



# THÈSE

## En vue de l'obtention du Diplôme de Doctorat en Sciences

Présentée par : *DAHMANI Saad*

*Optimisation de la gestion d'un réservoir à buts multiples*

*Faculté d'Architecture et de Génie Civil*

*Département : Hydraulique*

*Spécialité : Hydraulique*

*Option : Hydraulique*

*Devant le Jury Composé de :*

<i>Membres du Jury</i>	<i>Grade</i>	<i>Qualité</i>	<i>Domiciliation</i>
<i>CHERIF El-Amine</i>	<i>Professeur</i>	<i>Président</i>	<i>USTO-MB</i>
<i>YEBDRI Djilali</i>	<i>Professeur</i>	<i>Encadreur</i>	<i>USTO-MB</i>
<i>MANSOUR Hamidi</i>	<i>Professeur</i>	<i>Examineur</i>	<i>Univ. Oran 2</i>
<i>DRIS Mohammed El-Amine</i>	<i>Professeur</i>	<i>Examineur</i>	<i>Univ. Sidi Bel Abbès</i>

## ملخص :

الهدف من هذه الأطروحة هو معالجة مشكلة الإدارة التشغيلية المثلى لخزان متعدد الأغراض. حالة الدراسة المختارة هي سد حمام بوغرارة الواقع في ولاية تلمسان (شمال غرب الجزائر). تم الاعتماد على نهج التحسين العشوائي الضمني بغرض إدماج عشوائية التدفقات إلى الخزان. ركزت الدراسة على مناقشة مرحلة التنبؤ بالتدفقات الوافدة وكذا مرحلة وضع سياسات مثلى لإدارة الخزان.

أولاً، تم القيام بثلاث (03) تجارب للتنبؤ بتدفق ملئ الخزان على أساس ستة (06) نماذج انحدار تم تطبيقها على السلسلة الزمنية الشهرية للتدفق (من سبتمبر 1972 إلى ديسمبر 2016). الشبكات العصبية الاصطناعية متعددة الطبقات "ANNs" استخدمت لتجسيد نماذج الانحدار، ثم التنبؤ على مدى 120 شهراً. في التجربة الأولى، تم تدريب الشبكات العصبية بما يسمى الأساليب "الكلاسيكية" "BP" Backpropagation و "LM" Levenberg-Marquardt. في التجربة الثانية، تم تعليم نفس الشبكات باستعمال اثنين من خوارزميات البحث الشاملة التطورية، الأولى هي "خوارزمية وراثية ذات تشفير شفرة حقيقي RGA". أما الثانية فهي "الخوارزمية الهجينة من سرب الجزيئات ومحسن الذئب الرمادي HPSOGWO". أما الاختبار الثالث فقد ركز على اقتران الخوارزميات التطورية HPSOGWO و "خوارزمية البحث الجاذبية GSA" مع أساليب BP و LM التقليدية لإنتاج "الشبكات العصبية الاصطناعية التطورية EANNs". وقد استخدمت الخوارزميات الأولى بشكل منفصل لاختيار بارامترات هيكل الشبكات، بينما استخدمت الخوارزميات الكلاسيكية في تعليم شبكات الأعصاب الاصطناعية. أظهرت نتائج التجارب الأولى و الثانية ضعفا نسبيا في التعلم والتحقق، غير أن التجربة الثالثة أسفرت عن تحسن في التعلم و التسبب في الإفراط في ملائمة شبكات الأعصاب الاصطناعية في معظم الحالات.

ثانياً، تم تقييم أربعة من خوارزميات البحث التطورية الشاملة RGA و GSA و HPSOGWO و "مؤمئل الذئب الرمادي المحسن IGWO" في توليد سياسات إدارة الخزان المثلى من خلال دراسة مقارنة. تم استخدام الخوارزميات الأربعة لمعالجة نموذج تحسين مقيد مع دالة هدف غير خطية. في هذا الإطار تم استهداف تقليل العجز في التخصيص إلى أدنى حد ممكن من خلال تحسين الأداء، أين اعتبرت التدفقات الشهرية الصادرة من الخزان كمتغيرات قرار. لقد استخدمت ثلاثة سيناريوهات تدفق تم التنبؤ بها بواسطة شبكات عصبية اصطناعية جنباً إلى جنب مع نموذج وحيد لاستهلاك المياه الذي يوفره الخزان. أظهرت النتائج التي تم الحصول عليها تفوق خوارزمية RGA متبوعة بكل من GSA و HPSOGWO، و عدم ملائمة IGWO بسبب فشلها في إيجاد الحلول المثلى وهذا في حدود التجربة المنجزة.

**كلمات مفتاحية :** خزان، نموذج التحسين، التنبؤ، نموذج انحداري، الشبكات العصبية الاصطناعية،

التحسين المقيد، خوارزمية تطورية، سد حمام بوغرارة.

## **Résumé :**

L'objectif de cette thèse est de traiter la problématique d'optimisation de la gestion opérationnelle d'un réservoir multiusage. Le cas d'étude considéré est celui du Barrage Hammam Boughrara (Wilaya de Tlemcen, nord-ouest Algérien). L'approche d'optimisation stochastique implicite a été adoptée dans le but d'intégrer la stochasticité des apports dans le modèle d'optimisation. La phase de prévision des apports et celle relative à la génération des politiques de gestion optimale ont été traitées.

En premier lieu, trois (03) essais de prévision des apports du réservoir ont été élaborés à la base de six (06) modèles de régression appliqués sur la série chronologique mensuelle des apports du septembre 1972 à décembre 2016. Les réseaux de neurones artificiels "RNAs" multicouches ont été utilisés pour concrétiser la régression, puis la prévision sur une période de 120 mois. Dans le premier essai, les RNAs ont été entraînés par deux méthodes classiques (rétropropagation "BP" et Levenberg-Marquardt "LM"). Dans le deuxième essai, les mêmes RNAs ont été entraînés par deux algorithmes métaheuristiques évolutionnaires nommés "algorithme génétique à codage réel (RGA)" et "l'algorithme hybride d'essaim de particules et l'optimiseur de Loup-Gris (HPSOGWO)". Le troisième essai a été porté sur le couplage des métaheuristiques (HPSOGWO et recherche gravitationnelle (GSA)) avec les méthodes BP et LM, pour produire les "réseaux de neurones artificiels évolutionnaires (EANNs)". Les métaheuristiques ont été utilisées séparément pour le paramétrage structurel des RNAs, tandis que les méthodes classiques ont été employées dans l'apprentissage de ces derniers. Les résultats des deux premiers essais ont montré des corrélations relativement faibles dans l'apprentissage et la validation. Cependant, le troisième essai a surgi une amélioration de corrélation dans l'apprentissage en provoquant le sur-apprentissage des RNAs dans la plupart des cas.

En deuxième lieu, les quatre métaheuristiques RGA, GSA, HPSOGWO et l'optimiseur de Loup-Gris amélioré (IGWO) ont été évalués dans la génération des politiques de gestion optimales du réservoir à travers une étude comparative. Les quatre algorithmes ont été utilisés pour traiter un modèle d'optimisation sous contraintes avec fonction objectif non-linéaire. La minimisation des déficits d'allocations a été ciblée par l'optimisation, dans laquelle les lâchers mensuels sont considérés comme variables de décision. Trois scénarios d'apports prévus par les réseaux de neurones artificiels ont été utilisés à côté d'un modèle de consommation de l'eau desservie par le réservoir. Les résultats obtenus ont montré la supériorité de l'algorithme RGA, suivi par GSA et HPSOGWO respectivement, et l'inadéquation de l'IGWO à cause de son ratage des solutions optimales dans le contexte du paramétrage utilisé.

**Mots clés :** Réservoir, modèle d'optimisation, prévision, régression, réseaux de neurones artificiels, optimisation sous contraintes, algorithme évolutionnaire, barrage Hammam Boughrara.

## **Abstract :**

The aim of this thesis is to address the problem of multi-purpose reservoir management optimization. The considered case study is Hammam Boughrara dam (Tlemncen Wilaya, northwest Algeria). Implicit stochastic optimization approach was adopted to integrate inflow stochasticity into the optimization model. Reservoir inflow forecasting and optimal management policies generation were discussed.

Firstly, three (03) reservoir inflow forecasting trials were developed on the basis of six (06) regression models applied to the monthly inflow time series (from September 1972 to December 2016). Multi-layer Artificial Neural Networks "ANNs" were used to concretize the regression models, then the prediction over a period of 120 months. In the first trial, ANNs were trained by two classical methods (Back-Propagation "BP" and Levenberg-Marquardt "LM"). In the second, the same ANNs were trained by two evolutionary metaheuristic algorithms called "Real-coded Genetic Algorithm (RGA)" and "Hybrid Algorithm of Particle Swarm Optimizaion and Grey Wolf Optimizer (HPSOGWO)". The third trial focused on the coupling of metaheuristics (HPSOGWO and Gravitational Search Algorithm (GSA)) with BP and LM, in order to produce "Evolutionary Artificial Neural Networks (EANNs)". Metaheuristics were used separately for ANNs structural parameterization, while classical methods were used to their training. Results of the first and second trials showed relatively low correlations in learning and validation steps. However, the third trial came up with an improvement in learning correlation by causing ANNs overfitting in the most cases.

Secondly, the four metaheuristics RGA, GSA, HPSOGWO and Improved Grey Wolf Optimizer (IGWO) were evaluated in the generation of reservoir optimal management policies through a comparative study. The four algorithms were used to process a constrained optimization model with non-linear objective function. Allocation deficits minimization has been targeted through optimization, in which monthly releases are considered as decision variables. Three inflow scenarios forecasted by artificial neural networks were used alongside a model of water consumption served by the reservoir. The obtained results showed the superiority of RGA, followed by GSA and HPSOGWO respectively, and the inadequacy of IGWO because of its optimal solutions lack in the context of the settings used.

**Keywords :** Reservoir, optimization model, forecasting, regression, artificial neural networks, constrained optimization, evolutionary algorithm, Hammam Boughrara dam.

# Remerciements

J'aimerais tout d'abord remercier vivement, mon directeur de thèse **Pr. Djilali YEBDRI** *professeur à l'université des sciences et de la technologie d'Oran - Mohamed-Boudiaf- (USTO-MB)*, pour sa patience, disponibilité et surtout ses judicieux conseils, qui ont contribué à améliorer le travail jusqu'aux derniers moments. Nonobstant, sa relecture finale méticuleuse de chacun des chapitres m'a sans aucun doute permis de concrétiser mon travail et le conduire aux objectifs.

Je tiens à remercier considérablement les membres du jury d'avoir accepté la révision et l'évaluation de ce présent travail, **Pr. CHERIF El-Amine** *professeur à l'université des sciences et de la technologie d'Oran - Mohamed-Boudiaf- (USTO-MB)*; **Pr. MANSOUR Hamidi** *professeur à l'université Oran 2*; et **Pr. DRISS Mohammed El-Amine** *professeur à l'université Djillali Liabès de Sidi Bel Abbès*.

Je tiens à remercier également tous mes collègues du laboratoire L.G.T.E (USTO-MB).

Je tiens à remercier finalement mes parents et ma femme pour leurs soutiens constants et leurs encouragements.

# Dédicace

*À la mémoire de mon grand-père*

*À ma mère et mon père*

*À Roua-Hanine, Yasmine et Loudjayne*

*À mon frère et mes sœurs*

*À Wail-Abd Elsamad, Hatim-Abd Elbari, Maya, Belkais et Layane*

*J'ai pris de votre temps pour réaliser ce travail.*



# Table des matières

Résumé en langue Arabe	i
Résumé	ii
Abstract	iii
Remerciements	iv
Dédicace	v
Liste des tableaux	xv
Liste des figures	xix
Liste des abréviations	xxv
Introduction générale	1

## **Partie I L'optimisation de la gestion d'un barrage réservoir, les métaheuristiques et les réseaux de neurones artificiels** 7

<b>Chapitre 1: Optimisation de la gestion des réservoirs</b>	<b>9</b>
1.1 Introduction . . . . .	9
1.2 Nécessité et rôles des barrages – réservoirs . . . . .	10
1.3 Gestion opérationnelle des barrages-réservoirs . . . . .	12
1.3.1 Règles de gestion de la ressource en eau . . . . .	14
1.3.1.1 Règles de répartition entre usages . . . . .	15
1.3.1.2 Stratégie de distribution temporelle . . . . .	17
1.4 Approches d'optimisation de la gestion opérationnelle d'un réservoir . . . . .	19
1.4.1 Approche d'optimisation stochastique implicite . . . . .	19
1.4.2 Approche d'optimisation stochastique explicite . . . . .	21
1.4.3 Approche de paramétrage-simulation-optimisation . . . . .	22
1.5 Modèle général d'optimisation du fonctionnement d'un réservoir . . . . .	23
1.5.1 La fonction objectif . . . . .	23
1.5.2 Contraintes du système . . . . .	25
1.5.2.1 Équilibre de masses "Bilan hydrique" . . . . .	25
1.5.2.2 Contraintes de stockage . . . . .	26
1.5.2.3 Contraintes de lâchers . . . . .	26
1.5.2.4 Contrainte de déversement . . . . .	27

1.6	Modèle d'optimisation appliqué dans le cadre de l'étude . . . . .	27
1.6.1	Application des contraintes . . . . .	27
1.6.2	Fonction objectif choisie . . . . .	28
1.7	Méthodes d'optimisation appliquées : État de l'art . . . . .	29
1.7.1	La programmation linéaire . . . . .	29
1.7.2	La programmation non-linéaire . . . . .	31
1.7.3	La programmation dynamique . . . . .	31
1.7.3.1	La programmation dynamique déterministe . . . . .	33
1.7.3.2	La programmation dynamique stochastique . . . . .	33
1.7.4	Méthodes de calcul intelligent . . . . .	35
1.7.4.1	Théorie des ensembles flous . . . . .	35
1.7.4.2	Réseaux de neurones artificiels . . . . .	35
1.7.4.3	Algorithmes d'optimisation évolutionnaires . . . . .	36
1.8	Conclusion . . . . .	36
<b>Chapitre 2: Les algorithmes métaheuristiques</b>		<b>39</b>
2.1	Introduction . . . . .	39
2.2	Généralités sur l'optimisation . . . . .	40
2.3	Classification des méthodes d'optimisation . . . . .	42
2.4	Choix d'une méthode d'optimisation . . . . .	42
2.5	Les métaheuristiques . . . . .	45
2.5.1	Définition . . . . .	45
2.5.2	Principe de fonctionnement des métaheuristiques . . . . .	46
2.5.3	Domaines d'application . . . . .	47
2.5.4	Classification des métaheuristiques . . . . .	47
2.5.5	Pourquoi l'usage des métaheuristiques ? . . . . .	49
2.5.6	Inconvénients des métaheuristiques . . . . .	51
2.6	Les métaheuristiques utilisées dans le cadre de l'étude . . . . .	52
2.6.1	Algorithme génétique à codage réel (RGA) . . . . .	52
2.6.1.1	Principe et algorithme . . . . .	53
2.6.2	Algorithme de recherche gravitationnelle (GSA) . . . . .	56
2.6.2.1	Principe et algorithme . . . . .	57
2.6.3	Algorithme d'optimiseur de Loup Gris amélioré (IGWO) . . . . .	61
2.6.3.1	Principe et algorithme . . . . .	61
2.6.4	Algorithme hybride HPSOGWO . . . . .	64
2.6.4.1	Principe et algorithme . . . . .	66
2.7	Conclusion . . . . .	67
<b>Chapitre 3: Les réseaux de neurones artificiels</b>		<b>69</b>
3.1	Introduction . . . . .	69

