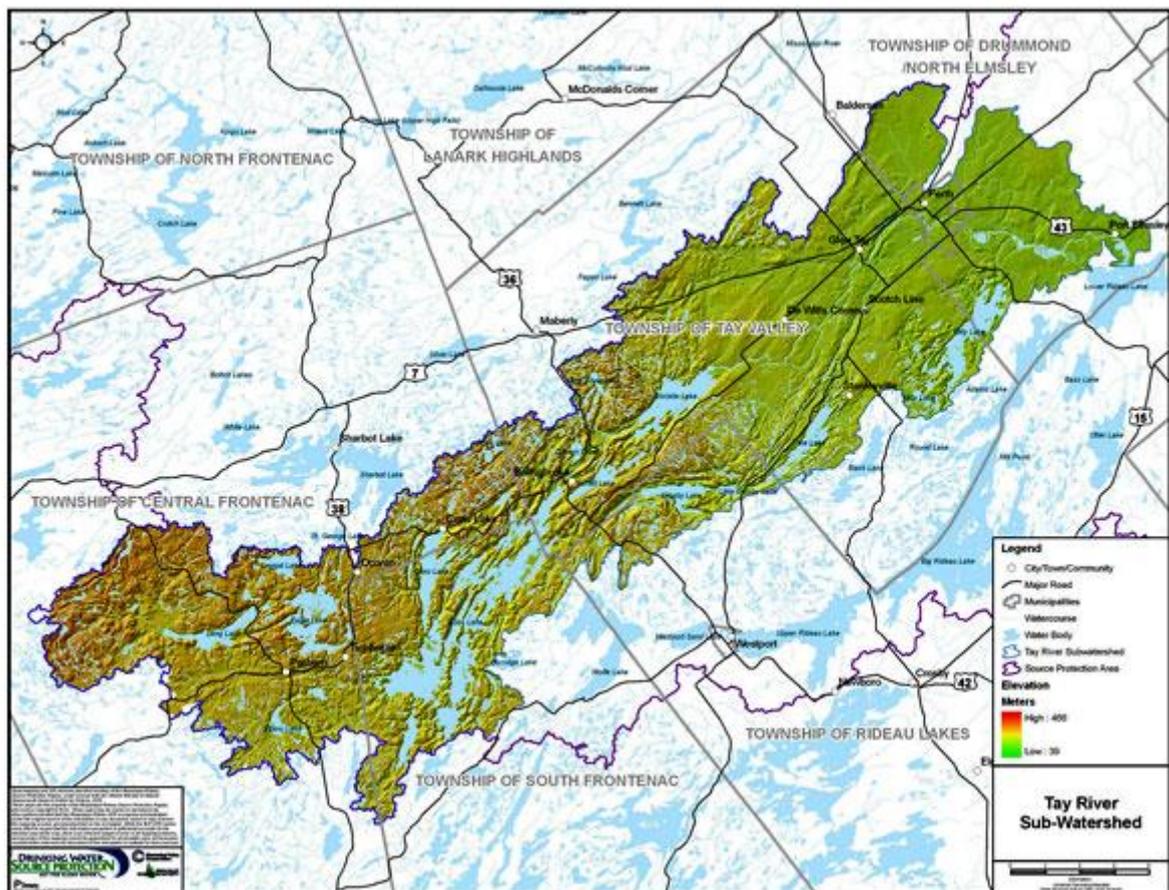


Figure 2.3 : Bassin versant de la région Canning, Australie

1. La pluviométrie (P) : précipitations zonale journalière (mm / jour), obtenue probablement par interpolation spatiale par Mr. Barry Croke.
2. Le débit (Q) : débits moyen journalier (mm / jour). Flux jauge ID 616024 "Canning rivière @ Scenic Drive". Latitude -33,4176; Longitude 115,9817.
3. L'évapotranspiration potentielle (E)

Pour plus de détails il faut prendre contact avec Mr. Barry Croke, manager de la communauté « hydromad users ».

La disposition graphique des données est visualisée dans la figure suivante :

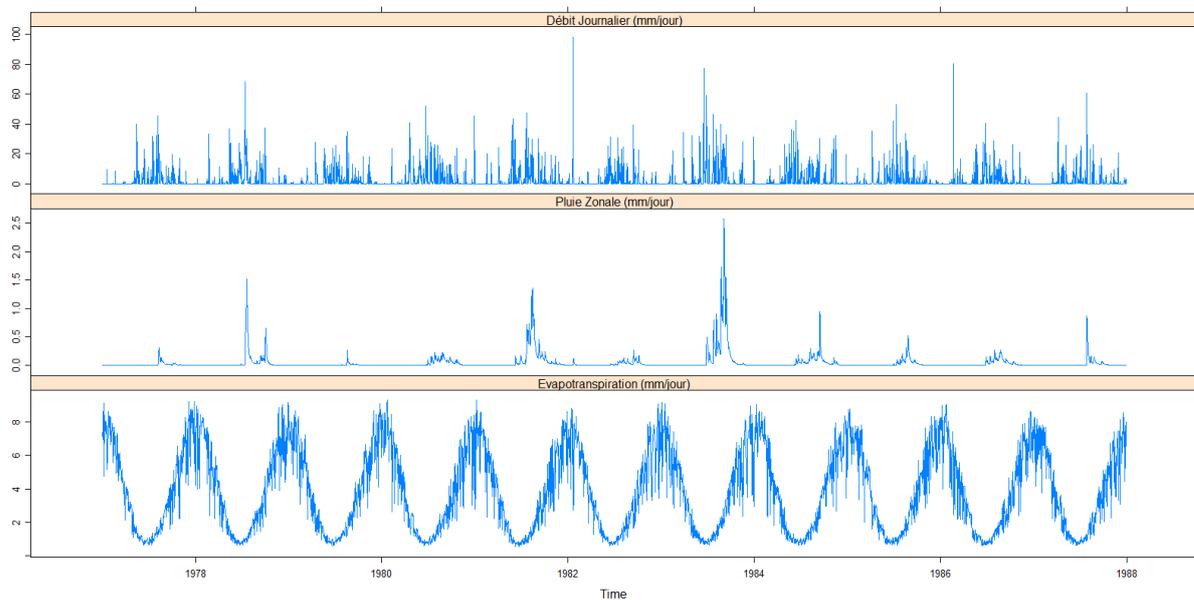


Figure 2.4 : Données hydrologique de la rivière de Canning, Australie

Tableau 2.1 :

Tableau résumant les statistiques simple des données de Canning, Australie

Statistique	Pluie (P)	Q (Débit)	ETP (E)
Min	0	0	0,530
1^{er} quantile	0	0	1,450
Médiane	0	0	3,43
Moyenne	2,449	0,04385	3,828
3^{ème} quantile	0,800	0,0250	6,010
Max	98	2,581	9,300

A partir de la **figure 4.5** on remarque immédiatement que la saison humide dans la région de Canning s'étend en général du mois de Mai jusqu'à Septembre, ce qui correspond bien avec la description du climat de la région. D'autre part, la plupart des débits mesurés sont des débits nuls. Ceci est remarqué à partir de l'histogramme des pluies ainsi que celui des débits vu que la dernière n'est que la réponse de l'autre. Evidemment plus de 50% des débits sont des débits nuls. Cependant dans la saison humide, la réponse de l'hydrogramme est une réponse en pic annuel généralement entre Mai et Septembre. Ceci correspond bien avec les climats méditerranéens semi-arides, similaire à celui de l'Algérie.

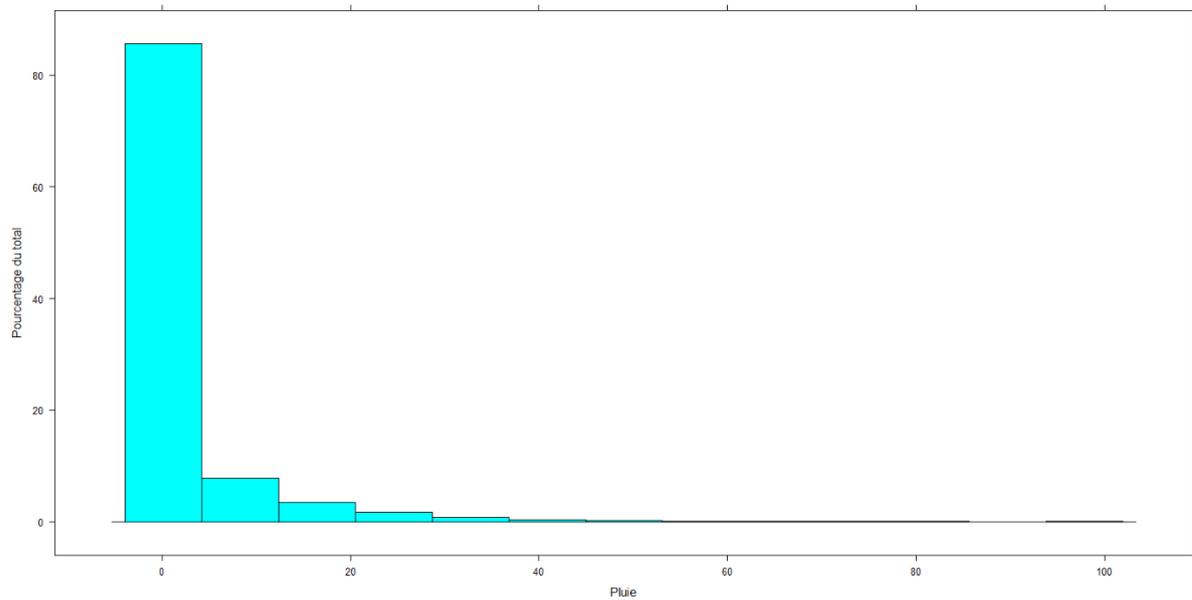


Figure 2.5.a : Histogramme des pluies journalières de la région de Canning, Australie