3.2	Historique . . . . .	70
3.3	Principe du fonctionnement d'un neurone formel . . . . .	72
3.3.1	La fonction d'activation . . . . .	73
3.4	Réseau de neurones multicouche . . . . .	74
3.4.1	Principaux avantages des RNAs multicouche . . . . .	75
3.5	Architectures des réseaux de neurones artificiels . . . . .	75
3.5.1	Les réseaux de neurones statiques . . . . .	76
3.5.2	Les réseaux de neurones dynamiques . . . . .	76
3.6	Modèle mathématique des réseaux de neurones . . . . .	76
3.7	Apprentissage des réseaux de neurones . . . . .	79
3.7.1	Apprentissage non supervisé . . . . .	79
3.7.2	Apprentissage supervisé . . . . .	79
3.7.3	Principe d'apprentissage supervisé des réseaux de neurones . . . . .	80
3.8	Méthodes classiques d'apprentissage des ANNs . . . . .	81
3.8.1	Méthode de rétropropagation du gradient de l'erreur (Back-propagation) . . . . .	82
3.8.2	Méthode de Levenberg-Marquardt . . . . .	83
3.8.3	Inconvénients des méthodes classiques d'apprentissage . . . . .	83
3.9	Entraînement des RNAs par les métaheuristiques . . . . .	84
3.9.1	Entraînement des RNAs par l'algorithme génétique RGA . . . . .	85
3.9.2	Entraînement des RNA par l'algorithme HPSOGWO . . . . .	87
3.10	Conclusion . . . . .	88
<b>Chapitre 4: Les réseaux de neurones artificiels évolutionnaires</b>		<b>89</b>
4.1	Introduction . . . . .	89
4.2	Problématique . . . . .	90
4.3	Recherche d'un RNA proche-optimal . . . . .	91
4.3.1	Les métaheuristiques pour la recherche d'un RNA proche-optimal . . . . .	92
4.3.2	Fonction objectif . . . . .	94
4.3.3	Codage des solutions . . . . .	95
4.4	Conclusion . . . . .	96
<b>Partie II Optimisation de la gestion opérationnelle d'un réservoir à usages multiples : Cas du Barrage Hammam Boughrara</b>		<b>97</b>
<b>Chapitre 5: Présentation de la zone d'étude</b>		<b>99</b>
5.1	Introduction . . . . .	99
5.2	Présentation du bassin versant de la Tafna . . . . .	100
5.2.1	Situation géographique . . . . .	100
5.2.2	Les barrages situés dans le bassin de la Tafna . . . . .	101
5.3	Le bassin versant d'Oued Mouilah . . . . .	102

5.3.1	Localisation géographique de bassin versant d'Oued Mouilah . . . . .	103
5.3.2	Morphologie de bassin versant d'Oued Mouilah . . . . .	103
5.3.2.1	Indice de compacité de Gravelius . . . . .	104
5.3.2.2	Rectangle équivalent : . . . . .	104
5.3.2.3	Le relief . . . . .	105
5.3.3	Facteurs physiographiques du bassin versant . . . . .	108
5.3.3.1	Densité de drainage . . . . .	108
5.3.3.2	Rapport de confluence . . . . .	108
5.3.3.3	Rapport des longueurs . . . . .	108
5.3.3.4	Coefficient de torrencialité . . . . .	109
5.3.3.5	Temps de concentration . . . . .	109
5.3.4	La géologie de la région . . . . .	109
5.3.5	Caractéristiques climatiques de la région . . . . .	110
5.3.5.1	La pluviométrie . . . . .	110
5.3.5.2	La température de la région . . . . .	111
5.4	Conclusion . . . . .	112
<b>Chapitre 6: Fonctionnement du barrage Hammam Boughrara</b>		<b>115</b>
6.1	Introduction . . . . .	115
6.2	Présentation du barrage Hammam Boughrara . . . . .	115
6.2.1	La digue . . . . .	116
6.2.2	Évacuateurs de crue . . . . .	117
6.2.2.1	Évacuateur de crue principal . . . . .	117
6.2.2.2	Évacuateur de crue fusible . . . . .	117
6.2.3	La tour de prise . . . . .	118
6.3	Pollution du réservoir Hammam Boughrara . . . . .	118
6.3.1	Pollution d'origine domestique . . . . .	120
6.3.2	Pollution d'origine agricole et industrielle . . . . .	120
6.4	Présentation hydraulique du réservoir H. Boughrara . . . . .	121
6.4.1	Relation hauteur-surface-capacité du réservoir H.Boughrara . . . . .	121
6.4.2	Les apports du réservoir . . . . .	121
6.4.3	Les pertes d'eau par évaporation . . . . .	125
6.4.4	Les pertes d'eau par infiltration . . . . .	126
6.5	Usages de l'eau du réservoir Hammam Boughrara . . . . .	126
6.5.1	Objectifs d'exploitation du réservoir H.Boughrara . . . . .	126
6.5.2	Approvisionnement en eau potable . . . . .	128
6.5.3	Approvisionnement en eau d'irrigation . . . . .	128
6.6	Modélisation de l'usage de l'eau du réservoir . . . . .	129
6.6.1	Projection des besoins en eau potable . . . . .	130

6.6.1.1	Projection des besoins en eau potable domestique . . . . .	130
6.6.1.2	Projection des besoins en eau industriels . . . . .	131
6.6.2	Projection des besoins en eau agricoles . . . . .	131
6.6.2.1	Estimation des besoins en eau des cultures . . . . .	132
6.6.2.2	Estimation des besoins en eau d'irrigation . . . . .	135
6.7	Conclusion . . . . .	139
<b>Chapitre 7: Prévision des apports mensuels du barrage Hammam Boughrara par réseaux de neurones artificiels</b>		<b>141</b>
7.1	Introduction . . . . .	141
7.2	Prévision des apports mensuels . . . . .	142
7.2.1	La série chronologique des apports . . . . .	142
7.2.2	Critères d'évaluation des résultats . . . . .	145
7.2.2.1	L'erreur moyenne absolue "Mean Absolute Error" (MAE) . . . . .	145
7.2.2.2	La racine de l'erreur moyenne quadratique "Root Mean Square Error" (RMSE) . . . . .	145
7.2.2.3	Le coefficient de corrélation ( $R$ ) . . . . .	145
7.2.2.4	Le coefficient de détermination ( $R^2$ ) . . . . .	146
7.2.2.5	Le critère de Nash-Sutcliffe . . . . .	146
7.2.2.6	L'indice de Willmot $d$ "Index of agreement" . . . . .	147
7.2.3	Organisation des données d'entraînement des RNAs . . . . .	147
7.2.4	Résultats d'apprentissage et de validation des RNAs . . . . .	150
7.2.4.1	Réseaux de neurones mono-couche cachée . . . . .	150
7.2.4.2	Réseaux de neurones à deux couches cachées . . . . .	152
7.2.5	Prévision par réseaux de neurones artificiels . . . . .	154
7.3	Conclusion . . . . .	158
<b>Chapitre 8: Prévision des apports mensuels du barrage Hammam Boughrara par réseaux de neurones artificiels évolutionnaires</b>		<b>161</b>
8.1	Introduction . . . . .	161
8.2	Recherche des réseaux de neurones évolutionnaires . . . . .	162
8.2.1	Organisation des données d'entraînement . . . . .	162
8.2.2	Paramétrage des algorithmes d'optimisation . . . . .	163
8.3	Résultats de recherche des EANNs . . . . .	163
8.3.1	Comparaison graphique des résultats de la phase d'apprentissage . . . . .	167
8.3.2	Comparaison graphique des résultats de la phase de validation . . . . .	168
8.3.3	Performance des algorithmes d'optimisation . . . . .	168
8.4	Résultats de prévision des apports . . . . .	169
8.5	Conclusion . . . . .	172

<b>Chapitre 9: Optimisation de la gestion du réservoir Hammam Boughrara : Résultats et discussions</b>	<b>175</b>
9.1 Introduction . . . . .	175
9.2 Fonction objectif et entrées du modèle d'optimisation . . . . .	176
9.3 Algorithmes d'optimisation et paramétrage . . . . .	178
9.4 Critères de performance . . . . .	179
9.4.1 La valeur finale de la fonction objectif . . . . .	180
9.4.2 Volumes déversés . . . . .	180
9.4.3 Fiabilité volumétrique . . . . .	180
9.4.4 Fiabilité temporelle . . . . .	180
9.4.5 Résilience . . . . .	181
9.4.6 Vulnérabilité . . . . .	181
9.4.7 Le degré de couverture aux échelles taux de satisfaction - périodes satisfaites . . . . .	182
9.5 Optimisation de la gestion du réservoir H. Boghrara . . . . .	182
9.5.1 Scénario 1 : Apports mensuels très faibles . . . . .	182
9.5.1.1 Résultats et discussion . . . . .	183
9.5.1.2 Synthèse des principaux résultats . . . . .	188
9.5.2 Scénario 2 : Apports mensuels produisant des volumes proches à la moyenne annuelle . . . . .	189
9.5.2.1 Résultats et discussion . . . . .	190
9.5.2.2 Synthèse des principaux résultats . . . . .	194
9.5.3 Scénario 3 : Séquence d'années humides . . . . .	195
9.5.3.1 Résultats et discussion . . . . .	196
9.5.3.2 Synthèse des principaux résultats . . . . .	200
9.6 Convergence, stabilité et temps d'exécution . . . . .	201
9.7 Conclusion . . . . .	203
<b>Chapitre 10: Présentation de l'application RMOSS</b>	<b>207</b>
10.1 Introduction . . . . .	207
10.2 Outils de développement utilisés . . . . .	207
10.3 Présentation de l'application . . . . .	208
10.4 Architecture de l'application . . . . .	209
10.4.1 Couche d'interface d'utilisateur . . . . .	209
10.4.2 Couche moteur du calcul . . . . .	210
10.4.2.1 Bibliothèque des algorithmes évolutionnaires . . . . .	211
10.4.2.2 Bibliothèque des réseaux de neurones artificiels . . . . .	211
10.4.2.3 Moteur d'optimisation de la gestion opérationnelle des réservoirs . . . . .	212

10.4.3 Couche de sauvegarde et récupération des données . . . . .	212
10.5 Conclusion . . . . .	212
Conclusion générale	<b>215</b>
Bibliographie	<b>221</b>
Annexes	<b>235</b>
Annexe A . . . . .	235
Annexe B . . . . .	241
Annexe C . . . . .	242
Annexe D . . . . .	243
Annexe E . . . . .	246
Annexe F . . . . .	250
Annexe G . . . . .	258
Manuscrit 1 : Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Reservoir Operation management	<b>264</b>
Manuscrit 2 : Development of Reservoir Management Optimal Rules : Case of Hammam Boughrara Dam, Wilaya of Tlemcen, Algeria	<b>281</b>



# Liste des tableaux

Table 1.1	Augmentation exponentielle de complexité de la DP . . . . .	32
Table 2.1	Inconvénients majeurs des méthodes classiques principales . . . . .	49
Table 3.1	Fonctions d'activation utilisées dans cette étude . . . . .	73
Table 5.1	Paramètres de forme de bassin versant d'Oued Mouilah . . . . .	104
Table 5.2	Paramètres de forme et dimensions du rectangle équivalent . . . . .	105
Table 5.3	Indices de pente relatifs au bassin versant d'oued Mouilah . . . . .	107
Table 5.4	Caractéristiques morphologiques du bassin d'Oued Mouilah . . . . .	108
Table 5.5	Paramètres physiographiques du bassin d'Oued Mouilah . . . . .	109
Table 6.1	Caractéristique du barrage Hammam Boughrara (TRACTEBEL 2003) . . . . .	119
Table 6.2	Qualité de l'eau du réservoir Hammam Boughrara (2001-2010) (valeurs moyennes) (Hamlat et al. 2014) . . . . .	119
Table 6.3	Caractéristiques de la série chronologique des apports du réservoir Hammam Boughrara à l'échelle annuelle et mensuelle . . . . .	124
Table 6.4	Caractéristiques des apports moyens mensuels du barrage Hammam Boughrara (1972 – 2016) . . . . .	125
Table 6.5	Estimation de l'évolution de la population et ses besoins annuels en eau potable de la région de Maghnia . . . . .	130
Table 6.6	Consommation industrielle en eau potable de la région de Maghnia (année 2006) (Bounoua 2014) . . . . .	132
Table 6.7	Projection des besoins en eau domestiques et industriels de la région de Maghnia à différents horizons . . . . .	132
Table 6.8	Assolement actuel et proposé du périmètre irrigué de Maghnia . . . . .	133
Table 6.9	Rendements du réseau, d'uniformité et global d'irrigation du périmètre de Maghnia . . . . .	137
Table 6.10	Variation des volumes de besoins bruts en eau en fonction de la surface cultivée du périmètre irrigué de Maghnia . . . . .	137
Table 7.1	Caractéristiques statistiques de la série temporelle des apports du barrage Hammam Boughrara . . . . .	143

Table 7.2	Modèles de régression et organisation de la série chronologique des apports en entrées et sorties . . . . .	148
Table 7.3	Structures des RNAs utilisés pour la prévision des apports . . . . .	149
Table 7.4	Paramètres des algorithmes d'apprentissage et fonctions d'activation . . . . .	150
Table 8.1	Modèles de régression et organisation de la série chronologique des apports en entrées et sorties . . . . .	162
Table 8.2	Paramètres des algorithmes d'optimisation . . . . .	163
Table 8.3	Le temps moyen de recherche des EANNs proches-optimaux . . . . .	169
Table 9.1	Paramétrage des algorithmes d'optimisation RGA, GSA, HPSOGWO et IGWO . . . . .	179
Table 9.2	Indicateurs de performance relatifs au scénario 1 . . . . .	185
Table 9.3	Surfaces au-dessous des courbes $N_{ps} = f(\tau_s)$ relatives aux lâchers optimisés par les quatre algorithmes d'optimisation -Scénario 1- . . . . .	187
Table 9.4	Indicateurs de performance relatifs au scénario 2 . . . . .	192
Table 9.5	Surfaces au-dessous des courbes $N_{ps} = f(\tau_s)$ relatives aux lâchers optimisés par les quatre algorithmes d'optimisation -Scénario 2- . . . . .	194
Table 9.6	Indicateurs de performance relatifs au scénario 3 . . . . .	199
Table 9.7	Surfaces au-dessous des courbes $N_{ps} = f(\tau_s)$ relatives aux lâchers optimisés par les quatre algorithmes d'optimisation -Scénario 3- . . . . .	200
Table 1	Répartition hypsométrique des surfaces du bassin d'Oued Mouilah (Bouanani 2004) . . . . .	242
Table 2	Classification de l'IRD (ex, ORSTOM) de relief à partir de l'indice de pente globale . . . . .	242
Table 3	Classification de relief de l'IRD (ex, ORSTOM) en fonction de la dénivelée spécifique (Laborde 2000) . . . . .	242
Table 4	Fuites par infiltration du barrage Hammam Bouhrara ( $Mm^3$ ) (Bounoua 2014) . . . . .	243
Table 5	Coefficients culturaux ( $K_c$ ) des cultures (Bensalem and Bouhenna 2017) . . . . .	244
Table 6	Pluies de fréquence (80%), pluies utiles, données et résultats de calcul de $ET_0$ (Bensalem and Bouhenna 2017) . . . . .	244
Table 7	Besoins nets, bruts et volumes mensuels de l'eau du périmètre irrigué de Maghnia (Surface totale = 7000 ha) . . . . .	245
Table 8	Résultats d'entraînement et de validation des réseaux de neurones dotés d'une seule couche cachée . . . . .	246
Table 9	Résultats d'entraînement et de validation des réseaux de neurones dotés de deux (02) couches cachées . . . . .	248

Table 13	Résultats d'entraînement et de validation des réseaux de neurones artificiels (Optimisation par HPSOGWO, apprentissage par BP et LM) . . . . .	250
Table 10	Résultats de recherche des EANNs (Optimisation par GSA, Apprentissage par BP et LM) . . . . .	251
Table 11	Résultats de recherche des EANNs (Optimisation par HPSOGWO, apprentissage par BP et LM) . . . . .	252
Table 12	Résultats d'entraînement et de validation des réseaux de neurones artificiels (Optimisation par GSA, apprentissage par BP et LM) . . . . .	253
Table 14	Performance et stabilité des algorithmes d'optimisation RGA, GSA, HPSOGWO et IGWO (scénarios 1 et 2) . . . . .	258
Table 15	Performance et stabilité des algorithmes d'optimisation RGA, GSA, HPSOGWO et IGWO (scénario 3) . . . . .	259



# Table des figures

Figure 1	La démarche d'optimisation de la gestion opérationnelle (génération des politiques optimales) appliquée sur le réservoir Hammam Bouhrara . . . . .	5
Figure 1.1	Principales composantes gouvernant la prise de décision sur l'allocation de la ressource en eau (François 2013) . . . . .	14
Figure 1.2	Règles de distribution temporelle des lâchers. (a) réservoir Hammam Bouhrara (Dahmani et al. 2018, annexes), (b) réservoir Dharoi (Inde) (François 2013), (c et d) réservoir Epitacio Pessoa (Brésil) (Celeste and Billib 2009) . . .	18
Figure 1.3	Élaboration des courbes règles optimales de la gestion opérationnelle d'un réservoir par l'approche d'optimisation stochastique implicite "ISO". Adapté de (Celeste and Billib 2009) . . . . .	20
Figure 1.4	Élaboration des courbes règles optimales de la gestion opérationnelle d'un réservoir par l'approche de paramétrisation–simulation–optimisation "P-S-O". Adapté de (Celeste and Billib 2009) . . . . .	22
Figure 1.5	Classification des méthodes d'optimisation de la gestion des réservoirs (Ahmad et al. 2014) . . . . .	29
Figure 2.1	Minimum local et global d'une fonction . . . . .	41
Figure 2.2	Classification des méthodes d'optimisation. Adapté d'après (Talbi 2009)	43
Figure 2.3	Les deux critères conflictuels (la diversification et l'intensification) des algorithmes métaheuristiques. Adapté d'après (Talbi 2009) . . . . .	46
Figure 2.4	Organigramme explicatif du fonctionnement de l'algorithme génétique à codage réel (RGA) . . . . .	56
Figure 2.5	Principe du GSA. Chaque masse accélère vers la force résultat de l'agissement des autres masses (Rashedi et al. 2009) . . . . .	57
Figure 2.6	. . . . .	60
Figure 2.7	La hiérarchie sociale des groupes de Loups Gris (Mirjalili et al. 2014) .	62
Figure 2.8	Mise à jour des positions des éléments de recherche dans le GWO et IGWO. Adapté d'après (Mirjalili et al. 2014) . . . . .	62
Figure 2.9	Organigramme général de processus d'optimisation des algorithmes GWO, IGWO et HPSOGWO . . . . .	64
Figure 3.1	. . . . .	71

Figure 3.2	Schéma d'un neurone formel . . . . .	72
Figure 3.3	Schéma explicatif de structure d'un RNA multicouche . . . . .	74
Figure 3.4	Exemple d'un réseau de neurones artificiels statique "non-récurrent" . . .	76
Figure 3.5	Exemple d'un réseau de neurones artificiels dynamique "récurrent" . . .	77
Figure 3.6	RNA multicouche . . . . .	78
Figure 3.7	Schéma d'un neurone formel, élément de base d'un RNA multicouche .	81
Figure 3.8	RNA multicouche (01 d'entrées, 01 de sortie, 02 couches cachées) . . .	82
Figure 4.1	Couches de recherche évolutionnaire des paramètres des RNAs (adaptée d'après Almeida and Ludermir 2008a) . . . . .	92
Figure 4.2	Approche adoptée de recherche d'un RNA proche-optimal . . . . .	93
Figure 4.3	Organigramme de recherche d'un RNA proche-optimal selon l'approche adoptée . . . . .	94
Figure 4.4	Codage des solutions et intervalles de définition des variables selon l'al- gorithme d'apprentissage : (a) Back-propagation, (b) Levenberg-Marquardt . .	95
Figure 5.1	Découpage de l'Algérie du nord en quatre régions (Remini 2005) . . . .	100
Figure 5.2	Situation géographique du bassin versant de la Tafna (Ghenim 2008) . .	101
Figure 5.3	Les sous bassins de la Tafna (Bounoua 2014) . . . . .	101
Figure 5.4	Situation géographique du bassin versant d'Oued Mouilah (Dahmani and Yebdri 2020) . . . . .	103
Figure 5.5	La courbe hypsométrique du bassin versant d'Oued Mouilah . . . . .	106
Figure 5.6	Reliefs du bassin d'Oued Mouilah . . . . .	106
Figure 5.7	Carte de distribution des pentes du bassin versant d'Oued Mouilah (Bou- noua 2014) . . . . .	107
Figure 5.8	La carte géologique du bassin versant de l'Oued Mouilah (Ghenim 2008)	110
Figure 5.9	Carte des isohyètes du bassin versant de Tafna (Bounoua 2014) . . . . .	111
Figure 5.10	(a) la répartition des précipitations moyenne mensuelles (Station de Magh- nia), (b) distribution mensuelle des fréquences de précipitations . . . . .	112
Figure 5.11	Températures de la région de Maghnia (1997-2016) . . . . .	112
Figure 6.1	La situation géographique du barrage Hammam Boughrara (W. Tlemcen)	116
Figure 6.2	Profil en travers du barrage Hammam Boughrara (Smail 2014) . . . . .	117
Figure 6.3	Courbes hauteurs-surfaces-volumes du réservoir Hammam Boughrara (DRE (2012); ANBT (2012)) . . . . .	121
Figure 6.4	Le réseau hydrographique principal alimentant le réservoir Hammam Boughrara (Bounoua 2014) . . . . .	122
Figure 6.5	Les apports annuels du réservoir Hammam Boughrara (1972- 2016) (DRE 2012; ANBT 2012) . . . . .	123

Figure 6.6	Les apports mensuels du réservoir Hammam Boughrara (Septembre 1972 - Décembre 2016) (DRE 2012; ANBT 2012)	123
Figure 6.7	Variation mensuelle interannuelle des apports du barrage Hammam Boughrara (Septembre 1972 - Décembre 2016)	125
Figure 6.8	Évaporation du réservoir Hammam Boughrara (septembre 2000 - août 2011)(DRE 2012)	126
Figure 6.9	Les objectifs initiaux d'allocation de l'eau du réservoir Hammam Boughrara	127
Figure 6.10	Les volumes annuels desservis à la région de Maghnia et son périmètre d'irrigation à partir du réservoir Hammam Boughrara (2001-2012) (ANBT 2012; DRE 2012)	128
Figure 6.11	Situation géographique du périmètre d'irrigation de Maghnia	129
Figure 6.12	Répartition mensuelle de la consommation de l'eau potable de Maghnia (année 2012) (DRE 2012)	131
Figure 6.13	Évolution des besoins domestiques et industriels en eau de la région de Maghnia (2005-2030)	133
Figure 6.14	Évaluation de l'évapotranspiration réelle des cultures	134
Figure 6.15	Estimation des besoins nets mensuels en eau des cultures de périmètre irrigué de Maghnia ( $m^3/ha$ )	135
Figure 6.16	Volumes des besoins bruts mensuels de différentes cultures du périmètre irrigué de Maghnia (surface totale 7000 ha)	138
Figure 6.17	Volumes des besoins en eau mensuels domestiques (AEP de la région de Maghnia), industriels et d'irrigation du périmètre agricoles de Maghnia (Jan-2017 – Déc-2026)	138
Figure 7.1	Tracé de la série chronologique des apports mensuels du réservoir Hammam Boughrara (septembre 1972 au décembre 2016)	144
Figure 7.2	Corrélation entre les pluies moyennes mensuelles enregistrées à la station de Maghnia et la série chronologique des apports (janvier 1997 au décembre 2016)	144
Figure 7.3	Tracé de la fonction d'autocorrélation partielle simple de la série entière de débits (septembre 1972 au décembre 2016)	147
Figure 7.4	Comparaison des apports observés et ceux calculés par les meilleurs RNAs (Phase d'apprentissage)	155
Figure 7.5	Comparaison des apports observés et ceux calculés par les meilleurs RNAs (Phase de validation)	156
Figure 7.6	Prévision des apports par les meilleurs réseaux de neurones artificiels (les apports moyens mensuels interannuels sont calculés à la base de la série chronologique entière (septembre - 1972 au décembre -2016))	157

Figure 8.1	Évolution de la valeur de la fonction objectif (équation (4.1)) en fonction de nombre d'itérations (les meilleurs modèles) . . . . .	170
Figure 8.2	Résultats de prévision des apports du réservoir Hammam Boughrara par les réseaux de neurones évolutionnaires . . . . .	171
Figure 9.1	Scénarios d'apports 1 et 2 prévus par les réseaux de neurones artificiels évolutionnaires (EANNs) et scénario 3 prévu par un réseau de neurones artificiels conventionnel (ANN), ainsi la courbe modélisant la demande en eau mensuelle en aval du réservoir Hammam Boughrara . . . . .	177
Figure 9.2	Les taux d'évaporation mensuels utilisés dans le processus d'optimisation	178
Figure 9.3	Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 1- . . . . .	183
Figure 9.4	Confrontation des lâchers mensuels optimisés par les différents algorithmes à la demande en eau mensuelle ciblée - Scénario 1 - . . . . .	184
Figure 9.5	Variation de nombre de mois $N_{ps}$ satisfaits -au minimum- à différents taux de satisfaction de la demande $\tau_s$ (Scénario 1). Par exemple, 117 demandes mensuelles peuvent être satisfaites au minimum à 20% (algorithmes RGA, GSA et HPSOGWO). La satisfaction de la demande au minimum à 20%, touche 108 mois au maximum (algorithme IGWO) . . . . .	187
Figure 9.6	Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 2- . . . . .	190
Figure 9.7	Confrontation des lâchers mensuels optimisés par les différents algorithmes à la demande en eau mensuelle ciblée - Scénario 2 - . . . . .	192
Figure 9.8	Nombre de mois satisfaits en fonction de taux de satisfaction de la demande en eau mensuelle - Scénario 2- . . . . .	193
Figure 9.9	Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 3- . . . . .	197
Figure 9.10	Comparaison des lâchers mensuels optimisés par les différents algorithmes avec la demande en eau mensuelle ciblée - Scénario 3 - . . . . .	198
Figure 9.11	Nombre de mois satisfaits en fonction de taux de satisfaction de la demande en eau mensuelle - Scénario 3- . . . . .	199
Figure 10.1	L'application "RMOSS" dans l'action d'optimisation de la gestion du réservoir Hammam Boughrara par l'algorithme HPSOGWO . . . . .	209
Figure 10.2	Architecture de l'application "RMOSS" . . . . .	210
Figure 3	Les fonctions d'activation usuelles ; (a) : Linéaire, (b) : Sigmoides, (c) : Bipolaire Sigmoides, (d) : Gaussienne. . . . .	235
Figure 4	Organigramme explicatif de l'entraînement des RNA par l'algorithme génétique RGA . . . . .	237

Figure 5	Organigramme explicatif de l'entraînement des RNA par l'algorithme HPSOGWO . . . . .	240
Figure 6	Codage des solutions et intervalles de définition des variables selon l'algorithme d'apprentissage : (a) l'algorithme génétique à codage réel "RGA", (b) l'algorithme HPSOGWO . . . . .	241
Figure 7	Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase d'apprentissage, optimisation par l'algorithme GSA) . . . . .	254
Figure 8	Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase d'apprentissage, optimisation par HPSOGWO) . . . . .	255
Figure 9	Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase du test) . . . . .	256
Figure 10	Évolution de la valeur de la fonction objectif (équation (4.1)) en fonction de nombre d'itérations pour les différents modèles . . . . .	257
Figure 11	Variation de stockage, déversement et évaporation, scénario 1 . . . . .	260
Figure 12	Variation de stockage, déversement et évaporation, scénario 2 . . . . .	261
Figure 13	Variation de stockage, déversement et évaporation, scénario 3 . . . . .	262

# Liste des abréviations

$D_t$  : Demande en eau au temps  $t$ .

$Q_t$  : Apport au temps  $t$ .

$R_t$  : Lâcher à la période  $t$ .

**ANNs** "Artificial Neural Networks", Réseaux de neurones artificiels.

**AR** Autoregressive.

**ARIMA** Autoregressive integrated moving average.

**ARMA** Autoregressive–moving-average.

**BP** Backpropagation, "Rétropropagation du gradient de l'erreur".

**DDP** Deterministic Dynamic Programming.

**DP** Dynamic Programming.

**EANNs** Evolutionary Artificial Neural Networks.

**EOAs** Evolutionary Optimization Algorithms.

**ESO** Explicit Stochastic Optimization.

**GA** Genetic Algorithm.

**GAs** Genetic Algorithms.

**GSA** Gravitational Search Algorithm.

**HPSOGWO** Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer.

**IGWO** Improved Grey Wolf Optimizer.

**IPM** Interior-Point Method.

**ISO** Implicit Stochastic Optimization.

**LM** Levenberg-Marquardt.

**LP** Linear Programming.

**NFP** Network Flow Programming.

**NLP** Non-Linear Programming.

**P-DIP** Primal-Dual Interior Point.

**P-S-O** Parameterization – Simulation – Optimization.

**PSO** Particle Swarm Optimization.

**RG** Real-coded Genetic Algorithm.

**RNA** Réseau de neurones artificiels.

**RNAs** Réseaux de neurones artificiels.

**SDP** Stochastic Dynamic Programming.

**SVR** Support Vector Regression.

# Introduction générale

Aujourd'hui, partout dans le monde, la principale utilisation des ressources en eau est l'approvisionnement en eau potable de la population et l'irrigation des terres agricoles. Les objectifs des projets d'eau ne sont toutefois plus limités à ces deux utilisations, bien qu'elles soient encore majoritairement mondiales. Avec l'irrégularité spatiale et temporelle des précipitations, l'eau n'est pas disponible ou suffisante pour certaines périodes de temps. En conséquence, les réservoirs sont indispensables pour stocker le surplus de l'eau et le redistribuer lorsque le besoin s'en fait sentir.

La commission mondiale sur les barrages (World Commission on Dams 2000) a noté que de nombreux pays en développement ont tendance à construire de nouveaux barrages plutôt que d'adopter des stratégies efficaces de gestion des réservoirs et d'optimisation de l'allocation de l'eau face à l'augmentation de la demande en eau et au changement d'utilisation des terres dans les bassins versants. Actuellement, il est de grand intérêt pour les gestionnaires des barrages, les producteurs de l'énergie électrique, les bureaux d'études de conception et génie-conseil, entre autres, d'avoir des outils performants permettant la génération des stratégies optimales de gestion opérationnelle des barrages réservoirs dans des conditions de fonctionnement climatologiques, démographiques et économiques fréquemment incertaines.

L'élaboration des politiques et règles de la gestion du fonctionnement des barrages réservoirs peut être un problème délicat à cause de divers motifs : la variation et l'incertitude des apports liquides des réservoirs ; les lâchers d'eau ne dépendent pas de la demande en eau seulement, mais se rattachent aussi au niveau d'eau dans le réservoir, la période du lâcher (la saison), les caractéristiques des ouvrages de lâcher, le mode de régularisation (annuel ou inter-annuel) du réservoir, ... etc. De plus, les réservoirs présentent des objectifs souvent conflictuels dont la satisfaction de l'un peut être au détriment d'un autre, tels que l'alimentation en eau potable, l'irrigation, la protection contre inondations, les activités récréatives, ... etc. Sans oublier la stochasticité et l'incertitude lors de la prévision et l'estimation des apports liquides et les pertes par évaporation et infiltration d'une part, et d'autre part, l'incertitude lors de la prévision et la quantification des volumes consommés et leurs impacts socio-économiques sur les divers consommateurs.

L'objectif général de cette thèse est de traiter la problématique de l'optimisation de la gestion opérationnelle d'un réservoir à usage multiple. Le cas d'étude choisi est le barrage Hammam Boughrara situé à la wilaya de Tlemcen au nord-ouest algérien. Cet ouvrage est construit pour l'allocation en eau de la région de Maghnia (W. Tlemcen), y compris l'alimentation en eau potable, industrielle et d'irrigation.

D'une manière générale, l'optimisation de la gestion du fonctionnement d'un réservoir consiste à dériver deux types de règles de gestion, (1) la règle de répartition de la ressources en eau entre usagers, et (2) la règle de distribution de cette ressource au fil du temps. Cette dernière règle peut être exprimée par la question suivante posée par le gestionnaire de tel ouvrage à chaque moment de décision :

*Est-ce j'ai l'intérêt à lâcher l'eau ou la stocker à un moment ultérieur ?*

La règle de distribution temporelle est celle qui a attiré plus l'intérêt des chercheurs dans ce domaine à cause de la stochasticité, l'incertitude et la complexité caractérisant les phénomènes naturels et socio-économiques qui l'influent. En effet, dans ce travail on s'intéresse uniquement à cette règle de répartition temporelle de la ressource en eau délivrée par le réservoir.