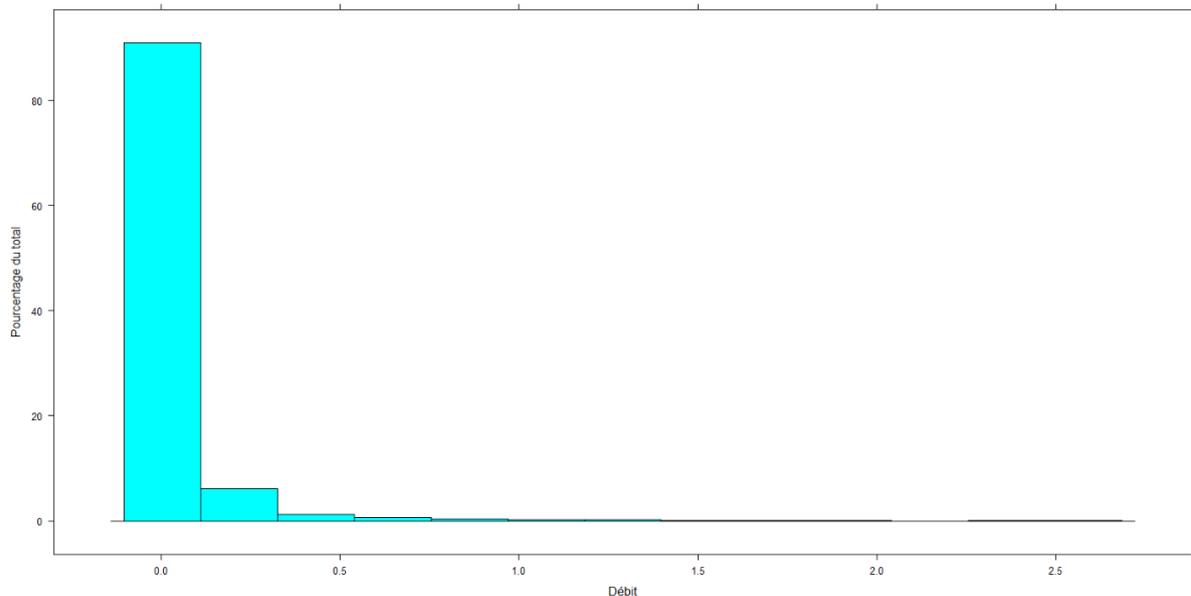


Figure 2.5.b : Histogramme des débits journaliers de la région de Canning, Australie

2. Méthodes

2.1. Conception de l'étude comparative de performance :

Un ensemble de sept problèmes de test a été choisi (**tableau**). Les six premiers d'entre eux sont des problèmes de tests bien établis de la littérature. Le septième est la combinaison des deux modèles pluie-débit (GR4J et IHACRES), qui ont fait leurs preuves pour simuler parfaitement les eaux de ruissellement dans la région de Canning, et dont structure contient les cinq caractéristiques des problèmes d'optimisation décrits dans le chapitre précédent. Chaque fonction a été ajustée de sorte que la valeur de la fonction partout dans l'espace réalisable est positive. Dans chaque cas, un point quelconque dont la valeur de la fonction objective est inférieure à 10^{-3} , est garanti pour se situer dans une très petite région autour de l'optimum global.

Tableau 4.2
Les problèmes tests utilisés dans cette étude

N°	Nom de la fonction	dim°	Nombre d'optimums	Optim
1	Goldstein	2	4	Min
2	Six-hump camelback	2	2	Min
3	Rastrigin	2	>50	Min
4	Ackley	4	>100	Min
5	Hartman	6	4	Min
6	Grienwank	10	1000	Min
7	Cwi-gr4jrouting	7	indénombrable	Max

Cette approche de test des deux algorithmes va les tester sur plusieurs épreuves de difficulté croissante. Ces difficultés sont les mêmes, mentionnées dans le chapitre 1, telles que l'existence des optimums locaux, la discontinuité des fonctions objectives, le non convexité...etc.

2.2. Les fonctions tests :

Les représentations mathématiques des problèmes tests sont présentées dans ce qui suit :

2.2.1. La fonction Goldstein-Price :

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

La fonction Goldstein-Price a plusieurs minima locaux

La fonction est généralement évaluée sur le carré de $x_i \in [-2, 2]$ pour tout $i = 1, 2$

Minimum global : $f(x^*) = 3$ à $x^* = (0, -1)$

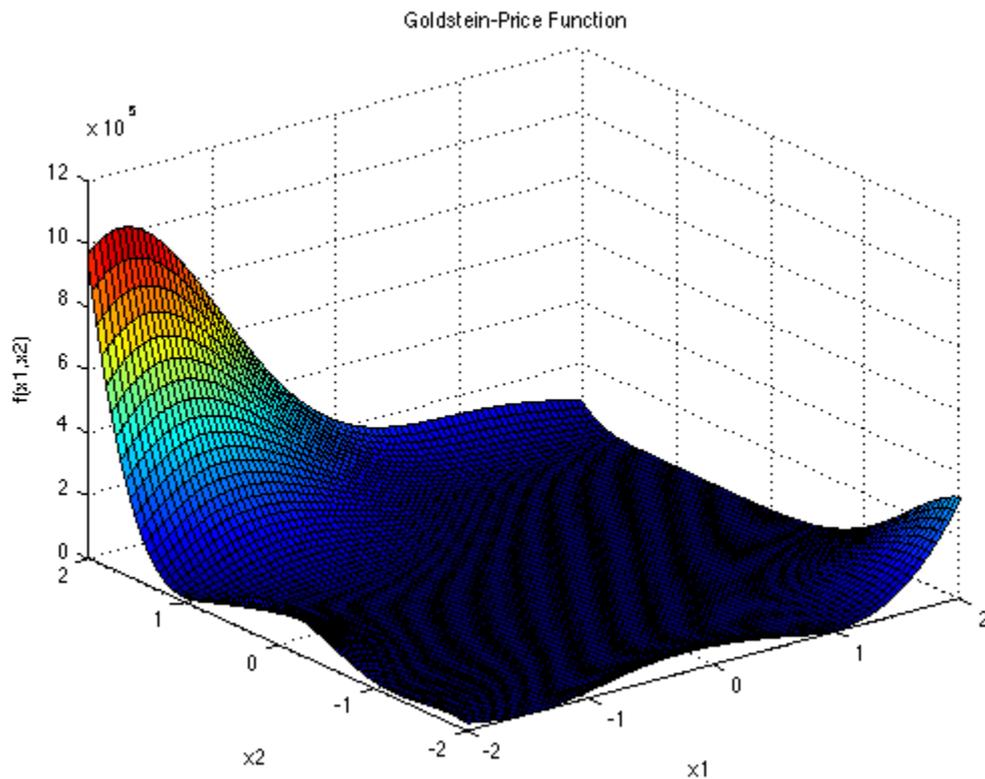


Figure 2.6.a : Représentation de la fonction Goldstein-Price sur son domaine de définition

2.2.2. La fonction Six-hump Camelback :

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

La courbe montre seulement une partie du domaine d'évaluation, afin de permettre une meilleure visualisation des caractéristiques clés de la fonction. La fonction dispose de six minima locaux, dont deux sont globales.

La fonction est généralement évaluée sur le rectangle de $x_1 \in [-3, 3]$, $x_2 \in [-2, 2]$

Minimum global : $f(x^*) = -1.0316$ à $x^* = (0.0898, -0.7126)$ et $x^* = (-0.0898, 0.7126)$

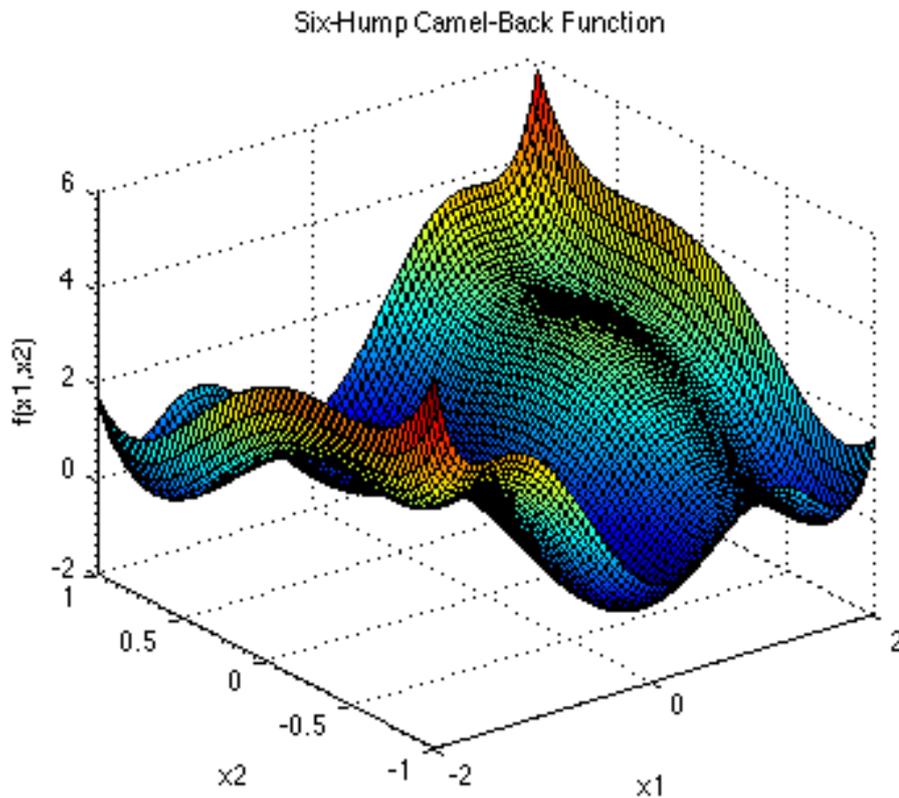


Figure 2.6.b : Représentation de la fonction Six-Hump Camelback sur son domaine de définition

2.2.3. La fonction Rastrigin :

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

La fonction Rastrigin a plusieurs minima locaux. Elle est très multimodale, mais les emplacements des minima sont régulièrement distribués comme montré dans **la courbe ()** dans sa forme à deux dimensions.