Trois approches "d'optimisation stochastique" sont développées dans la littérature à fin de considérer la nature stochastique des différents phénomènes liés à la gestion du fonctionnement des réservoirs, notamment ceux relatifs à l'hydrologie et au comportement des bassins versants au cours du temps (p. ex., les apports). La première est dite "approche d'optimisation stochastique implicite" ("Implicit Stochastic Optimization, ISO" en Anglais), la deuxième "approche d'optimisation stochastique explicite" ("Explicit Stochastic Optimization, ESO"), et la troisième "approche par paramétrage-simulation-optimisation" ("Parameterization – Simulation – Optimization, P-S-O") (Chapitre 1). La première approche (ISO) est celle retenue dans cette étude à cause de sa simplicité. Elle consiste à provenir les règles de la gestion optimale d'un réservoirs par trois étapes (en considérant les apports comme variable stochastique) :

1. Générer  $M$  séquences synthétiques des apports (scénarios d'apports) pour une durée de  $N$  mois. La multiplication des scénarios permet d'exprimer implicitement la stochasticité de la variable apport ;
2. Calculer les lâchers optimaux (décisions) pour chaque séquence (scénario) en utilisant une méthode d'optimisation appropriée. Cette étape conduit à la génération de  $M$  politiques de gestion ;
3. Utiliser l'ensemble des scénarios et lâchers optimisés ( $M \times N$ ) pour développer les règles optimales de gestion. Des techniques de régression peuvent être utilisées dans cette étape, telles que la régression multiple, les réseaux de neurones artificiels, la régression à vecteurs de support ("Support Vector Regression,SVR"), ... etc.

L'approche ISO n'est pas appliquée intégralement dans cette étude, seulement les deux premières étapes qui sont abordées. La qualité des résultats produits à la fin du processus d'élaboration des règles optimales de gestion d'un réservoir, est tributaire aux sorties de chaque

étape. En effet, une focalisation est opérée sur les réseaux de neurones artificiels tant que des modèles de prévision des apports (pour concrétiser la 1<sup>ère</sup> étape) et sur quatre algorithmes métaheuristiques évolutionnaires tant que des méthodes d'optimisation, à savoir :

1. L'algorithme génétique à codage réel ("Real-coded Genetic Algorithm, RGA");
2. L'algorithme de recherche gravitationnelle ("Gravitational Search Algorithm, GSA");
3. L'algorithme hybride de l'optimisation par l'essaim de particules et l'optimiseur de Loup Gris ("Hybrid algorithm of Particle Swarm Optimization and Grey Wolf Optimizer, HPSOGWO");
4. L'algorithme amélioré de l'optimiseur de Loup Gris ("Improved Grey Wolf Optimizer, IGWO").

Ces méthodes sont utilisées dans ce travail pour l'entraînement des réseaux de neurones artificiels, le paramétrage de ces derniers (de point de vue structure) et pour générer les politiques de gestion optimales (la 2<sup>ème</sup> étape de l'approche ISO) du réservoir Hammam Boughrara.

## **Problématique et organisation de la thèse**

L'élaboration des modèles d'optimisation de la gestion opérationnelle des réservoirs marque le besoin des méthodes de prévision et d'optimisation performantes. La performance s'exprime ici par la qualité de solutions en premier lieu, et la vitesse d'exécution en deuxième lieu. Dans le processus général de l'approche stochastique implicite (ISO), la qualité de solutions de chaque phase (prévision, optimisation et régression) détermine la qualité globale des règles de gestion obtenues à la fin.

Dans la première phase du processus, un modèle efficient de prévision des apports peut générer des séquences plus au moins étroites dans les limites de réponse du bassin versant. Ces séquences prévues restent mieux que des autres aléatoires "aveugles", car elles permettent l'achèvement aux règles de gestion plus affinées du réservoir. En effet, la recherche des modèles de prévision efficaces s'avère utile.

Dans ce travail, nous avons concentré sur les réseaux de neurones artificiels (RNAs) tant que des modèles de prévision largement utilisés. En effet, trois (03) essais sont effectués sur les RNAs multicouches à propagation avant et d'apprentissage supervisé, dans le but d'améliorer leurs résultats de prévision des apports mensuels du réservoir Hammam Boughrara. Notons que la série chronologique mensuelle des apports du septembre 1972 au décembre 2016 est utilisée. Les trois essais se résument en :

1. Préviation des apports mensuels par des réseaux de neurones artificiels paramétrés structurellement d'une façon manuelle et entraînés par deux méthodes dites "classiques" amplement adoptées (la rétropropagation du gradient de l'erreur "BP" et Levenberg-Marquardt "LM"). Les paramètres de structure d'un RNA sont tels que le nombre de couches cachées, nombre de neurones dans chaque couche, la fonction d'activation, ... etc ;
2. Emploi des algorithmes métaheuristiques évolutionnaires RGA et HPSOGWO pour l'entraînement (ou "apprentissage") des RNAs manuellement paramétrés. À la limite de notre connaissance, l'algorithme HPSOGWO n'a pas été utilisé auparavant dans la littérature pour l'entraînement des RNAs ;
3. Couplage des algorithmes métaheuristiques évolutionnaires GSA et HPSOGWO avec ceux classiques d'apprentissage (BP et LM) pour produire des réseaux de neurones artificiels dits "évolutionnaires" (Evolutionary Artificial neural networks, "EANNs"). Les métaheuristiques GSA et HPSOGWO sont utilisés pour choisir "automatiquement" des "meilleurs" paramètres de structure, tandis que les méthodes classiques sont réservées à l'entraînement des EANNs. Des tentatives de couplage de ce genre ont été réalisées dans la littérature, cependant celle proposée dans cette étude est plus simple et emploie des algorithmes différents.

La deuxième phase du processus de génération des règles de gestion optimales d'un réservoir à la base de l'approche ISO, implique la génération de nombreuses "politiques" de gestion référées aux différentes séquences "scénarios" d'apports. Cette phase consiste le cœur de ce travail. De fait de multiples appels de l'algorithme d'optimisation utilisé (au moins autant de séquences d'apports employées), la vitesse d'exécution de l'algorithme influe considérablement sur le temps global du processus. L'estime de la vitesse d'exécution ne doit pas être au détriment de la qualité des décisions préconisées (solutions). Ces dernières peuvent nuire l'efficacité des règles de gestion résultantes si sont qualitativement dégradées.

Plusieurs méthodes dites "classiques" ou "exactes" sont utilisées pour traiter le problème d'optimisation de l'exploitation des réservoirs et systèmes de réservoirs, notamment la programmation linéaire, non linéaire, la programmation dynamique déterministe, stochastique (utilisée dans l'approche ESO), ... etc. Malgré leurs capacités à atteindre les solutions optimales, elles souffrent de plusieurs inconvénients limitant leurs utilisations particulièrement dans les problèmes de tailles importantes (nombre important de variables inconnues à déterminer), sans oublier le temps et les ressources du calcul (traitement "processing" et mémoire des machines) excessifs nécessaires pour atteindre tel espèce de solutions.

Avec les points faibles constatés des anciennes méthodes, la recherche des nouvelles était indispensable. Des méthodes d'optimisation sont développées récemment notamment ces

dernières 30 années. Elles sont de l'intelligence artificielle, et sont connues sous l'appellation "Algorithmes Métaheuristiques". Ces derniers sont du calcul approché, et ont montré une efficacité (capacité de trouver les solutions optimales) ainsi une rapidité de traitement dans divers domaines. Cependant, aucune preuve ne démontre la généralisation de la performance et la supériorité d'une méthode métaheuristique dans tous les domaines d'applications. Par autre manière, si un algorithme métaheuristique donne de bonnes solutions d'optimisation des circuits électroniques par exemple, ceci ne prouve pas qu'il est bon pour l'optimisation des chaînes de distribution ou de la gestion des stocks. Cela implique l'étude de ces algorithmes dans chaque domaine d'application. Et vu les avantages qui présentent, ces méthodes méritent d'être explorées et exploitées dans l'optimisation de la gestion des réservoirs et les ressources en eau en général.

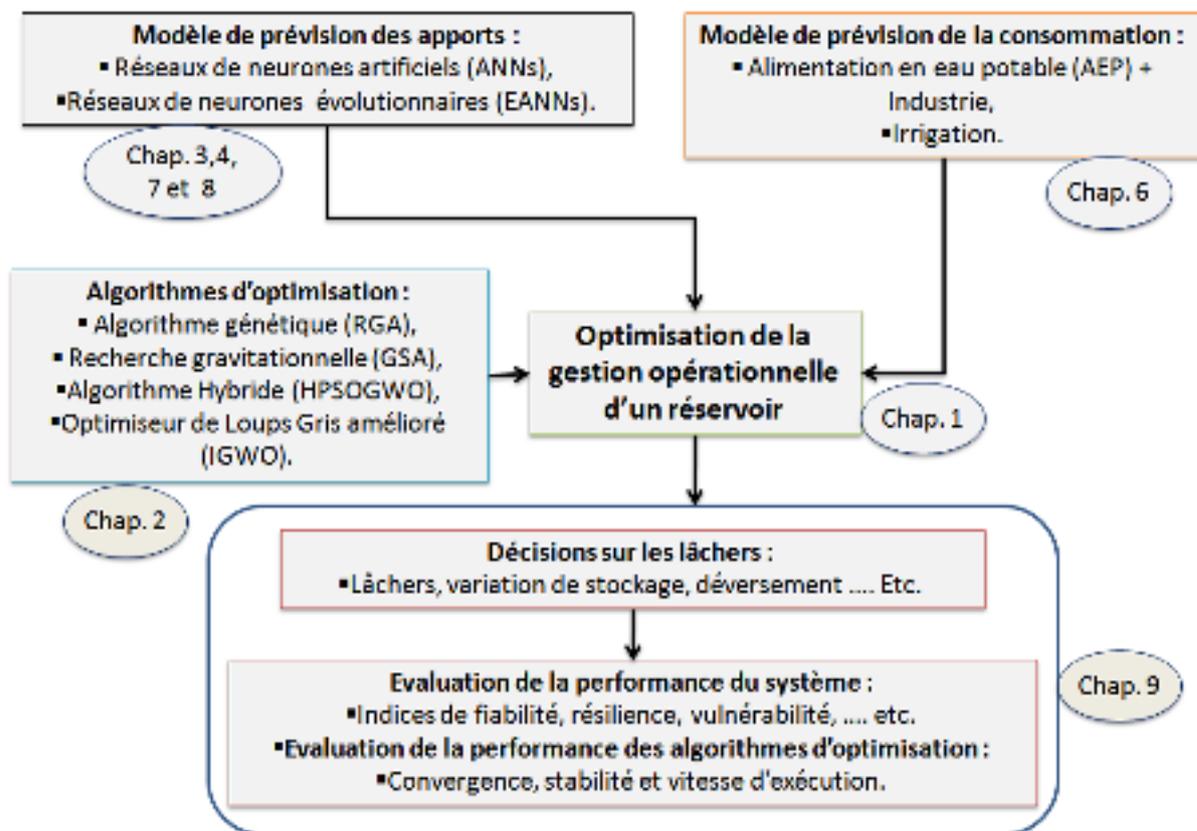


FIGURE 1 – La démarche d'optimisation de la gestion opérationnelle (génération des politiques optimales) appliquée sur le réservoir Hammam Boughrara

Dans ce contexte, les métaheuristiques RGA, GSA, HPSOGWO et IGWO sont évaluées dans la matière de déduction des politiques de gestion optimales à travers une étude comparative axée sur trois scénarios d'apports prévus du réservoir Hammam Boughrara. La motivation derrière le choix de ces algorithmes est bien exposée dans le chapitre 2.

La figure 1 montre le schéma de la démarche globale d'optimisation appliquée et

l'organisation générale de la thèse. Au cœur de cette démarche, un modèle d'optimisation sous contraintes avec une fonction objectif non-linéaire est adopté. La consommation de l'eau (AEP, industrie et irrigation) en aval du réservoir est présumée présente une variation mensuelle avec une tendance d'évolution au cours du temps jusqu'à l'atteint du volume régularisé ciblé par le gestionnaire de l'ouvrage.

L'évaluation des résultats obtenus touche deux (02) volets, à savoir :

1. Jugement de la qualité des politiques de gestion produites par les quatre algorithmes d'optimisation via des indices et critères tels que la fiabilité, résilience, vulnérabilité, ... etc ;
2. Métrique de la convergence, stabilité et vitesse du calcul des quatre algorithmes d'optimisation dans la matière du traitement de tels problème.

Le modèle d'optimisation et l'ensemble d'algorithmes métaheuristiques sont implémentés à zéro dans une application informatique via les langages de programmation *VisualBasic.NET* et *C#.NET* (C Sharp dotnet). L'application développée est nommée "Reservoir Management Optimization and Simulation Software, RMOSS". Son architecture est décrite dans le dernier chapitre du présent manuscrit.

Le travail est clôturé par une conclusion générale et une présentation de quelques perspectives paraissant mériter d'être explorées par la suite dans des travaux futurs.

## **Première partie**

# **L'optimisation de le gestion d'un barrage réservoir, les métaheuristiques et les réseaux de neurones artificiels**



# Chapitre 1

## Optimisation de la gestion des réservoirs

### 1.1 Introduction

L'augmentation de la population, l'amélioration du niveau de vie, la diminution en quantité et la dégradation en qualité de la ressource en eau compliquent d'un jour à l'autre la problématique de gestion de cette ressource vitale. Dans le monde, plus de la moitié de l'eau potable consommée est extraite des rivières ou des retenues construites le long de leurs berges.

Les systèmes de ressources en eau se caractérisent par de multiples composants interdépendants entre eux. Ces derniers produisent de divers impacts économiques, sociaux, environnementaux et écologiques qui remettent la gestion de tels systèmes une opération complexe, fastidieuse et sujette à nombreuses incertitudes.

Les planificateurs et les gestionnaires travaillent pour améliorer la performance de ces systèmes (augmentation de la fiabilité et résilience d'une part, et réduction de la vulnérabilité d'autre part) via des stratégies efficaces. Une stratégie de gestion basée sur une prévision de la ressource et des usages à satisfaire sera a priori plus performante qu'une stratégie "aveugle". En effet, ils doivent évaluer selon la phase abordée, les diverses conceptions, plans d'aménagement et politiques de gestion, et comparer leurs performances prévues avec les objectifs et les buts désirés.

Ce chapitre constitue une introduction brève à l'optimisation de la gestion opérationnelle des réservoirs. Dans ce contexte, les principaux objectifs de réalisation d'un barrage-réservoir, les règles de leur gestion, les approches et le modèle général d'optimisation de leur fonctionnement sont abordés. Aussi, le modèle d'optimisation appliqué dans ce qui suit du travail est décrit. Finalement, le chapitre est clôturé par un passage en revue sur les principales méthodes d'optimisation appliquées dans ce domaine et par une conclusion.

## **1.2 Nécessité et rôles des barrages – réservoirs**

Un système de réservoir unique ou multiples, est un ensemble formé de divers composants physiques, y compris réservoirs, canaux, tunnels, pipelines, stations de pompage, centrales hydroélectriques, zones d'irrigation et systèmes urbains d'approvisionnement en eau, fonctionne pour fournir de l'eau pour les besoins municipaux, industriels et d'irrigation, la production hydroélectrique, le contrôle des inondations, les loisirs, la navigation ou les besoins écologiques (Rani and Moreira 2010).

L'homme a constaté depuis les anciennes civilisations la nécessité de ces ouvrages pour sa survie et son développement. À titre indicatif, le premier ouvrage sert à stopper et à dériver l'eau remonte à 2900 av. J.C en Egypt. À cette époque, le Pharaon "MENES" fondateur de la première dynastie des Pharaons, a dérivé le Nil à "Koseish" pour construire sa capitale "Memphis". Ensuite, sous la gouvernance de la 3<sup>ème</sup> et la 4<sup>ème</sup> dynasties le barrage dit "SADD EL KAFARA" a été construit (environ 2650-2645 av.J.C) sur l'oued "Garawi" situé à 10km au sud-est de la ville de "Helwan" et à 30km au sud du Caire (Comité Français des Barrages et Réservoirs 2013).

La construction des barrages-réservoirs à nos jours est un investissement très coûteux. Tels ouvrages nécessitent la mobilisation directe des moyens humains et matériels excessifs, sans oublier le coût indirect représenté par les multitudes impactées sur le territoire. Ces ouvrages provoquent des déformations et des changements sur plusieurs dimensions : sociale, économique, environnemental, écologique, climatologiques, ... etc. Les coûts ne peuvent être justifiés dans la plupart des cas, sauf si l'ouvrage est de multiples objectifs.

Les barrages-réservoirs sont construits durant l'histoire de l'humanité pour deux principaux objectifs :

1. La réponse et la satisfaction des besoins en eau distribués spatialement et temporellement de la société ;
2. La régulation et le contrôle des débits des cours d'eau.