La fonction est généralement évaluée sur l'hypercube $x_i \in [-5,12 ; 5,12]$ pour $i = 1, \dots, d$

Minimum global : $f(x^*) = 0$ à $x^* = (0, \dots, 0)$

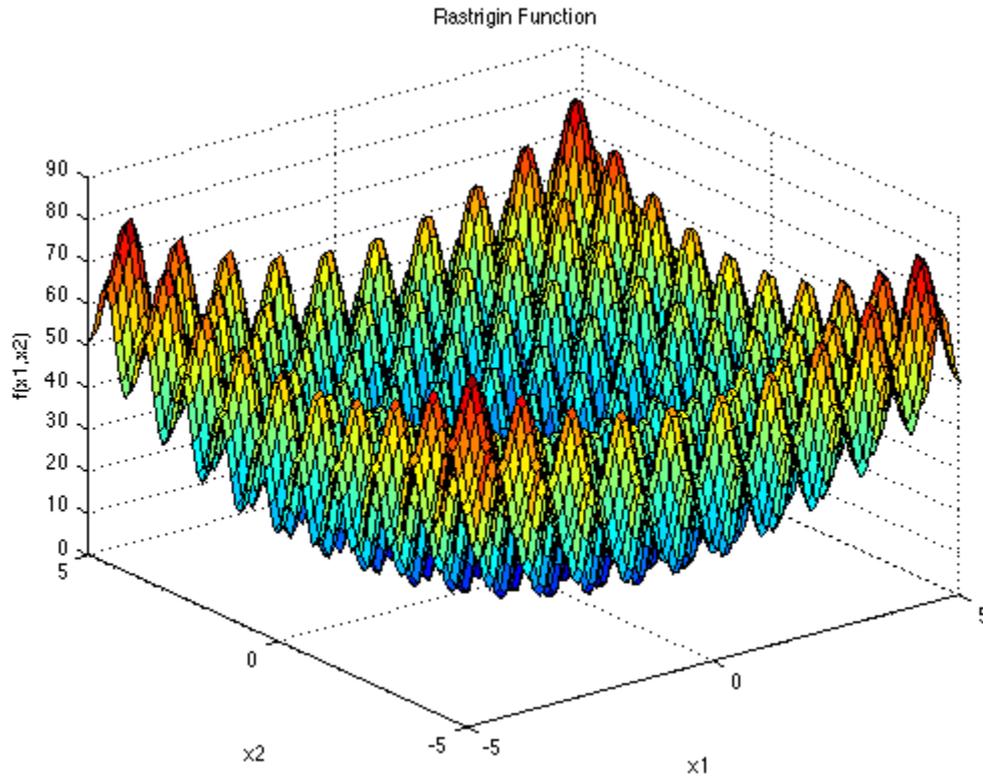


Figure 2.6.c : Représentation de la fonction Rastrigin sur son domaine de définition

2.2.4. La fonction de Ackley

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

La fonction Ackley est largement utilisée pour tester les algorithmes d'optimisation. Dans sa forme à deux dimensions, comme le montre le graphique, elle est caractérisée par une région extérieure presque plate, et un grand trou au centre. La fonction pose un risque pour les algorithmes d'optimisation, en particulier des algorithmes Hill climbing, pour être pris au piège dans un de ses nombreux minima locaux.

Les valeurs recommandées des paramètres : $a = 20$, $b = 0.2$, $c = 2\pi$

La fonction est généralement évaluée sur l'hypercube $x_i \in [-32.768 ; 32.768]$ pour $i = 1, \dots, d$

Minimum global : $f(x^*) = 0$ à $x^* = (0, \dots, 0)$

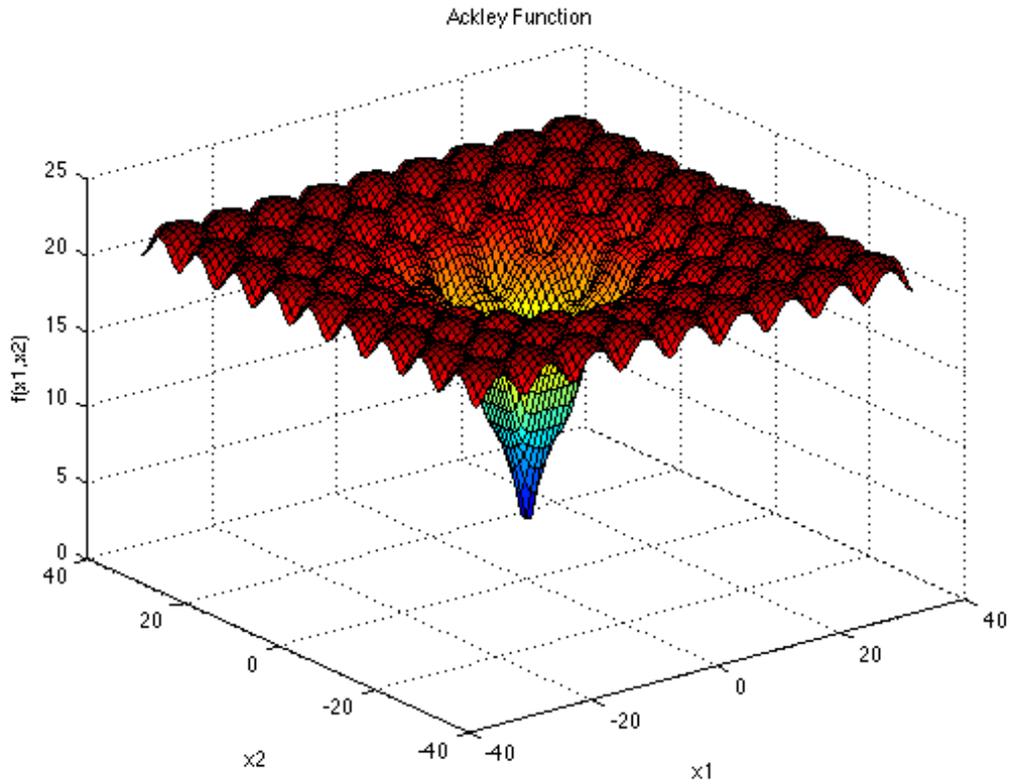


Figure 2.6.d: Représentation de la fonction de Ackley sur son domaine de définition

2.2.5. Hartmann 6D

$$f(x) = - \sum_{i=1}^4 \alpha_i \exp \left(- \sum_{j=1}^6 A_{ij} (x_j - P_{ij})^2 \right)$$

$$\text{où } \alpha = (1, 1.2, 3, 3.2)^T$$

$$A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$$

$$P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}$$

La fonction Goldstein-Price a six minima locaux

La fonction est généralement évaluée sur l'hypercube $x_i \in [0, 1]$ pour $i = 1, \dots, 6$

Minimum global :

$$f(x^*) = -3.042455 \text{ à } x^* = (0.20168, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573)$$

Il existe plusieurs variations dimensionnelles de cette fonction (3D, 4D, 5D et 6D). Nous avons choisi la fonction à 6 paramètres afin d'augmenter le degré de difficulté souhaité pour tester les deux algorithmes mais aussi pour suivre la continuité croissante des nombres de paramètres des problèmes tests.

2.2.6. Griewank :

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

La fonction Griewank a de nombreux minima locaux répandus, qui sont régulièrement distribués. La complexité est représentée dans les parcelles zoomées.

La fonction est généralement évaluée sur l'hypercube $x_i \in [-600 ; 600]$ pour $i = 1, \dots, d$

Minimum global : $f(x^*) = 0$ à $x^* = (0, \dots, 0)$

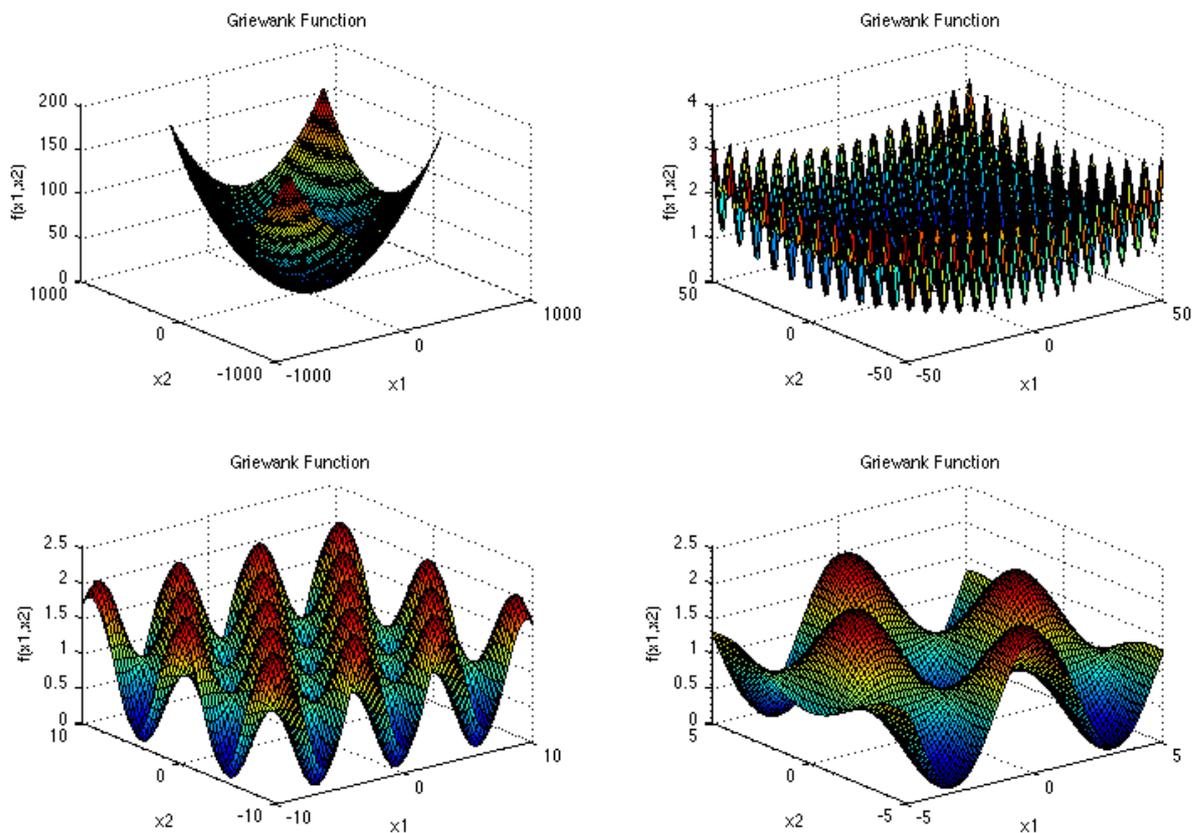


Figure 2.6.e : Représentation de quelques parcelles zoomées de la fonction Griewank

2.2.7. Modèle cwi-gr4jrouting :

Ce modèle qui a donné d'excellents résultats et qui s'ajuste presque parfaitement aux données de Canning, va être utilisé. Il va être calé sur la période 1977-1978, en prenant l'année 1977 comme période de mise en route. Le calage sera effectué sur le critère de Nash en le maximisant dans l'espace de paramètres suivant :

$$Nash = 1 - \frac{\sum_{i=1}^n (Q_{i,sim} - Q_{i,obs})^2}{\sum_{i=1}^n (Q_{i,obs} - \overline{Q_{obs}})^2}$$

Fonction de production « cwi »:

$$tw \in [min = 0 ; max = 100]$$

$$f \in [min = 0 ; max = 8]$$

$$l \in [min = 1 ; max = 1200]$$

$$p \in [min = 0.01 ; max = 5]$$

Fonction de routage « gr4jrouting » :

$$x_2 \in [min = -20 ; max = 10]$$

$$x_3 \in [min = 20 ; max = 300]$$

$$x_4 \in [min = 1.1 ; max = 5]$$

Le critère de Nash sur la période de 1978 atteint une valeur optimale de 99%, simulant ainsi parfaitement l'hydrogramme observé. La simulation est représentée graphiquement.

Pour plus de détails se référer au travail de l'ingénieur du même auteur.

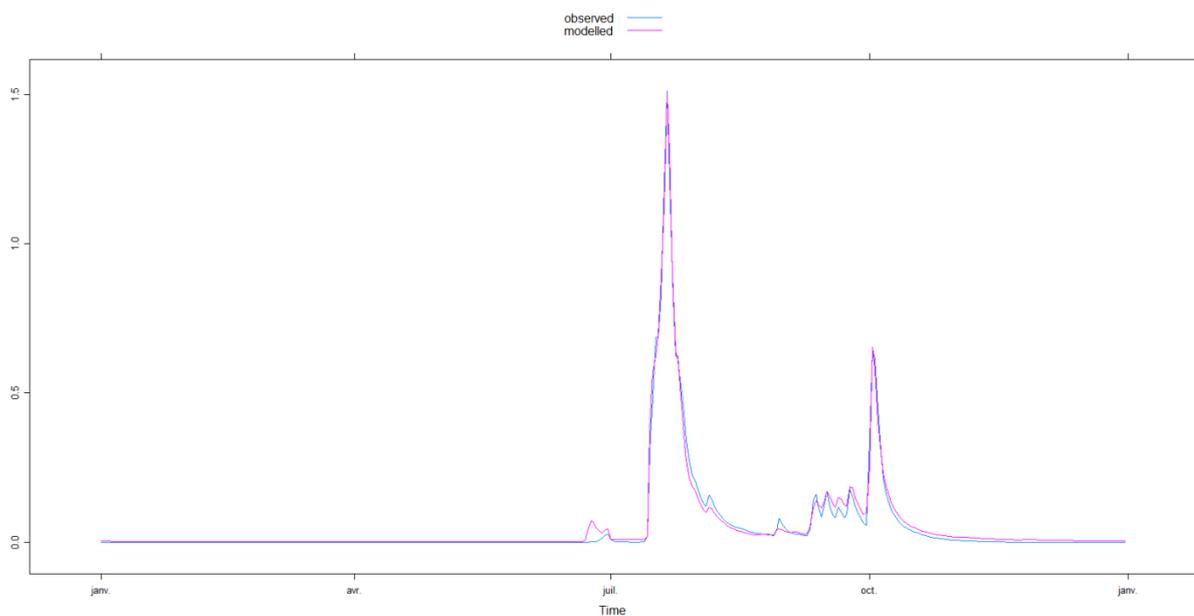


Figure 2.7 : Hydrogrammes observé et simulé sur la période 1978. (Canning, Australie)

2.3.Calcul

Deux algorithmes d'optimisation globale ont été testés et comparés à chacun des sept problèmes tests. Ceux-ci sont le Shuffled Complex Evolution (SCE-UA) et l'Algorithme Génétique. Chaque algorithme d'optimisation a été testé en exécutant 100 essais de la procédure sur chaque fonction test. Chaque essai commence avec un échantillon aléatoire indépendant de points, sélectionnés de manière uniforme de l'espace réalisable. Les critères d'arrêt sont les suivants:

- Dans chaque cas, un essai a été considéré comme un succès dès que la meilleure valeur de la fonction objective dans l'échantillon devient inférieure à 10^{-3} .
- Si pendant l'essai le nombre d'évaluation de la fonction objective atteint une grande quantité, ou la convergence des paramètres échantillonnés se produit sans réduire la meilleure valeur de la fonction objective en dessous de 10^{-3} , l'essai est considéré comme un échec.

Deux statistiques ont été enregistrées. La première statistique est le nombre de défaillances (NF) sur les 100 essais. En divisant NF par 100 donne une approximation de la probabilité d'échec et, par conséquent, de mesurer l'efficacité (robustesse) de l'algorithme. La seconde statistique est le nombre moyen d'évaluations de la fonction objective (AFE), (les cas échec sont inclus). Ceci donne une idée approximative de l'efficacité de l'algorithme. Un tracé d'une statistique contre l'autre donne une idée du lien entre l'efficacité et la robustesse inhérente à ces deux algorithmes probabilistes. Les faibles valeurs des deux statistiques indiquent un algorithme efficace et robuste. Chaque algorithme a été lancé pour ces valeurs suggérées par défaut dans la littérature :

Tableau 2.3:
Résumé des paramètres par défauts des deux algorithmes SCE et GA

Paramètres	SCE	Paramètres	GA
fnscale	1 ou -1	pcrossover	0.8
elitism	1	pmutation	0.1
initsample	« latin »	elitism	0.05 (5%)
reitol	1e-5	maxiter	5000
tolsteps	7	run	100
maxit	10000		
maxtime	Inf		
returnpop	Inf		
trace	1		

Pour le SCE:

ncomplex = m, ## nombre des complexes, m à préciser selon le cas
cce.iter = NA, ## nombre des itérations à l'intérieur de la boucle (CCE algorithme)
elitism = 1, ## contrôle les coefficient de l'échantillonnage
initsample ## méthode de l'échantillonnage des points initiaux
reltol = 1e-5, ## limite de convergence: facteur d'amélioration relative requis
tolsteps = 7, ## nombre d'itérations dans un reltol pour confirmer la convergence
maxit = 10000, ## nombre maximal d'itérations
maxeval = Inf, ## nombre maximal d'évaluation de la fonction objective
maxtime = Inf, ## durée maximale d'optimisation, en secondes
returnpop = FALSE, ## Afficher toutes les populations
trace = 1, ## Niveau de feedback, 1 veut dire affichage du feedback durant l'optimisation

Pour le GA :

popSize #### taille de la population
pcrossover #### la probabilité de croisement entre paires de chromosomes
pmutation #### la probabilité d'une mutation dans un chromosome parent
elitism #### le nombre de meilleurs individus de conditionnement physique pour survivre
 à chaque génération
maxiter #### le nombre maximum d'itérations à exécuter avant la recherche GA est
 interrompue
run #### le nombre de générations consécutives sans aucune amélioration dans le
 meilleur la valeur de remise en forme avant l'AG est arrêté.
maxFitness ##### la limite supérieure de la fonction de remise en forme après que la
 recherche GA est interrompue.