Comme premier objectif, l'eau constitue un élément vital de la vie, indispensable à l'alimentation en eau de qualité potable pour la population et de qualité requise pour les autres êtres vivants. La réalité hydrologique est que là où il n'y a pas d'eau, il n'y a pas de vie humaine, animale ou végétale (Jorgensen et al. 2005). L'homme utilise quotidiennement des quantités importantes de l'eau pour boire, l'hygiène, production des aliments et des biens. La consommation mondiale de l'eau durant le siècle passé (1900-2000) a passé de 600  $Km^3$  au début du siècle à 5500  $Km^3$  environ à sa fin (Atchia and Tropp 1995). Depuis les trois derniers siècles, la quantité

d'eau extraite des sources d'eau douce a été multipliée par 35, tandis que la population mondiale par 8 (CIGB 2020). Encore, pendant le 20<sup>ème</sup> siècle, la population mondiale a été triplée (multipliée par 3), tandis que la consommation mondiale a été multipliée par 6 (Szollosi-Nagy 2000 in Kelman and Kelman 2002). D'après (Atchia and Tropp 1995), à l'échelle de la planète, la quantité consommée se partage approximativement sur : 69% pour l'agriculture (principalement l'irrigation), 23% pour l'industrie et 8% pour l'usage domestique. L'avantage des barrages réservoirs réside dans le fait qu'ils peuvent assurer des grandes quantités d'eau de bonne qualité en temps continu (Jorgensen et al. 2005). La qualité d'eau issue des réservoirs peut être améliorée via un traitement relativement simple tolérant sa consommation sans dangers par les êtres humains et les autres êtres vivants.

Les activités industrielles constituent un consommateur important de l'eau. Certaines industries, comme les métallurgies par exemple, consomment des volumes énormes de l'eau, nécessitant la construction des barrages-réservoirs en proximité ou la réalisation des grands transferts pour assurer la continuité de la production. Malgré que les nappes souterraines représentent des sources d'eau préférées qualitativement, mais des restrictions comme la limitation des quantités soutirées et les taux de régénération moyens et faibles (voir parfois très faibles ou négligeables) et la sensibilité aux précipitations annuelles empêchent leurs exploitation à grandes échelles. Ces dernières contraintes donnent l'avantage aux eaux superficielles.

Certains pays ont décidé de mobiliser et exploiter les eaux superficielles en premier lieu en préservant les autres ressources plus vulnérables. En Algérie par exemple, les eaux de barrages sont pompées via transferts sur des longues distances (entre quelques kilomètres à quelques centaines de kilomètres) pour alimenter, notamment, les centres urbains (tels que le transfert de MAO (Mostaganem-Arzew-Oran), Béni-Haroune, Koudiat Acerdoune, Tilesdit, ... etc).

Le second objectif de construction des réservoirs permet en premier lieu la protection de la vie et les biens contre les effets destructeurs soient des inondations ou de sécheresse. En effet, les réservoirs sont des ouvrages préférés et très utiles pour le contrôle des débits des cours d'eau (Votruba and Broža 1989). Aussi, ils permettent -entre autres-, la régulation fluviale pour navigation, la production d'hydroélectricité, les loisirs et sports, l'aquaculture et la pêche, ainsi le contrôle de la qualité de l'eau. La majorité des réservoirs construits actuellement sont multi-objectifs. Même si un réservoir est construit pour un objectif unique, il sert à la protection contre les crues par conséquence (Votruba and Broža 1989).

À travers la régulation des cours d'eau, l'homme a construit les réservoirs pour générer l'énergie électrique. À présent, il n'existe pas une technique permettant de stocker des grandes quantités d'énergie électrique directement. Le seul moyen disponible actuellement, est de stocker soit l'eau ou les fluides fossiles (gaz, gasoil, ...) pour produire cette énergie. Selon

la commission internationale des grands barrages (CIGB 2020) les usines hydro-électriques à l'échelle mondiale disposent une capacité de  $675 \times 10^3$  mégawatt de production, et assurent 24% des besoins mondiaux de l'énergie électrique. Encore, quelques pays comme la Norvège, le Congo (ancien Zaïre) et le Brésil recourent aux usines hydro-électriques pour produire la quasi-totalité de leurs besoins en électricité avec un taux de 99% pour les deux premiers pays, et 91% pour le dernier. Aussi, plus de 90% de l'électricité renouvelable produite dans le monde provient des barrages, ce qui constitue la plus grande source d'énergie renouvelable au monde. De plus, ces usines génèrent l'électricité d'une façon économique. Elles exploitent l'énergie produite pendant la nuit lorsque la demande est basse pour pomper l'eau vers le réservoir. Cette eau est réutilisée pendant la journée ou lorsque la demande en énergie s'accroît. Un autre avantage s'aperçoit dans la réponse rapide des usines hydroélectriques face aux pics de demandes d'énergie électrique par rapport aux autres techniques telles que les centrales nucléaires (François 2013 ; CIGB 2020).

### **1.3 Gestion opérationnelle des barrages-réservoirs**

La réalisation et la gestion des barrages-réservoirs à objectifs multiples sont des opérations complexes. Elles nécessitent la considération d'un nombre important de facteurs et contraintes : techniques, économiques, sociales, environnementales et même politiques. La complexité s'amplifie dans les régions dotées de plusieurs ouvrages, et s'amplifie encore si les réservoirs sont connectés entre eux formant des systèmes. De plus, de nos jours, aucun ouvrage ne peut être réalisé sans de grands retards résultant d'innombrables objections, demandes et suggestions de changements et de modifications, notamment en ce qui concerne les effets environnementaux potentiels (Nagy et al. 2013).

La gestion des réserves hydrauliques superficielles artificielles consiste à dériver et restituer de l'eau à différentes périodes de l'année, de façon à satisfaire « *au mieux* » différents usages, domestiques (alimentation en eau potable et assainissement), industriel, irrigation, hydroélectricité, refroidissement des centrales thermiques (à flamme ou nucléaires), navigation, écrêtement des crues, loisirs nautiques, sauvegarde des milieux aquatiques, ... etc., (Roche et al. 2012). À cause de stochasticité de la pluviométrie qui représente la principale source d'alimentation des réservoirs, la problématique de gestion se complexifie encore surtout avec les fortes variations mensuelles, saisonnières et annuelles des apports.

L'opération d'un réservoir se résume à la réponse de deux questions principales d'une manière efficace, (1) quelle quantité à lâcher et (2) quand est ce qu'il faut la lâcher. En effet, le challenge est de déterminer la quantité optimale à allouer aux consommateurs à n'importe quel moment futur et n'importe quel état de stockage pouvant être produite dans l'avenir. Selon

(Celeste and Billib 2009), les décisions (commandes) prises sur les lâchers (et éventuellement sur le stockage) peuvent être déduites soit à partir des courbes règles ("Rule curves") issues des modèles de simulation ou à partir des modèles d'optimisation axés sur différentes techniques d'optimisation développées dans la recherche opérationnelle. Néanmoins, il existe toujours un déphasage entre le développement théorique des modèles et méthode dans la matière d'un côté, et l'application pratique sur terrain de l'autre côté (Wurbs 1993; Labadie 2004; Celeste and Billib 2009). Ajoutant à ça, les modèles d'optimisation ne sont pas facilement adoptés par les gestionnaires (opérateurs) des réservoirs à cause de leur complexité mathématique d'une part et l'inclusion explicite des variables stochastiques telles que les apports liquides (Celeste and Billib 2009).

Selon (François 2013), *"pour le gestionnaire d'une retenue hydraulique, la gestion résulte en une séquence de décisions portant sur les volumes à affecter à un moment donné aux différents utilisateurs de la ressource, que ce soit sous forme de prélèvements pour l'irrigation ou l'approvisionnement en eau potable des zones riveraines, ou sous forme de lâchers d'eau, éventuellement valorisés par une production d'hydroélectricité, pour la satisfaction de différentes demandes en aval. Les décisions prises quotidiennement par le gestionnaire sont guidées par une règle de gestion qui pourrait se décomposer en (1) une règle de répartition entre usages et (2) une stratégie de gestion (distribution temporelle de l'eau)."*

La décision journalière sur le volume d'eau à stocker ou à relâcher par un gestionnaire d'un système de ressources en eau dépend de trois (03) composants à savoir (François 2013) :

1. Le système physique, formé par le ou les bassins versants et leurs cours d'eau, ainsi les ouvrages et aménagements construits au sein de cet espace (barrages, retenues, ouvrages de dérivation, ... etc). Ces éléments sont sujets à des contraintes physiques telles que la capacité minimale et maximale de stockage des réservoirs, débit minimal et maximal de lâchers, distribution spatiale des ouvrages, distance entre ouvrages et points de consommation, capacité de production des centraux hydroélectriques, ... etc.
2. Les objectifs de la gestion du système physique définis par les usages souhaités de la ressource en eau.
3. La stratégie de gestion adoptée pour répondre au mieux les objectifs voulus, pour lesquels les différents aménagements ont été construits.

L'ensemble de ces composants constitue un *système de gestion de la ressource en eau* (figure 1.1) (François 2013).

L'élaboration d'une politique de gestion de ces ouvrages obéit à plusieurs incertitudes qui peuvent être classées en (Pabiot 1999) :

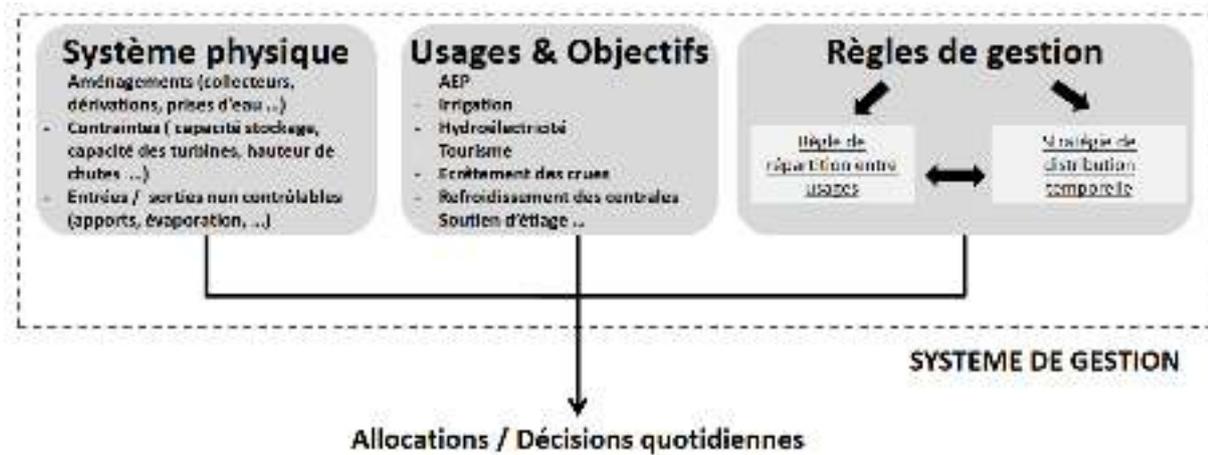


FIGURE 1.1 – Principales composantes gouvernant la prise de décision sur l'allocation de la ressource en eau (François 2013)

- Incertitudes stratégiques : incluant l'évaluation de l'opportunité de réalisation d'un aménagement et la prise de décision sur l'investissement, ainsi la définition d'une vision claire et permanente des objectifs de gestion ;
- Incertitudes physiques : dues à la variabilité spatio-temporelle des phénomènes hydrologiques qui s'intensifient dans les régions semi-arides ;
- Incertitudes technologiques : liées à la méconnaissance des processus physiques (prévision des précipitations, réponse de bassin versant via modèles "pluie-débit", changement climatique, ...) et économiques (revenus agricoles, industriels, touristiques, ...).

Les opérations de gestion touchent généralement les aspects suivants :

- L'approvisionnement en eau pour répondre à la demande qui se caractérise par une variabilité spatiale et temporelle,
- La production d'hydroélectricité,
- Le contrôle des inondations,
- Les loisirs et la navigation,
- La pêche et le contrôle de la qualité de l'eau.

### 1.3.1 Règles de gestion de la ressource en eau

Face aux diverses contraintes et objectifs, le gestionnaire se trouve dans la position de décideur sur la répartition de cette ressource sur le plan temporel (au fil du temps) ou sur le plan spatial (entre usagers). C'est pourquoi qu'il a besoin de règles claires afin de prononcer une décision (ou commande) sur l'allocation des volumes de l'eau dans l'espérance de satisfaire le

mieux possible les besoins des consommateurs au cours du temps.

Dans une unité de gestion de l'eau, qui peut être un bassin versant ou une aquifère, les règles de gestion peuvent être définies par le gouvernement ou adoptée suite à un dialogue au sein de la communauté des bénéficiaires (Richter and Evrard 2017). Cependant, l'allocation par les gouvernements, habituellement appelée allocation publique, ne porte pas sur l'efficacité économique généralement, mais elle est nécessaire en raison de plusieurs caractéristiques qui distinguent l'eau des autres ressources rares (Ariel et al. 1997). Cela permet d'éviter de subir cette matière au monopole ou aux spéculations.

Les règles de gestion de la ressource en eau peuvent être catégorisées en : i) règles de répartition entre usagers et ii) stratégies de distribution temporelle (François 2013).

### **1.3.1.1 Règles de répartition entre usages**

Au cours des années, les règles de gestion d'un nombre important de réservoirs ont été élaborées empiriquement à la base des considérations politiques et économiques pour équilibrer l'allocation de l'eau entre les usagers (François 2013).

La répartition de cette ressource entre usagers à un moment donné, doit tenir en compte plusieurs aspects, sociaux, économiques, environnementales et même politiques. Les conflits entre les usages de la ressource en eau sont parfois importants, notamment pour les réservoirs multi-objectifs. Les demandes en eau, domestique, d'irrigation, de récréation et de tourisme (exigeant un niveau cible des retenues par exemple) tendent à la hausse simultanément pendant la période estivale. Ces conflits s'aggravent encore dans les régions semi-arides ou pendant les périodes de sécheresse à cause de la limite ou l'insuffisance de l'eau.

Idéalement, la répartition de l'eau devrait être économiquement efficace et socialement équitable. Cependant, ces deux obligations sont souvent en conflit. Une allocation économiquement efficace vise à distribuer l'eau en maximisant les revenus économiques en amont sans différenciation entre les bénéficiaires. En revanche, l'allocation avec équité sociale vise à distribuer l'eau dans le but de protéger les intérêts des utilisateurs avec une moindre considération aux avantages économiques supplémentaires (Kelman and Kelman 2002). De plus, selon (Ariel et al. 1997), dans certains cas, l'allocation de l'eau par les gouvernements vise à renforcer les réformes agricoles et la bonification des terres pour atteindre des objectifs supplémentaires tels que la sécurité alimentaire, la réduction des taux de chômage, la redistribution des revenus et l'établissement des régions éloignées éléments de la sécurité sociale.

La conférence internationale sur l'eau et l'environnement déroulé à Dublin en 1992 a ressorti quatre (04) principes connus sous "Les principes de Dublin de la gestion intégrée de la

ressource en eau et de droit à l'eau" à savoir (Gonzalez-Villarreal and Solanes 1999) :

1. L'eau douce est une ressource limitée et vulnérable, essentielle pour soutenir la vie, le développement et l'environnement ;
2. Le développement et la gestion de l'eau devraient être fondés sur une approche participative, impliquant les utilisateurs, les planificateurs et les décideurs à tous les niveaux ;
3. Les femmes jouent un rôle central dans la fourniture, la gestion et la sauvegarde de l'eau ;
4. L'eau a une valeur économique dans toutes ses utilisations concurrentes, et devrait être reconnue comme un bien économique.

La considération de cette matière comme un bien économique (défini par une valeur ou un revenu monétaire) facilite la prise de décision sur l'allocation, car cette dernière peut être faite sur la maximisation des revenus ou la minimisation des déficits. Malheureusement, ce n'est pas le cas dans la majorité des systèmes de gestion.

(Ariel et al. 1997) considèrent deux (02) principes fondamentaux de répartition de l'eau entre les usagers à savoir :

**§1 L'efficacité économique :** La vision ici est purement économique. L'eau est considérée comme un capital dans les projets d'investissement, utilisée comme entrée dans différents secteurs économiques pour aboutir à un bénéfice. Une allocation économiquement efficace est atteinte si les revenus sont égaux d'un secteur à l'autre en utilisant la même unité (volume) de la ressource en eau. Dans le cas contraire, la société bénéficie plus d'intérêt en allouant plus d'unités (volume d'eau en plus) aux secteurs les plus rentables.