Les deux algorithmes ainsi, seront comparés en mesurant leurs efficience et efficacité qui mesurent tout simplement la vitesse de convergence et la robustesse d'atteindre l'optimum. Après les avoir testé sur les fonctions tests spécifiquement choisies pour leurs difficultés croissante, l'application des deux algorithmes sur le modèle hydrologique à 7 paramètres cwi-gr4grouting va être effectué tel que mentionné dans le manuscrit du travail de l'ingénieur. Il est cependant important de mentionner qu'un tel modèle à 7 paramètres ne nécessite pas obligatoirement l'utilisation d'algorithme d'optimisation globale pour les caler au vue du petit

nombre de paramètres à caler. Cependant, sa structure est assez complexe pour présenter tous les difficultés trouvées dans un modèle dont le nombre de paramètres est supérieur. Le résultat obtenu dans ce travail servira de preuve pour quel algorithme utiliser pour le calage des modèles hydrologiques.

Dans le prochain chapitre, les résultats obtenus vont être présentés et une discussion ainsi que des conclusions vont être tirés.

Chapitre III :

**Observations
et Discussion**

Dans ce chapitre, les résultats obtenus après l'exécution des méthodes explicitement mentionnées dans le chapitre précédents vont être rapportées, observées et par la suite analysées et discutées.

Les fonctions tests choisies pour mettre à l'épreuve des deux algorithmes d'optimisation, sont de difficultés croissantes et de caractéristiques différentes. Le modèle utilisé à la fin est issu de l'étude de l'ingénierie établie par le même auteur de ce manuscrit.

1. Les résultats de l'étude comparative de performance :

Les deux statistiques NF et AFE ont été calculées pour des tailles croissantes de l'échantillon initial pour l'algorithme génétique (GA), et pour un nombre croissant de complexes pour le Shuffled Complex Evolution (SCE-UA). Les résultats sont présentés dans les tableaux a-g et reportés graphiquement sur les figures a – g. Le trait en bleu correspond aux résultats obtenus avec le SCE et le trait en orange correspond à ceux obtenus par l'algorithme génétique.

Le nombre de points dans un complexe tel que suggéré par Duan, est calculé ainsi :

$$Nbr\ points = nbr\ complexes \times (2 \times dimension + 1)$$

Il est important de noter que le temps de calcul pour les différents problèmes de test croît exponentiellement. Pour les problèmes de 2 à 4 paramètres, le temps moyen pour toute l'opération ne dépasse pas 20 minutes. Pour les fonctions tests d'un nombre de paramètres élevé, les cents essais n'ont pas dépassé 1 heure 30. Par contre, Le dernier lancement du modèle hydrologique pluie débit est le plus coûteux. Dans un sens où l'ajustement individuel est effectuée en 20 minutes, mais les cents essais atteignent 16 heures de calcul pour un seul nombre de complexe. Pour cette raison nous avons réduit le nombre d'essais à 50 afin de pouvoir obtenir tous les résultats à temps.

Les courbes en bleu représentent les résultats du Shuffled Complex Evolution alors que les courbes en orange représentent les résultats obtenus par l'Algorithme génétique.

Tableau 3.a :
Résultats pour la fonction Goldstein-Price

Goldstein-Price						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	5	44	208	10	80	509
2	10	9	390	15	58	874
3	15	0	556	20	41	1067
4	20	1	738	25	38	1186
5	25	0	920	30	20	1747
6	30	0	1079	35	15	2162
7	35	0	1273	50	10	2339
10	50	0	1757	75	0	4728

Le SCE atteint une robustesse parfaite à 500 évaluations, ce qui correspond à 15 points de départ. Alors que l'AG atteint ce niveau de robustesse en commençant la recherche globale par une population de 75 points.

Le SCE nécessite pour ce problème moins de temps pour converger sûrement à l'optimum global.

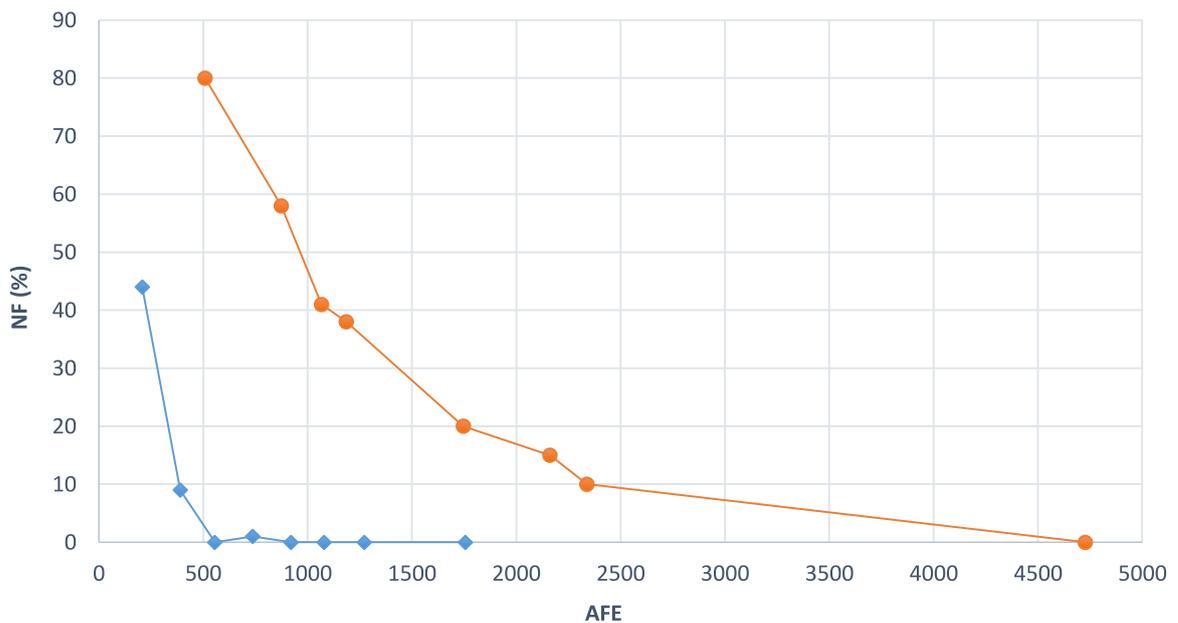


Figure 3.a : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Goldstein-Price

Tableau 3.b
Résultats pour la fonction Six-Hump Camelback

Six-Hump Camelback						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	5	21	188	10	33	521
2	10	2	340	15	14	802
3	15	0	507	20	11	962
4	20	0	677	25	9	1174
6	30	0	1008	30	3	1761
8	40	0	1333	35	3	2008
10	50	0	1623	40	2	2054
				50	1	2341

La même observation s'inscrit dans le cas de la fonction Six-Hump Camelback. Le SCE atteint une robustesse parfaite en initiant la recherche à 15 points (3 complexes). L'AG nécessite une distribution initiale moins dense pour atteindre la robustesse cela revient à l'aspect de la fonction qui ne contient pas beaucoup d'obstacle ou l'algorithme peut se piéger. C'est un test d'adaptation à la convexité.

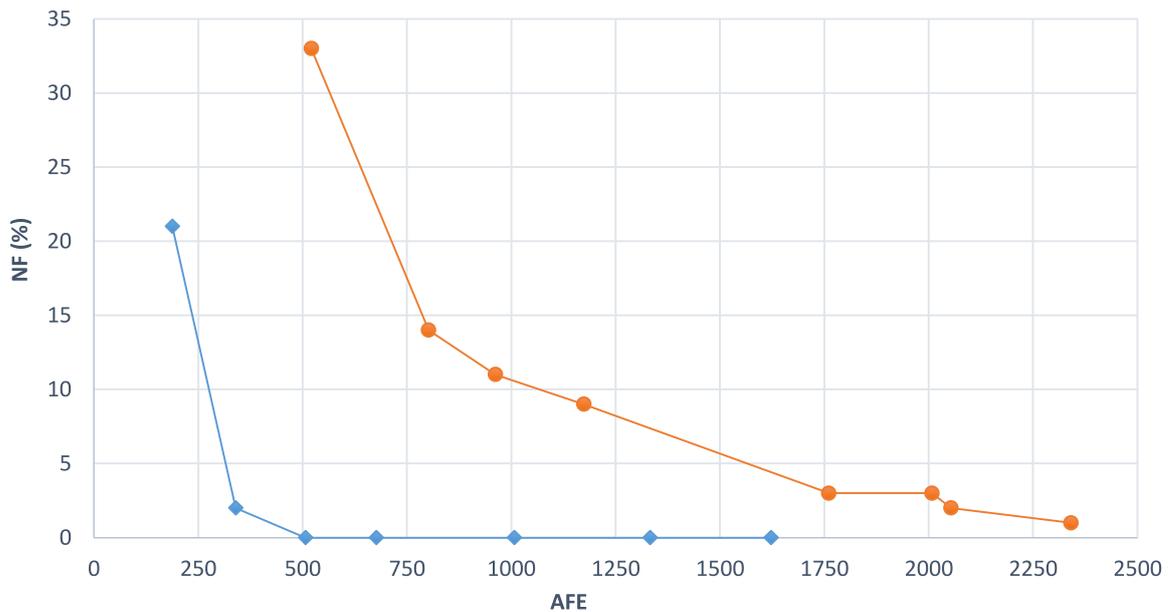


Figure 3.b : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Six-Hump Camelback

Tableau 3.c
Résultats pour la fonction Rastrigin

Rastrigin						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	5	94	221	10	87	524
2	10	74	485	15	70	881
3	15	57	812	20	53	1254
4	20	36	1227	25	53	1518
5	30	37	1508	30	20	2123
6	30	25	1866	35	18	2478
7	35	22	2220	40	9	2822
8	40	5	2706	50	10	3444
10	50	5	3314	60	1	4361
12	60	5	3950	70	2	5280

Cette Fonction de Rastrigin semble fournir plus de difficultés aux deux algorithmes par la distribution d'optima locaux dans l'espace de recherche. Les deux algorithmes ont une performance assez proche. Le SCE l'emporte sur l'AG lorsque le nombre de points initiaux est faible. Cependant, on remarque qu'à partir du huitième complexe la probabilité d'échec se stabilise à 5% alors qu'avec 10 points de plus l'AG atteint la robustesse parfaite.

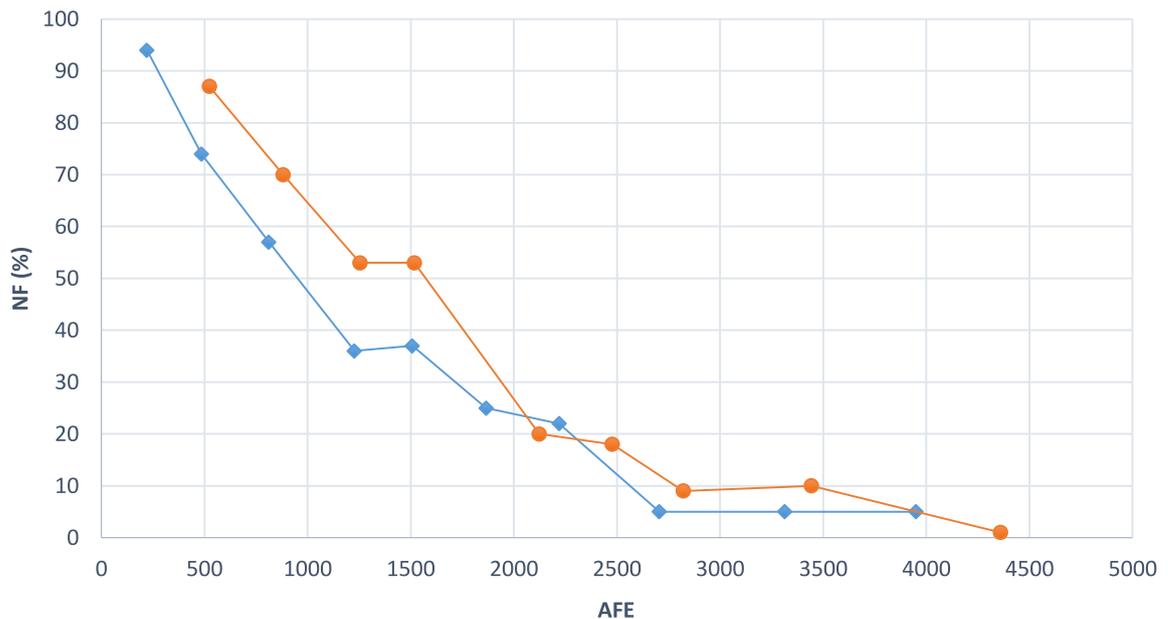


Figure 3.c : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Rastrigin

Tableau 3.d
Résultats pour la fonction de Ackley

Ackley						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	9	74	1031	5	100	3220
2	18	35	1967	10	45	8545
3	27	7	3282	18	5	24219
4	36	9	4224			
5	45	3	5436			
6	54	2	6572			
8	72	0	8812			

La fonction de Ackley, étant plus complexe au vue de sa dimension élevée mais aussi au nombre important d'optima locaux, offre plus de difficultés aux deux algorithmes de recherches. On voit clairement qu'à l'encontre des autres fonctions, Le SCE n'atteint la robustesse parfaite qu'après 9000 Evaluation, ceci est presque triplé pour l'AG, qui nécessite 25000 évaluations pour être robuste. Mais il est clair qu'avec peu de points initiaux le SCE atteint l'optimum plus surement que l'AG.

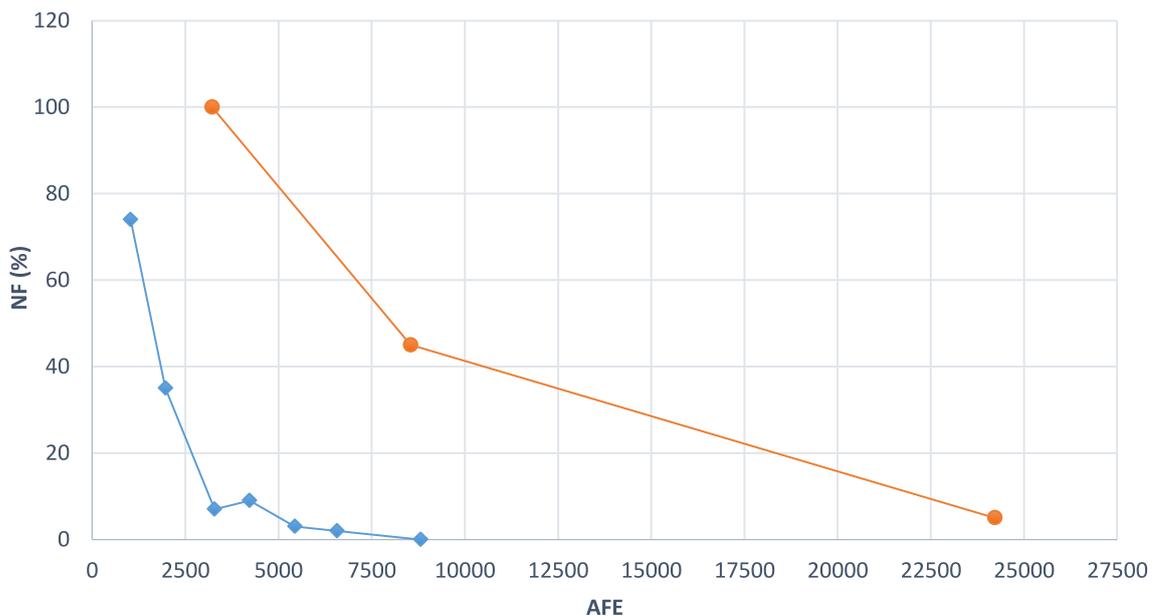


Figure 3.d: Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Ackley

Tableau 3.e
Résultats pour la fonction de Hartmann 6D

Hartmann 6D						
Nombre de complexes	SCE-UA			GA		
	points	NF (%)	AFE	points	NF (%)	AFE
1	13	43	883	13	34	4715
2	26	30	1479	26	27	7549
3	39	28	1830	39	28	14646
4	52	28	2155	52	18	19850
5	65	26	2480	65	22	23256
6	78	19	2871	78	24	30351
8	104	18	3769	104	11	38916
10	130	13	4539	130	15	48330
12	156	13	5435	156	10	58828
15	195	8	6794	195	17	76046
20	260	7	8695	260	5	104879
25	325	4	10590			

La fonction de Hartmann connue pour sa complexité liée non par à la distribution des optima locaux mais à la convexité et la super non-linéarité qui rend la tâche de recherche typiquement difficile, a conduit à observer la force de SCE qui arrive à atteindre 5% d'échec à 15000 évaluations seulement alors que l'AG dépasse 100000 évaluations pour atteindre le même résultat. D'autre part, on remarque qu'il a fallu un nombre très élevé de complexes et points initiaux pour atteindre ce résultat.

On note aussi que les résultats de l'AG aurait été pire si on n'avait pas modifié les paramètres de l'algorithme à 5000 générations et 100 répétitions.