**§2 L'équité :** La répartition de la ressource en eau peut être effectuée sur l'équité aussi. Les objectifs de l'équité sont particulièrement préoccupés par l'équité de l'allocation entre les groupes économiquement différents et peuvent ou non être compatibles avec les objectifs d'efficacité économique. Sur le volet domestique, l'équité signifie le droit à l'eau et aux services de l'eau pour les foyers quelles que soient leurs capacités d'achat. Pour atteindre cet objectif, les gouvernements peuvent accorder des subventions, des services gratuits, ou adopter une structure de prix différentielle fondée sur le pouvoir d'achat.

Plusieurs critères sont utilisés pour comparer les formes d'allocation de l'eau. Kelman and Kelman 2002 ont présenté un modèle d'allocation basé sur le coût d'opportunité de l'eau

pour différents utilisateurs. Cependant, les critères les plus courants sont (Howe et al. 1986 in Ariel et al. 1997) :

1. **La flexibilité** : Elle signifie la souplesse de changement d'allocation de la ressource en eau d'une utilisation à l'autre lorsque la demande en eau change, sans compter la position spatiale des usagers.
2. **La sécurité** : L'alimentation en eau pour les utilisateurs permanents notamment, doit être assurée. La flexibilité ne contrarie pas la sécurité tant qu'il y a une réserve d'eau suffisante pour répondre aux différents besoins des utilisateurs.
3. **Le coût réel de l'opportunité** : Le coût réel des opportunités de fourniture de la ressource est payé par les utilisateurs, ce qui permet au gestionnaire de tenir en compte des utilisations environnementales n'ayant pas une valeur marchande (comme la fourniture d'un habitat pour la faune). Cela dirige l'orientation de la ressource vers des activités ayant les valeurs alternatives les plus élevées.
4. **La prévisibilité** : Les résultats d'une stratégie d'allocation doivent être prévisibles. Ceci permet la réalisation de la stratégie et réduire l'incertitude au minimum.
5. **L'équité** : L'équité de la stratégie d'allocation devrait être perçue par les différents utilisateurs notamment les potentiels.
6. **L'acceptabilité politique et publique** : La stratégie de répartition de la ressource en eau doit servir les différents objectifs. Cela permet son acceptation et son adoption par la société.

Dans cette étude, la stratégie de répartition de la ressource en eau du réservoir Hammam Boughrara entre les usagers (alimentation en eau potable, industrie et irrigation) n'est pas détaillée. En effet, la somme des besoins prévus (sans différenciation) est considérée dans l'optimisation de son fonctionnement. Cette stratégie de répartition pourra faire l'objet d'une étude ultérieure.

#### 1.3.1.2 Stratégie de distribution temporelle

Un équilibre doit être maintenu entre l'eau prélevée pour les différents usages et celle apportée par le ruissellement de surface. Normalement, le ruissellement pendant les périodes des hautes eaux (l'hiver généralement) dépasse de loin la demande d'approvisionnement de sorte que l'excès d'eau peut être stocké et utilisé lorsque le ruissellement est inférieur à la demande des consommateurs. Un problème majeur se produit quand il y a un hiver sec, de telle manière l'excès prévu d'eau ne se produit pas, en impactant sur les réservoirs par une insuffisance de

remplissage au début de l'été. Dans ces circonstances, des pénuries d'eau peuvent se produire même si l'été n'est pas excessivement sec (Gray 2005).

La stratégie (ou règle) de distribution temporelle vise à répondre à la question principale posée par le gestionnaire d'un réservoir. Cette question peut être reformulée comme : "Est ce qu'il y a l'intérêt à stocker l'eau à un moment donné ou à lâcher?" (François 2013), ou bien "Quelle est la quantité d'eau à lâcher à un moment donné permettant de maximiser le profit de gestion?". Selon (Moy et al. 1986), la règle de gestion est tributaire à la capacité de stockage (volume utile) du réservoir et peut être construite au moyen de l'historique de l'apport et de demandes. La plupart des études établies et publiées dans la thématique de la gestion opérationnelle des réservoirs pointent sur la détermination des règles de distribution de la ressource en eau issues de ces ouvrages au fil du temps (Labadie 2004; Rani and Moreira 2010; Ahmad et al. 2014).

Les règles de gestion sont exprimées sous différentes formes selon les enjeux d'exploitation du réservoir. Cependant, la représentation graphique sous forme de courbes guides est souvent utilisée (François 2013). Ces courbes peuvent exprimer la décision préconisée sur les lâchers en fonction de l'état de stockage du réservoir (volume ou niveau d'eau dans la retenue) et les apports (figures 1.2a et 1.2c), volume ou niveau d'eau requis en fonction du temps (figure 1.2c), les lâchers en fonction du stockage (figure 1.2d), ... etc.

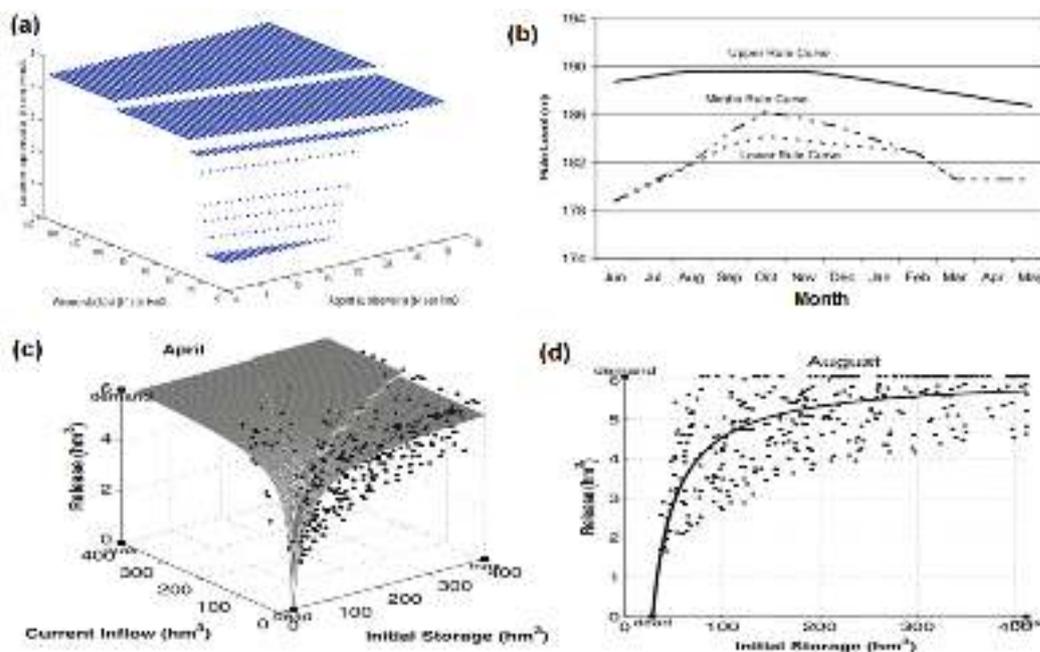


FIGURE 1.2 – Règles de distribution temporelle des lâchers. (a) réservoir Hammam Boughrara (Dahmani et al. 2018, annexes), (b) réservoir Dharoi (Inde) (François 2013), (c et d) réservoir Epitacio Pessoa (Brésil) (Celeste and Billib 2009)

Dans cette étude, nous allons concentrer sur la dérivation des politiques de gestion

temporelle de la ressource en eau du réservoir Hammam Bouhrara. En effet, les sections suivantes concernent principalement les approches utilisées dans l'élaboration des règles de distribution temporelle de la ressource en eau délivrée par les réservoirs.

## **1.4 Approches d'optimisation de la gestion opérationnelle d'un réservoir**

D'une manière générale, les modèles d'optimisation appliqués dans le domaine de la gestion des réservoirs peuvent être classés selon trois (03) approches (Celeste and Billib 2009), à savoir :

1. Approche d'optimisation stochastique implicite (Implicit Stochastic Optimization "ISO"),
2. Approche d'optimisation stochastique explicite (Explicit Stochastic Optimization "ESO"),
3. Approche de paramétrage-simulation-optimisation (Parameterization – Simulation – Optimization "P-S-O").

### **1.4.1 Approche d'optimisation stochastique implicite**

Les modèles basés sur l'approche implicite (ISO) utilisent l'optimisation déterministe pour produire la (ou les) solution (décision) optimale. Dans l'optimisation déterministe aucune notion d'incertitude n'est explicitement présente à l'entrée de la méthode d'optimisation employée (Carpentier and Cohen 2017). L'expression mathématique 1.1 connue sous l'appellation "fonction objectif" (section 1.5.1) est à minimiser ou à maximiser pour optimiser le fonctionnement d'un réservoir en connaissant au départ son état du stockage initial ( $S_{t=t_0}$ ) et les apports à différents moments ( $Q_t$ ). En effet, plusieurs scénarios d'apports probables sont générés puis utilisés pour aboutir aux politiques de gestion optimales. Par une autre manière, l'optimisation s'effectue en considérant que les apports sont connus a priori. La multiplication des scénarios permet d'intégrer implicitement la stochasticité des apports dans ce cas. Par la suite, l'ensemble des résultats obtenus sont examinés pour développer les courbes règles de gestion du réservoir (figure 1.3). La stochasticité des demandes est peut-être traitée de la même manière, c-à-d., par multiplication des scénarios (Seifi and Hipel 2001). Une courbe règle de gestion exprimant la variation des lâchers optimaux en fonction de l'état de stockage (au début de la période) plus l'apport est peut-être élaborée pour chaque mois par exemple. Donc, l'approche ISO tente à trouver la forme de la courbe règle de gestion du réservoir en exploitant les données produites par l'optimisation. La concrétisation d'un modèle d'optimisation à l'appui de cette approche consiste à réaliser les étapes suivantes (Celeste and Billib 2009) :

1. Générer  $M$  séquences d'apports synthétiques pour une durée de  $T$  mois ;
2. Trouver les lâchers optimaux pour chaque séquence en utilisant une méthode d'optimisation déterministe (chapitre 2) ;
3. Utiliser l'ensemble des séquences d'apports et des lâchers optimisés ( $M \times T$ ) pour développer les règles optimales de gestion.

$$F_{obj} = \text{Maximiser (ou Minimiser)} \left\{ \sum_{t=t_0}^{t=T} f(R_t, S_t, Q_t) \right\} \quad (1.1)$$

Tels que :

- $F_{obj}$  : fonction exprimant l'objectif d'optimisation ;
- $R_t$  : lâcher (à optimiser) par le réservoir au temps  $t$  ;
- $S_t$  : l'état de stockage du réservoir au temps  $t$  ;
- $Q_t$  : l'apport (entrée) du réservoir au temps  $t$ .

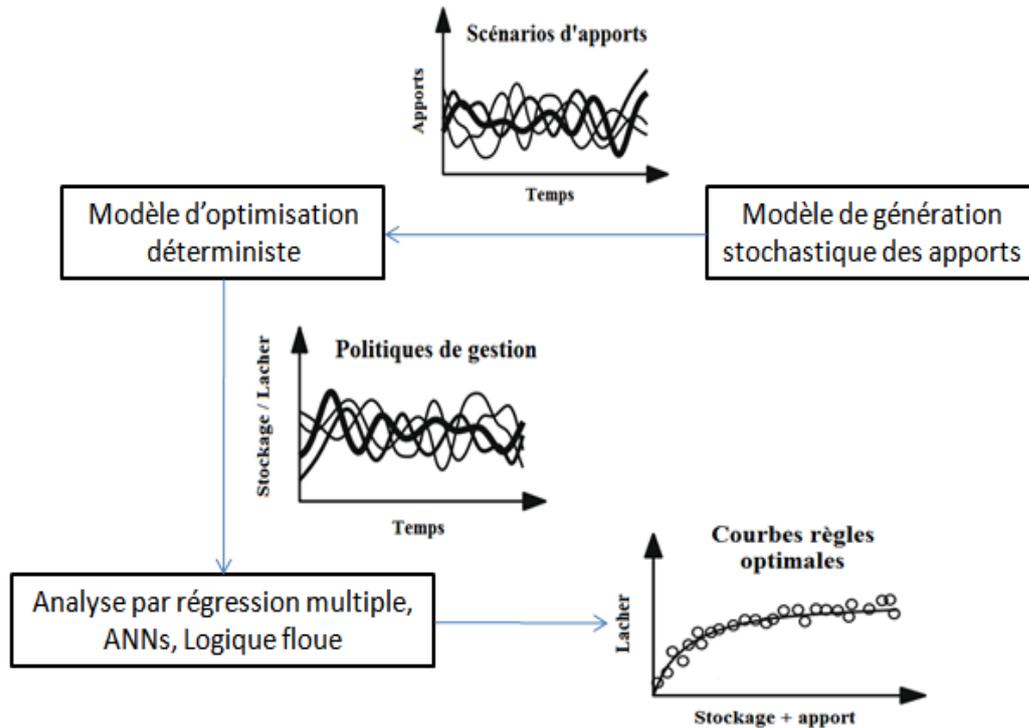


FIGURE 1.3 – Élaboration des courbes règles optimales de la gestion opérationnelle d'un réservoir par l'approche d'optimisation stochastique implicite "ISO". Adapté de (Celeste and Billib 2009)

L'approche ISO a été employée partiellement (les 02 premières étapes) dans beaucoup d'études d'optimisation du fonctionnement des réservoirs telles que (Ahmed and Sarma 2005; Nagesh Kumar et al. 2006; Ghimire and Reddy 2013; Devisree and Nowshaja 2014; Karami et al. 2018; Yaseen et al. 2018; ... etc).

Cette approche est adoptée dans la suite de ce travail, néanmoins elle n'est pas appliquée intégralement. Seulement les deux (02) premières étapes qui ont été réalisées et discutées (la prévision des apports et la dérivation des politiques de gestion optimales, figure 1.4). Car l'élaboration des règles de gestion nécessite au départ des modèles performants de prévision et d'optimisation, sur lesquels ce travail concentre. Les sorties des deux étapes indiquées peuvent être utilisés par la suite pour synthétiser les courbes règles via une analyse de régression multiple (ou autres telles que les réseaux de neurones artificiels, systèmes flous, apprentissage approfondi "Deep-Learning", ... etc).

Selon (Oliveira and Maia 2018), plusieurs modèles sont utilisés pour la génération stochastique des écoulements des cours d'eau, tels que les modèles d'autorégression (AR) et leurs dérivés (ARMA, ARIMA, ...), les ondelettes, K-plus proches voisins (K-Nearest Neighbors, KNN), les réseaux de neurones artificiels (ANNs), ... etc. Comme évoqué précédemment, les ANNs sont adoptés dans cette étude.

### **1.4.2 Approche d'optimisation stochastique explicite**

Les modèles stochastiques explicites (ESO) sont ceux intégrant la probabilité sur les variables aléatoires directement dans la méthode d'optimisation (Zéphyr 2015). Dans cette approche, les variables d'intérêt telles que les apports liquides sont considérées comme variables aléatoires, où les réalisations de ces dernières à un moment  $t$  sont exprimées par les distributions statistiques observées dans le passé de ces variables. L'objectif d'optimisation d'un réservoir est réexprimé dans ce cas par l'équation 1.2 (François 2013). La programmation dynamique stochastique (Stochastic Dynamic Programming "SDP") est la méthode la plus populaire de cette approche (Celeste and Billib 2009). Elle emploie les chaînes de Markov pour exprimer l'évolution probabiliste de la variable aléatoire hydrologique (les apports) (Kelman et al. 1990). Dans notre papier (Dahmani et al. 2018, Annexes), la SDP est appliquée sur le réservoir Hammam Boughrara dans le but de dériver les règles de gestion optimales de son fonctionnement. La programmation dynamique stochastique est décrite avec plus de détails dans la section 1.7.3.2.

$$F_{obj} = \text{Maximiser (ou Minimiser)} \left[ E \left( \sum_{t=t_0}^{t=T} f(R_t, S_t, Q_t) \right) \right] \quad (1.2)$$

Tel que :

—  $E(\cdot)$  : l'espérance mathématique.