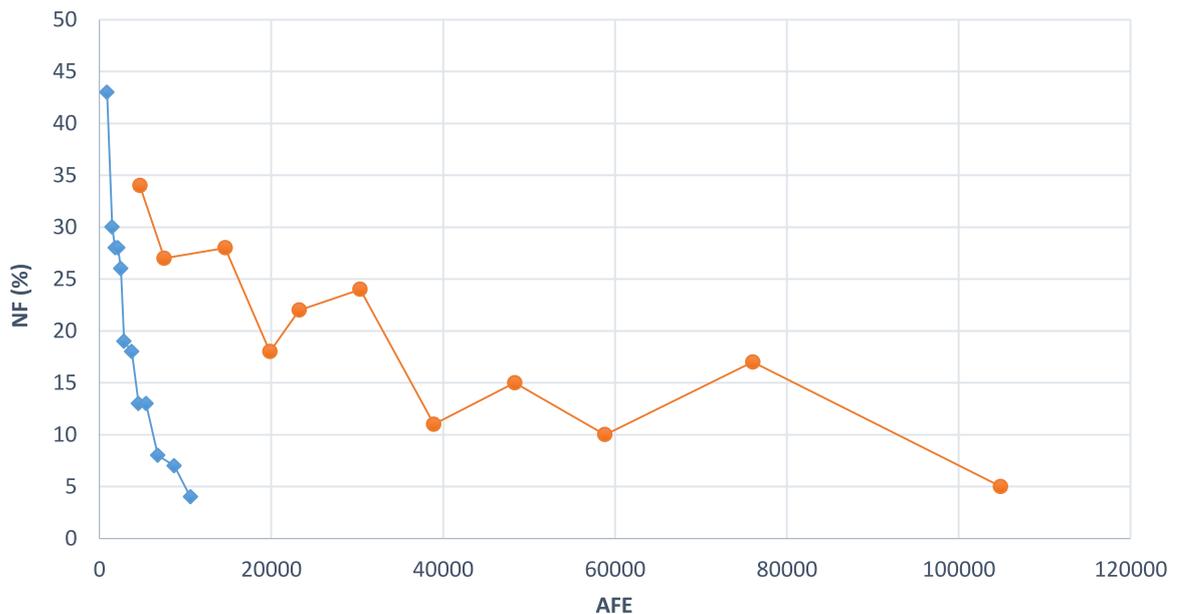


Figure 3.e : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour la fonction Hartmann 6D

Tableau 3.f
Résultats pour la fonction Griewank

Griewank						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	21	100	2561	21	100	7338
2	42	100	6947	42	100	10219
3	63	99	11139	525	97	100000
5	105	90	13031			
8	168	78	14897			
10	210	68	17960			
13	273	65	23170			
15	315	65	27033			
20	420	48	36947			
25	525	33	45968			

En atteignant ce problème de 10 paramètres avec un nombre d’optima locaux qui dépasse 1000, on remarque immédiatement la supériorité de SCE. Le SCE, ne commence à donner des résultats satisfaisant qu’à partir de 20 complexes ce qui se traduit à 40000 évaluations de la FO. L’AG étant pris aux paramètres améliorés par rapport aux paramètres par défaut est mis à l’échec à 97% à 100000 évaluations (525 points). Le résultat de l’AG peut être amélioré mais ça nécessite pleins de tentatives et de tâtonnement afin de trouver le meilleur jeu de paramètres initiaux de l’algorithme.

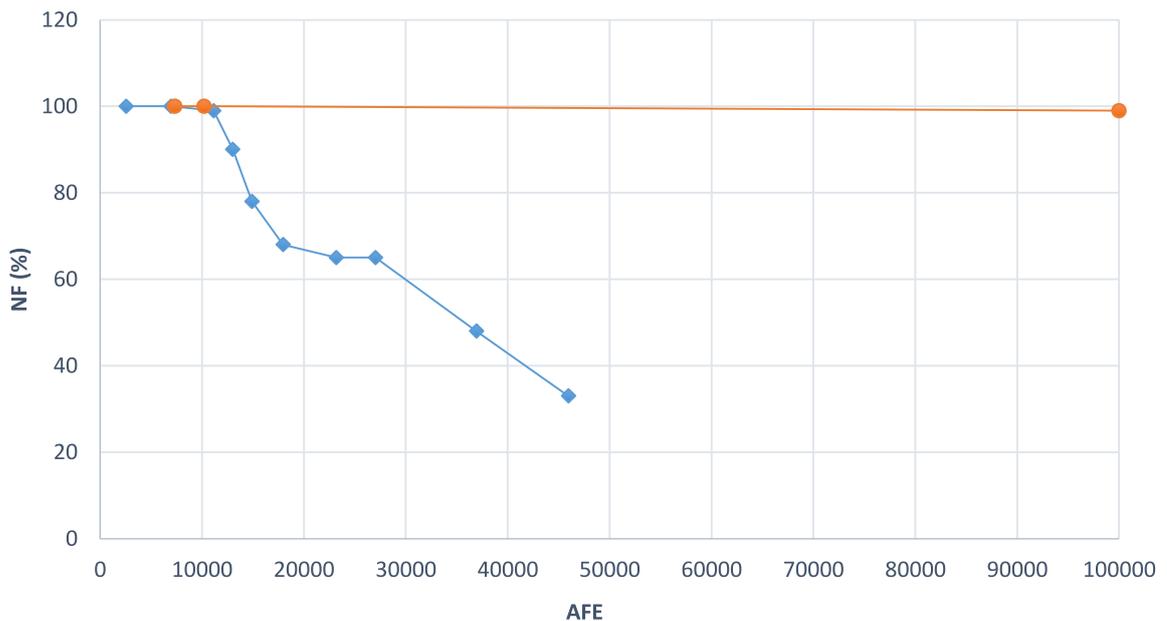


Figure 3.f: Evolution de la probabilité d’échec (NF%) en fonction de la moyenne du nombre d’évaluation de la fonction objective pour la fonction Griewank

Tableau 3.g
 Résultats pour le calage du modèle hybride sma = cwi, routing = gr4jrouting
 sur la période 1978 des données de Canning, Australie.

Modèle : IHacres(cwi)/gr4jrouting						
SCE-UA				GA		
Nombre de complexes	points	NF (%)	AFE	points	NF (%)	AFE
1	15	100	1094	15	100	40000
3	45	96	7016	30	60	65000
6	90	62	12536	50	0	100000
8	120	34	13232			
10	150	15	14017			
12	180	4	20708			

On observe immédiatement que le SCE a une meilleure performance qui assure une robustesse certaine avec moins d'évaluations de la FO. Cependant ce résultat a été assuré par moins de points initiaux.

Il est important de noter que les calculs pour cette partie était très lourds, et au lieu de faire le test sur 100 essais on s'est restreints à dix essais seulement, ce qui reste représentatif.

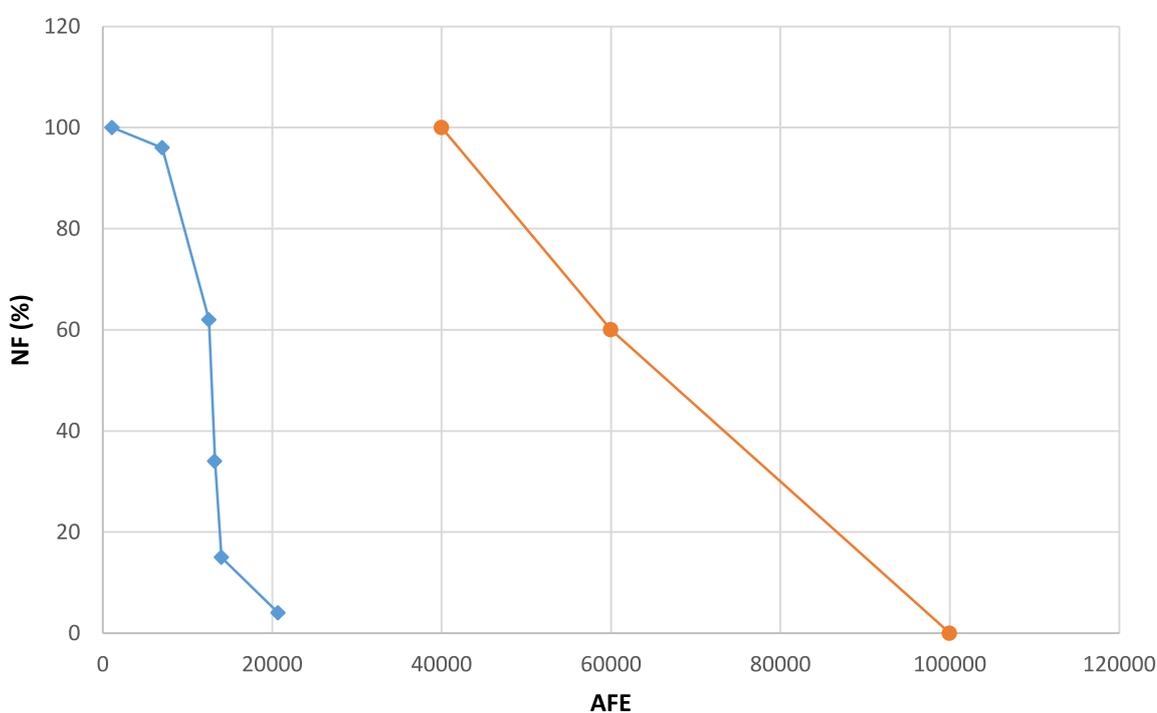


Figure 3.g : Evolution de la probabilité d'échec (NF%) en fonction de la moyenne du nombre d'évaluation de la fonction objective pour le calage du modèle hybride sma = cwi, routing = gr4jrouting sur la période 1978 des données de Canning, Australie.

2. Discussion :

Dans toutes les courbes, nous observons le comportement prédit, que l'efficacité augmente (les échecs diminuent) et l'efficiency diminue (moyenne d'augmentation des évaluations de la fonction) en augmentant la taille de l'échantillon (nombre de complexes pour le SCE et le nombre de la population initiale pour l'AG). Cependant, le comportement des deux algorithmes n'est pas linéaire. Dans certains cas, un taux d'échec nul a été obtenu avec les premiers ensembles initiaux d'essais et, par conséquent, d'autres essais n'étaient pas nécessaires. En outre, pour la fonction Griewank, l'algorithme génétique a complètement échoué, avec un taux d'échec de 100% pour les points initiaux et 97% pour les points finaux. Les caractéristiques importantes des résultats obtenus peuvent être résumées comme suit :

- L'algorithme de SCE est nettement supérieur sur tous les problèmes de test.
- L'Algorithme génétique a une performance variante dépendante des valeurs de ces paramètres initiaux.
- Le calcul de ces résultats a atteint 16 heures pour la fonction de Hartmann et Griewank. Et plus de 24 heures pour les 100 tentatives de calage du modèle cwi-gr4jrouting.
- Il est clair que plus la difficulté des problèmes tests augmente, plus le SCE donne de meilleurs résultats.
- Une comparaison des 100 résultats de l'algorithme SCE nous suggèrent que cet algorithme soit il converge soit il ne converge pas même pour le cas le plus faibles des complexes.
- L'algorithme génétique par contre peut atteindre un meilleur résultat si le critère d'arrêt est augmenté.
- La distribution uniforme des optimums locaux rend la tâche plus facile aux deux algorithmes, alors que pour les fonctions objectives des modèles hydrologiques, ces optimums sont distribués d'une manière non-uniforme.
- Le SCE arrive à caler le modèle cwi-gr4j avec plus d'efficiency en comparaison avec l'AG.
- Il est très difficile de trouver les meilleurs paramètres de l'AG et ça nécessite une connaissance profonde du problème et une expérience dans l'optimisation.
- Le SCE est très simple à utiliser et l'unique paramètre à gérer est le nombre de complexes.

Le résultat est clair que le SCE est meilleur que l'AG pour en considérant les paramètres par défaut des deux algorithmes. Cependant l'AG peut être amélioré mais cette tâche

nécessite énormément de temps et de connaissances, mais cela consomme pleins de temps, et on jouera au tâtonnement afin de déterminer les meilleurs paramètres de l'algorithme qui donne une une bonne performance de l'AG spécifique au problème. Cette difficulté écarte l'utilisation de l'AG pour les tâches qui nécessite un calage rapide des modèles, à l'exemple des modèles utilisés pour les alertes des crues.

Le SCE par contre est une procédure générale, géré initialement uniquement par le nombre de complexes. Le mélangeage des points ne permet pas la perte de l'information dans l'espace de recherche ce qui assure que l'algorithme ne soit pas piégé dans des optimums locaux. D'où une robustesse assurée.

En essayant d'augmenter les paramètres de l'AG on remarque que les cas que nous avons pris pour échec pouvait être améliorés si le critère d'arrêt (nombre max de générations) n'était pas atteint. Cela peut indiquer que l'AG est plus robuste au déprit d'un nombre très élevé d'évaluations.

Le modèle *cwi-gr4jrouting*, étant à 7 paramètres représente toute les difficultés reliés à l'optimisation (optima locaux, zones d'indifférence, insensibilité...). Cette étude peut être considérée comme référence pour les modèles à un nombre de paramètres élevé (où l'utilisation des algorithmes globaux est plus intéressante). Le SCE est le mieux adapté.

3. Conclusion :

Face à un problème donné, il existe certainement une procédure particulière qui est la plus efficace et efficiente à le résoudre. Vu sous cet angle, la procédure SCE est clairement prometteuse, affichant une performance constante sur toute la gamme des problèmes testés. Alors que les algorithmes génétiques luttent à trouver les solutions optimales le plus difficile le problème du test devient. Le problème de calage du modèle pluie-débit à 7 paramètres démontrent que le SCE est mieux adapté à franchir les difficultés connues à faire obstacles aux méthodes de recherche dans les problèmes d'optimisation.

CONCLUSION GENERALE

On a traité dans ce travail le problème du choix du meilleur algorithme d'optimisation parmi les plus connus et les plus cités dans la littérature scientifique, pour le calage automatique des modèles hydrologiques. La comparaison s'est faite entre les algorithmes génétiques, proposés en 1989 et représente la méthode la plus utilisées dans le domaine de l'optimisation, et le Shuffled Complex Evolution, une méthode récente spécifiquement développée pour le calage des modèles hydrologique et connu pour sa robustesse et efficacité. On s'est intéressé principalement à comparer l'efficience et l'efficacité des deux algorithmes sur un nombre de problème de test de difficulté croissante. Et par la suite une même application a été faite sur le modèle cwi-gr4jrouting jugé meilleur à reproduire les données de Canning de l'Australie Occidentale.

Les principaux résultats trouvés indiquent clairement que le SCE est le mieux performant pour tous types de problème. Principalement le problème du calage du modèle cwi-gr4jrouting pour lequel l'AG a nécessité pour la convergence quatre fois le nombre d'évaluation de la FO effectué par le SCE, et ce avec un taux d'échec considérable. Ce résultat est à retenir uniquement pour les paramètres par défaut des deux méthodes. La performance de l'AG peut donc être améliorée en modifiant les paramètres d'entrée de l'algorithme mais cela peut s'avérer une tâche encore plus difficile que celle de la recherche de l'optimum.

L'utilisation du SCE se confirme fortement pour le calage des modèles pluie-débit et l'efficience qu'il offre, permet son intégration non pas uniquement à la simulation mais aussi aux modèles de prévisions d'alertes contre les crues et les inondations par exemple. L'AG en contrepartie est mieux adapté pour les problèmes qui ne nécessitent pas une vitesse de calcul immense, entre autre les problèmes de gestion ou de prise de décision.

L'idée d'amélioration serait dans le développement d'une méthodologie pour la détermination guidée des paramètres initiaux de l'algorithme génétique afin de faciliter leur implémentation dans tous les aspects d'optimisation. Alors que pour le SCE la suggestion d'amélioration réside dans le choix de la meilleure fonction objective pour le calage des modèles pluie-débit. Même si celui-là peut tacler ce problème de complexité de la fonction objective mais pour une exploitation plus efficiente, il est très important de prêter attention à ça.

Références

- [1] Andrews F. T., Croked B. F. W., Jakeman A. J., « *An open software environment for hydrological model assessment and development* », Environmental Modelling and Software, Vol. 26, 2011.
- [2] Andrews F., « *Hydromad tutorial* », 2011.
<http://hydromad.catchment.org/downloads/tutorial.pdf>
- [3] Berthier, C.H., « *Quantification des incertitudes des débits calculés par un modèle pluie-débit empirique* », Mémoire de Master 2 en Hydrologie, Hydrogéologie et sols, encadré par M. Claude MICHEL, Université Paris Sud 11, 2005.
- [4] Dakhlaoui, H., « *Vers une Procédure de Calage Automatique plus Efficiente du Modèle HBV* », thèse de doctorat en hydraulique, sous la direction de Zoubeida Bargaoui, Tunis, Ecole Nationale d'Ingénieurs de Tunis, 2014.
- [5] 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*.
- [6] 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, 1994*.
- [7] Hydromad website, <http://hydromad.catchment.org/>
- [8] Jairo A.T., Edzer P., « State of R in Hydrological Modelling », 2nd OpenWater symposium and workshops, Institute for Geoinformatics, ifgi, University of Münster, Germany, 2013. http://ifgi.uni-muenster.de/~epebe_01/OpenWater2013_E01.pdf
- [9] Jeon J.H., Park C.G, and Engel B.A., « Comparison of Performance between Genetic Algorithm and SCE-UA for Calibration of SCS-CN Surface Runoff Simulation », *Water* 2014, 6, 3433-3456; doi:10.3390/w6113433.
- [10] JOHN R., « Non-Linear Genetic Algorithms for solving problems ». United States Patent 4, 935, 877. 1988.
- [11] Lafaye de Micheaux P., Drouilhet R., Liquet B., « *Le logiciel R : Maitriser le langage, Deuxième édition* », Edition Springer, 2014.
- [12] Moriasi D. N., Arnold J. G., Van Liew M. W., Bingner R. L., Harmel R. D., Veith T. L., « Model evaluation guidelines for systematic quantification of accuracy in watershed

- simulations », American Society of Agricultural and Biological Engineers, Vol. 50(3): 885–900, 2007.
- [13] Seong C., Her Y., and Benham B.L., « Automatic Calibration Tool for Hydrologic Simulation Program-FORTRAN Using a Shuffled Complex Evolution Algorithm », *Journal of Water*, Vol. 7, pp. 503-527, 2015.
- [14] TRAORE, V.B., SAMBOU, S., TAMBA, S., FALL, S., DIAW, A.T., CISSE. M.T., « *Calibrating the Rainfall-Runoff Model GR4J and GR2M on the Koulountou River Basin, a Tributary of the Gambia River* ». *American Journal of Environmental Protection*. Vol. 3, No. 1, 2014, pp. 36-44.
- [15] Ytoui, Y., « *Modélisation de la Relation Pluie-Débit aux Pas de Temps Mensuel et Journalier par les Modèles Conceptuels et Neuro-Flous (Application au bassin versant de l'ISSER)* », Projet de Fin d'Etudes pour Ingéniorat en Hydraulique, sous la direction du Pr. Abdelmalek Bermad, Ecole Nationale Polytechnique d'Alger, 2012.
- [16] Zambrano M.B., and Bellin, A., « R: A statistical environment for hydrological analysis», *Geophysical Research Abstracts* Vol. 12, 2010.
- [17] ZERARI N., « *Les Algorithmes Genetiques En Maintenance* », Mémoire de Magister. Université El Hadj Lakhdar Batna, 2006