**1.4.3 Approche de paramétrage-simulation-optimisation**

Dans l'approche d'optimisation par paramétrage-simulation-optimisation (P-S-O), la recherche d'une règle de gestion opérationnelle d'un réservoir commence par une courbe règle pré-établie dont la forme est définie par un ensemble de paramètres (peu de paramètres). Au début, ces derniers sont choisis arbitrairement, puis à la base des scénarios d'apports prévus ou au moyen d'une longue série observée de ceux-ci, la simulation de fonctionnement du réservoir est lancée via cette courbe. Une évaluation de la performance du fonctionnement du réservoir est peut-être effectuée au moyen d'une fonction objectif à minimiser ou à maximiser. Si la courbe règle de gestion ne permet pas d'accomplir le (ou les) but désiré, leurs paramètres sont mis à jour à l'aide d'un algorithme d'optimisation déterministe. Puis, la simulation est refaite à la base de la courbe règle modifiée et sous les mêmes scénarios d'apports. Ce processus est répété jusqu'à ce que la courbe paramétrée produit une opération du fonctionnement optimale ou elle atteint un niveau de performance requis (Celeste and Billib 2009). La figure 1.4 illustre le schéma explicatif de déroulement du processus P-S-O.

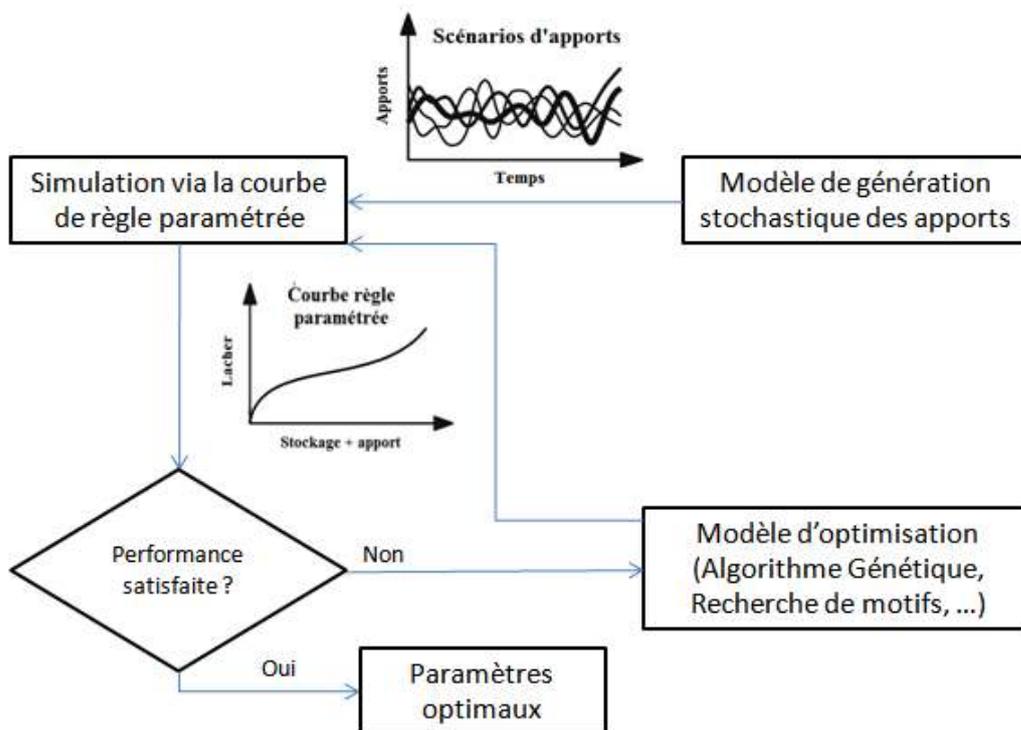


FIGURE 1.4 – Élaboration des courbes règles optimales de la gestion opérationnelle d'un réservoir par l'approche de paramétrisation–simulation-optimisation "P-S-O". Adapté de (Celeste and Billib 2009)

## **1.5 Modèle général d'optimisation du fonctionnement d'un réservoir**

Dans ce qui suit, on se limite au modèle général de la gestion opérationnelle d'un réservoir unique assurant un seul objectif d'exploitation ou plusieurs (objectifs multiples). Les approches d'optimisation déjà citées (sections 1.4.1, 1.4.2 et 1.4.3) incorporent tous ce modèle d'optimisation dans leur processus de traitement, car il comprend les principales équations gouvernant le fonctionnement d'un réservoir. Néanmoins, selon (Yeh 1985) il n'existe pas une formulation générale tolérant la résolution de tous les problèmes de gestion. En effet, d'autres modèles peuvent être substitués à partir du modèle générale pour traiter des problèmes spécifiques tels que l'optimisation de l'allocation de l'eau d'un réservoir pour l'irrigation (Reddy and Kumar 2007), l'optimisation de la génération de l'hydroélectricité (Nagesh Kumar and Janga Reddy 2007 ; Kamodkar and Regulwar 2010 ; François 2013 ; Nagy et al. 2013 ; Ghimire and Reddy 2013), l'optimisation de la qualité de l'eau des eaux délivrées par un réservoir (Dhar and Datta 2008), ... etc.

### **1.5.1 La fonction objectif**

Selon (Wurbs 1993), la fonction objectif (ou critère d'optimisation) constitue le cœur d'un modèle d'optimisation. Elle exprime d'une manière explicite le (ou les) objectif désiré par l'optimisation. Cette fonction prend des formes différentes et dépend principalement du problème à traiter et aux buts ciblés par le gestionnaire. (Wurbs 1993) a classé les fonctions objectifs rapportées dans la littérature de l'optimisation de la gestion des réservoirs selon trois catégories, à savoir :

1. Coûts et les avantages économiques (p.ex., minimisation des coûts de production, minimisation des dommages liés aux crues, maximisation des revenus économiques issus de l'allocation de l'eau, ...etc) ;
2. Production de l'énergie hydroélectrique (p.ex., maximisation de la production électrique et/ou leur revenu, minimisation des coûts de production, ... etc) ;
3. Fiabilité de systèmes et disponibilité de l'eau (p. ex., maximiser le taux de couverture de la demande en eau des consommateurs, minimiser les déversements, minimiser les défaillances et vulnérabilité d'un système de gestion, ... etc).

Néanmoins, selon (Lebdi et al. 1997), il est très rare d'exprimer la fonction objectif par des valeurs (revenus) monétaires, sauf le cas d'optimisation de la production d'hydroélectricité.

La gestion opérationnelle des réservoirs cible fréquemment l'atteint de plusieurs objectifs à la fois, qui pouvant être même contradictoires. En effet, plusieurs fonctions objectifs

sont à minimiser (ou à maximiser, ou encore à minimiser certaines et maximiser d'autres) en utilisant les méthodes d'optimisation dites "multi-objectifs" ("multi-objective optimization"). Néanmoins, selon (Talbi 2009; Yang et al. 2012) n'importe problème d'optimisation multi-objectif est peut-être reformulé pour être traité comme problème à objectif unique ("single-objective optimization"). Ce passage peut être établi par le regroupement des différents objectifs dans la même fonction objectif, où la pondération entre eux peut être utilisée pour favoriser un objectif par rapport à l'autre. Une autre manière est utilisable aussi, consiste à garder un seul objectif (principal) exprimé par la fonction et traiter les autres comme contraintes dans le modèle d'optimisation.

L'exemple ci-après exprime mathématiquement le passage d'une optimisation multi-objectifs à une autre mono-objectif. La première fonction objectif (équation 1.3) vise à maximiser la satisfaction des demandes en eau ( $D_t$ ) en aval d'un réservoir par les lâchers ( $R_t$ ), tandis que la deuxième fonction (équation 1.4) cible à garder un stock d'eau dans la retenue suivant une courbe guide de remplissage. Ces deux objectifs peuvent être contradictoires à certains moments et cohérents à certains autres. Le respect de la courbe guide de remplissage sollicité par la 2<sup>ème</sup> fonction favorise les lâchers (et par conséquent la réalisation de la 1<sup>ère</sup> fonction objectif) lorsque le stockage est important, et il les défavorise dans le cas contraire, c-à-d., lorsque le stockage diminue.

$$F_{objective-1} = \text{Minimiser} \sum_{t=t_0}^{t=T} (R_t - D_t)^2 \quad (1.3)$$

$$F_{objective-2} = \text{Minimiser} \sum_{t=t_0}^{t=T} |S_t - S_{gt}| \quad (1.4)$$

Avec :

- $D_t$  : la demande en eau au temps  $t$ ,
- $S_{gt}$  : stockage du réservoir requis par la courbe guide de remplissage.

Les deux objectifs peuvent être rassemblés simplement dans la même fonction à minimiser en ajoutant des coefficients de pondération ( $W_1$  et  $W_2$ ) si un objectif est prioritaire par rapport à l'autre (équation 1.5).

$$F_{objective-3} = \text{Minimiser} \left[ W_1 \times \sum_{t=t_0}^{t=T} (R_t - D_t)^2 + W_2 \times \sum_{t=t_0}^{t=T} |S_t - S_{gt}| \right] \quad (1.5)$$

De même, le 1<sup>er</sup> objectif est peut-être gardé dans l'optimisation en considérant le 2<sup>ème</sup>

comme contrainte. Dans ce cas une certaine pénalité est appliquée sur les solutions satisfaisant le 1<sup>er</sup> objectif dès l'infraction de la contrainte (équations 1.6 et 1.7). Aussi, il existe d'autres approches d'application des contraintes dans les processus d'optimisation telles que l'élimination des solutions avortant les contraintes, approche hybride, ... etc., (Yeniay 2005).

$$F_{objective-4} = Minimiser \left[ \sum_{t=t_0}^{t=T} (R_t - D_t)^2 + P_t \right] \quad (1.6)$$

Tel que :

- $P_t$  : une pénalité appliquée dès l'infraction de la contrainte sur le stockage cible. Elle peut être exprimée comme suit :

$$P_t = \begin{bmatrix} (S_t - S_{gt})^n, n \geq 1 \leftarrow \text{si } S_t \neq S_{gt} \\ 0 \leftarrow \text{sinon} \end{bmatrix} \quad (1.7)$$

## 1.5.2 Contraintes du système

En général, l'exploitation d'un réservoir est régie par les contraintes suivantes :

### 1.5.2.1 Équilibre de masses "Bilan hydrique"

Le bilan hydrique d'un réservoir s'exprime par l'équation suivante :

$$S_{t+1} = S_t + Q_t - R_t - Ev_t - I_t - O_t \quad t = 1, 2, \dots, T \quad (1.8)$$

Tels que :

- $S_t$  et  $S_{t+1}$  : le stockage (en  $Mm^3$ ) du réservoir au début et à la fin d'une durée  $t$  et  $t + 1$  respectivement ;
- $Q_t$  : l'apport liquide "entrée" du réservoir ( $Mm^3$ ) pendant la durée  $t$  ;
- $R_t$  : lâcher du réservoir "sortie" (en  $Mm^3$ ) pendant la durée  $t$  ;
- $Ev_t$ ,  $I_t$  et  $O_t$  : l'évaporation, l'infiltration et le déversement respectivement en ( $Mm^3$ ) pendant la durée  $t$  ;
- $T$  : la durée totale considérée de fonctionnement du réservoir.

Les pertes par évaporation peuvent être calculées comme suit :

$$Ev_t = ev_t A_t \quad (1.9)$$

Tels que :

- $ev_t$  : le taux d'évaporation en (m) pendant la durée  $t$ ;
- $A_t$  : la surface moyenne du plan d'eau du réservoir pendant la durée  $t$  (en  $m^2$ ).

### 1.5.2.2 Contraintes de stockage

Le fonctionnement normal d'un réservoir est soumis aux contraintes liées à sa capacité de stockage. Elles sont exprimées par les inégalités suivantes :

$$S_{min} \leq S_t \leq S_{max} \quad t = 1, 2, \dots, T \quad (1.10)$$

Tels que :

- $S_{min}$  : la capacité de stockage minimale de la retenue (en  $Mm^3$ ), correspondant au volume mort généralement ;
- $S_{max}$  : la capacité de stockage maximale de la retenue (en  $Mm^3$ ), au-delà, il y aura de déversement.

### 1.5.2.3 Contraintes de lâchers

La gestion du fonctionnement d'un réservoir s'articule en premier lieu sur la distribution de l'eau emmagasinée dans sa retenue au fil du temps sous forme des lâchers utiles pour les usagers. En effet, les déversements causés par les fortes crues sont exclus (considérés comme pertes) malgré qu'ils peuvent être bénéfiques pour l'agriculture et/ou l'écologie. Les lâchers ne doivent pas dépasser la somme des besoins des différents consommateurs de l'eau du réservoir à un moment donné. En général, les contraintes sur lâchers s'expriment par les inégalités suivantes :

$$0 \leq R_t \leq D_t \quad t = 1, 2, \dots, T \quad (1.11)$$

Tel que :

- $D_t$  : la demande en eau en aval du réservoir au temps  $t$  (en  $Mm^3$ ).

### 1.5.2.4 Contrainte de déversement

Durant le fonctionnement du réservoir, le volume déversé  $O_t$  est estimé par l'équation suivante :

$$O_t = \text{Max}((S_{t+1} - S_{max}), 0) \quad t = 1, 2, \dots, T \quad (1.12)$$

## 1.6 Modèle d'optimisation appliqué dans le cadre de l'étude

À la base du modèle générale de l'optimisation du fonctionnement d'un réservoir décrit ci-avant, nous avons dérivé un autre pour l'appliquer sur le cas d'étude envisagé (Barrage Hammam Boughrara, W. Tlemcen). Dans ce contexte, la manière d'appliquer les différentes contraintes liées au fonctionnement du réservoir dans le processus itératif d'optimisation est spécifiée. De plus, une fonction objectif unique est décrite à fin d'exprimer explicitement les buts d'exploitation de cet ouvrage.

### 1.6.1 Application des contraintes

L'approche d'application des contraintes par pénalités est adoptée dans cette étude. En effet, si les contraintes définies par les inégalités 1.10, 1.11 et 1.12 ne sont pas satisfaites, des pénalités sont appliquées dans les algorithmes durant le processus d'optimisation. Les pénalités utilisées dans cette étude sont rapportées dans (Jothiprakash and Shanthi 2006 ; Karami et al. 2018).

$$P_{1,t} = \left[ \begin{array}{l} \left( \frac{(S_{min} - S_{t+1})^2}{S_{min}} \right) \leftarrow \text{si } S_{t+1} < S_{min} \\ 0 \leftarrow \text{sinon} \end{array} \right] \quad (1.13)$$

$$P_{2,t} = \left[ \begin{array}{l} \left( \frac{(S_{t+1} - S_{max})^2}{S_{max}} \right) \leftarrow \text{si } S_{t+1} > S_{max} \\ 0 \leftarrow \text{sinon} \end{array} \right] \quad (1.14)$$

$$P_{3,t} = \left[ \begin{array}{l} \left( \frac{(R_t - D_t)^2}{D_{max}} \right) \leftarrow \text{si } R_t > D_t \\ 0 \leftarrow \text{sinon} \end{array} \right] \quad (1.15)$$

Tel que :

- $D_{max}$  : la demande en eau maximale de la durée de fonctionnement considérée  $T$  du réservoir (en  $Mm^3$ ).

### 1.6.2 Fonction objectif choisie

La fonction objectif (critère d'optimisation) est formulée selon les buts visés par le gestionnaire de l'ouvrage (les autorités compétentes). Malgré que le réservoir Hammam Boughrara est à usages multiples (AEP, industrie et irrigation), les demandes en eau des différents usagers peuvent être regroupées dans une seule demande globale. En effet, le recours à une seule fonction objectif est suffisant. L'unique fonction permettra d'utiliser des algorithmes d'optimisation mono-objectif qui présentent des architectures plus simples et plus maniables, sans oublier la simplicité des structures des résultats aboutis à la fin de traitement par rapports aux algorithmes d'optimisation multi-objectifs.

Dans cette étude, réduire (minimiser) les déficits d'approvisionnement en eau du réservoir (équation 1.16) est le but d'optimisation. Par autre manière, les lâchers durant une durée de fonctionnement du réservoir doivent satisfaire les demandes en eau au maximum possible. Par conséquent, les volumes de lâchers ( $R_t$ ) sont considérés comme variables de décision.

$$F_{obj} = \text{Minimiser} \sum_{t=1}^T \left( \frac{R_t - D_t}{D_{max}} \right)^2 \quad (1.16)$$

Tel que :

- $D_{max}$  : la demande en eau maximale en aval au cours de la période d'exploitation totale  $T$  considérée.

En appliquant les différentes pénalités, le critère d'optimisation est réécrit comme suit :

$$F_{obj} = \text{Minimiser} \sum_{t=1}^T \left[ \left( \frac{R_t - D_t}{D_{max}} \right)^2 + P_{1,t} + P_{2,t} + P_{3,t} \right] \quad (1.17)$$

Cette dernière formulation de la fonction objectif (équation 1.17) est utilisée dans la phase d'optimisation de la gestion opérationnelle du réservoir Hammam Boughrara présentée dans le chapitre 9, dans lequel a été traitée séparément par les quatre algorithmes évolutionnaires RGA, GSA, HPSOGWO et IGWO.

## 1.7 Méthodes d'optimisation appliquées : État de l'art

L'optimisation de la gestion opérationnelle des réservoirs et des systèmes de réservoirs est un domaine vaste et fait appel à plusieurs disciplines notamment la recherche opérationnelle. Cette dernière s'intéresse aux modèles de l'aide à la décision et aux différentes approches et méthodes d'optimisation.

Dans la suite, nous concentrons sur l'optimisation de la gestion opérationnelle d'un réservoir unique et les techniques d'optimisation utilisées, notamment les plus récentes. Par conséquent, la gestion des systèmes de réservoirs (ensemble de réservoirs interconnectés) ne sera pas abordée. Sans oublier que les méthodes d'optimisation sont applicables dans les deux cas, mais avec performances différenciables. Pour le lecteur intéressé, un passage en revue sur l'optimisation de tels systèmes est décrit dans (Wurbs 1991; Wurbs 1993; Labadie 2004) et autres ...

La littérature s'enrichit chaque année par des nouvelles méthodes et techniques d'optimisation dont chacune a ses avantages et inconvénients. Selon (Ahmad et al. 2014) les méthodes d'optimisation utilisées dans le traitement de la gestion opérationnelle des réservoirs peuvent être classées selon l'organigramme montré dans la figure 1.5.

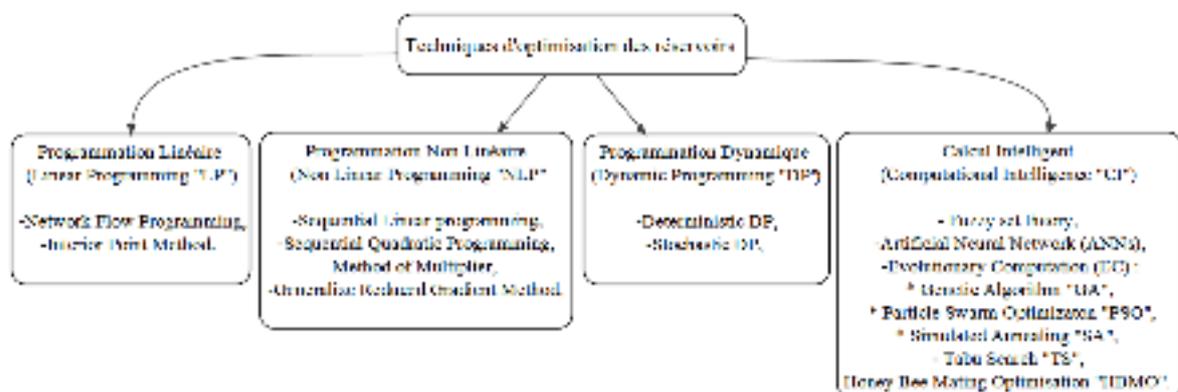


FIGURE 1.5 – Classification des méthodes d'optimisation de la gestion des réservoirs (Ahmad et al. 2014)

### 1.7.1 La programmation linéaire

La programmation linéaire (Linear Programming "LP") est utilisable lorsque la fonction objectif et les contraintes sont linéaires. Hormis, l'approximation des fonctions non-linéaires à d'autres linéaires permet d'étendre l'usage de cette méthode (Arunkumar and Jothiprakash 2012). La méthode LP est peut-être la plus utilisée dans le domaine de l'optimisation depuis

longtemps. Sa popularité est supportée par ses principaux avantages, la capacité de convergence à la solution optimale globale, l'applicabilité sur des problèmes à grandes échelles (nombre de variables relativement important), et sa disponibilité et support dans différents logiciels tels que MATLAB, LINGO, SciLab, CPLEX, MOSEK, ... etc. Dans la gestion des réservoirs, la méthode LP est utilisée avec l'approche d'optimisation implicite (ISO, section 1.4.1). Les inconvénients majeurs de cette technique sont l'incapacité de traiter les problèmes dont la fonction objectif et les contraintes ne sont pas linéaires ou non-convexes (Ahmad et al. 2014; Rani and Moreira 2010). Trois principaux cheminements d'amélioration de cette méthode ont séduit les chercheurs. Le premier est de surmonter son incapacité à traiter les problèmes dont la fonction objectif et les contraintes sont non-linéaires. Dans ce contexte, des méthodes sont proposées telles que LP non linéaire (NLP), LP séparable, LP successive, ... etc., (section 1.7.2). Le deuxième cheminement est de franchir la barrière de non-convexité. Pour ceci, des extensions sont proposées telles que LP binaire "Binary LP", LP entier "Integer LP", LP entier mixte "Mixed Integer LP", ... etc. Le dernier cheminement consiste à intégrer la stochasticité explicitement dans la méthode. Pour ce dernier souci, des évolutions sont suggérées telles que LP stochastique "stochastic LP", LP à chance limitée "Chance Constrained LP", ... etc. La stochasticité est intégrée en considérant que les apports suivent une chaîne de Markov (Rani and Moreira 2010).

La programmation des flux réseaux (Network Flow Programming, "NFP") est une autre forme de la programmation linéaire (Rani and Moreira 2010). Elle est proposée par L. FORD et D. FULKERSON en 1962 pour l'objectif de déterminer le flux maximal permanent d'un point à l'autre dans un réseau soumis à des limitations de capacité sur les arcs (Ford Jr and Fulkerson 2015). À cause de sa performance, cette méthode est largement intégrée dans les logiciels de simulation et d'optimisation des systèmes de réservoirs dont la configuration ressemble à un réseau. Dans ce contexte, on trouve des logiciels tels que MODSIM, HEC-3, HEC-5, HEC-ResSim, WATHNET, DWRSMI, ARSP, WASP, EMNET, ... etc., (Rani and Moreira 2010).

La méthode du point intérieur (Interior-Point Method, "IPM") (Karmarkar 1984) est une autre variante performante de la programmation linéaire (Rani and Moreira 2010). (Seifi and Hipel 2001) a proposé un modèle d'optimisation stochastique pour les systèmes de réservoirs en se basant sur l'IPM. Les auteurs ont appliqué le modèle sur le système de réservoir des grands lacs (Amérique du nord, Canada et les États-Unis). Les résultats trouvés indiquent que l'algorithme appliqué surmonte le problème de dimensionnalité (connu sous : "*fléau de la dimension*" ou "*malédiction de la dimensionnalité*", ou encore "*curse of dimensionality*" en anglais). Aussi, il permet d'intégrer la non-linéarité et d'envisager plusieurs scénarios d'apports (et de demande en eau), ce qui donne lieu aux politiques de gestion plus réalistes.

Une version appelée "Primal-Dual Interior Point (P-DIP)" a été développée pour trai-

ter les problèmes d'optimisation de haute dimensionnalité. La méthode P-DIP a été employée efficacement pour traiter un problème d'optimisation avec fonction objectif non-linéaire dont le nombre de variables et de contraintes sont 12'000 et 6'000 respectivement par (Mousavi et al. 2004). Une comparaison a été effectuée entre la méthode P-DIP, LP" (via LINGO) et la programmation dynamique "DP" (section 1.7.3). Les résultats ont marqué l'échec des deux derniers contre le P-DIP (Rani and Moreira 2010). L'algorithme P-DIP est implémenté dans plusieurs applications mathématiques telles que GLPK, LOQO, SDPA, GGPLAB, ... etc (swMath.org 2020).

### **1.7.2 La programmation non-linéaire**

La majorité des modèles d'optimisation du fonctionnement des réservoirs et des systèmes de réservoirs incorporent des équations non-linéaires dans leur formulation (telles que les fonctions objectifs, équations de génération de l'hydro-électricité, ... etc). Comme évoqué précédemment, la programmation linéaire est appliquée soit successivement ou après approximation de ces fonctions. Aussi, d'autres méthodes de programmation non-linéaires (NLP) sont proposées dans la littérature, telles que la programmation séquentielle quadratique "esequential Quadratic Programming", méthode de multiplicateur "method of multiplier", la méthode de gradient réduit "generalize reduced gradient method", ... etc., (Rani and Moreira 2010; Ahmad et al. 2014). Les implémentations de ces algorithmes sont disponibles dans plusieurs logiciels tels que Matlab, SciLab, LINGO, LOQO, ... etc., (swMath.org 2020). Les inconvénients majeurs des méthodes NLP sont la limitation devant les problèmes à grandes échelle et à haute non-linéarité, où elles peuvent facilement être piégées dans des optimales locales et rater les solutions globales (Janga Reddy and Nagesh Kumar 2020).

### **1.7.3 La programmation dynamique**

La programmation dynamique (Dynamic Programming "DP") et ses variantes (programmation dynamique déterministe "DDP", stochastique "SDP", ... etc) sont parmi les méthodes les plus utilisées dans la problématique de gestion opérationnelle des réservoirs et des systèmes de réservoirs. La méthode DP a été proposée par (Bellman 1957). Elle est basée sur le principe portant le nom de l'auteur ("principe d'optimalité de BELLMAN"). Ce principe est évoqué par :

*"Une suite de commandes optimales dans un processus d'optimalité dynamique est telle que quels que soient l'état et l'instant considérés sur une trajectoire optimale, les commandes ultérieures constituent pour le problème ayant cet état et cet instant comme éléments initiaux, une suite de commandes optimales ..."* (Bellman 1957 in Parent 1991).

La programmation dynamique se compose en deux (02) phases séparées. La première est dite d'optimisation. Elle conduit à la constitution d'une grille de valeurs de BELLMAN. La deuxième est une phase de simulation, dans laquelle une suite de décisions optimales est calculée à la base de la grille d'optimisation. Dans le cas de l'optimisation du fonctionnement d'un réservoir, à l'issue de cette dernière phase la trajectoire optimale de la réserve (et/ou de lâchers) est calculée (François 2013).

L'adoption de la DP est encouragée par ses multitudes avantages. La méthode permet de calculer la meilleure séquence de décisions (commandes) pour un état possible de système et prendre en compte les fonctions objectifs linéaires, non-linéaires et non-linéarisables (Parent 1991). Aussi, elle amène au balayage de l'espace de recherche par combinaison des variables discrétisées (c-à-d., discrétisation du temps par pas  $\Delta t$ , du volume en classes par pas  $\Delta S$ , les apports en classes par pas  $\Delta Q$ , ... etc). Ce balayage peut garantir la décision optimale globale du problème. De plus, avec sa version stochastique (SDP), elle admet l'inclusion de la stochasticité des variables aléatoires directement (explicitement) dans le traitement au contraire aux autres méthodes.

Néanmoins, la programmation dynamique dispose des limitations défavorisant son emploi à grandes échelles. Garantir l'optimal global (évident) nécessite la discrétisation fine des espaces de variables (temps, stockage, apports, ... etc). Cela conduit à l'augmentation exponentielle des combinaisons possibles, et par conséquence l'explosion de la consommation des mémoires des machines et de la durée du calcul. Ce problème représente la "*malédiction de la dimensionnalité*" (Parent 1991 ; François 2013 ; Rani and Moreira 2010 ; Zéphyr 2015). À cause de ce problème, l'usage de la DP a été limité aux modèles à deux ou trois réservoirs seulement. Le tableau suivant (1.1) illustre l'augmentation de nombre d'états à évaluer par l'algorithme DP en fonction de nombre de réservoirs  $n$  et nombre de classes de discrétisation du stockage  $k$ . Le nombre d'états suit la fonction  $k^n$  (Zéphyr 2015). Sans oublier qu'à chaque état, la DP doit évaluer toutes les combinaisons (état - apport possible) aussi.

TABLE 1.1 – Augmentation exponentielle de complexité de la DP

$n$	$k$	nombre d'états
1	8	8
2	8	64
5	8	32'768
7	8	209'7152
10	8	1'073'741'824
12	8	68'719'476'736
15	8	3.51844E+13

### **1.7.3.1 La programmation dynamique déterministe**

Selon (Celeste and Billib 2009; Rani and Moreira 2010) les premières applications de la programmation dynamique déterministe (Deterministic Dynamic Programming "DDP") pour l'optimisation du fonctionnement des réservoirs ont été établies par (Young 1967; Hall et al. 1968). La DDP est utilisable lorsque l'ensemble des variables d'entrées d'un problème d'optimisation sont connues a priori (déterminées). En effet, la DDP est souvent employée dans l'approche d'optimisation stochastique implicite (ISO), où la fonction objectif à traiter ressemble à l'équation 1.1.

Plusieurs améliorations ont été effectuées sur la DDP dans le but de palier le problème du fléau de dimensionnalité. Dans ce cadre, des méthodes sont proposées telles que la DP incrémentielle (Incremental DP "IDP"), différentielle discrete DP (discrete differential DP "DDDP"), DP avec approximations successives (DP with successive approximations "DPSA"), DP incrémentielle avec approximations successives (DP with successive approximation "IDPSA"), DP itérative pliée (iterative folded DP "FDP"), ... etc. Aussi, d'autres versions sont introduites capables de manipuler les problèmes d'optimisation sans recours à la discrétisation des variables (Rani and Moreira 2010).

### **1.7.3.2 La programmation dynamique stochastique**

La programmation dynamique stochastique (SDP) est la méthode la plus utilisée dans l'approche d'optimisation stochastique explicite (ESO) du fonctionnement des réservoirs (Celeste and Billib 2009). Comme évoqué précédemment, la SDP intègre directement la probabilité sur les variables aléatoires (telles que les apports, les demandes en eau, ...) sans nécessité de multiplication des scénarios sur ces variables. Cette méthode est très convenable aux problèmes de gestion de stocks dans l'incertain au fil du temps. En conséquence, plusieurs études de gestion opérationnelle des réservoirs et systèmes de réservoirs ont été réalisées à la base de cette méthode telles que (Torabi and Mobasheri 1973; Kelman et al. 1990; Pabiot 1999; Jothiprakash and Shanthi 2009; Celeste and Billib 2009; François 2013; Zéphyr 2015; Dahmani et al. 2018, ... etc).

En revanche, le fléau de dimensionnalité reste un obstacle devant l'application de la SDP à grandes échelles (nombre de réservoirs supérieur à trois). La discrétisation de l'intervalle du stockage et des apports d'un réservoir affecte considérablement sa performance. Pour surmonter ce problème l'utilisation de l'agrégation – désagrégation des réservoirs et la décomposition successive unique "one-at-a-time" est très fréquente (Rani and Moreira 2010).

### Description brève de procédure d'application de la SDP

La procédure de traitement par la SDP commence par la discrétisation de volume du stockage à  $k$  intervalles. La même chose pour les apports, dont ils doivent être organisés en  $m$  classes. Les principales approches de discrétisation sont bien présentées dans (Klemeš 1977).

L'intégration de probabilité sur la variable hydrologique aléatoire (les apports) avec la programmation dynamique stochastique s'effectue en considérant les apports suivent une chaîne de Markov d'ordre un (I), c-à-d., l'apport à un instant  $t$  ne dépend que du précédent (celui à l'instant  $t - 1$ ). Dans ce cas, la *matrice* dite "*de transition*" est peut-être élaborée. Cette matrice exprime à quelle probabilité  $p_{ij}$  on peut passer à l'apport  $Q_j$  à l'instant  $t + 1$  à partir d'un apport  $Q_i$  à l'instant  $t$ .

Généralement, le processus d'optimisation par la SDP commence sur le plan temporel par un état futur (au temps  $T$ ) fixé a priori (réservoir plein par exemple) en revenant en arrière jusqu'à un moment  $t_0$ . Au cours de ce processus, la SDP construit une *grille* (dite "*de variable de BELLMAN*"). Cette grille est formée par les valeurs de l'équation suivante (Eq. 1.18) :

$$F(S_t) = \text{Max (ou Min)} \{f(Q_t, S_t, R_t) + E [F_{t+1}(S_{t+1})]\}, \quad t \in [t_0, T] \quad (1.18)$$

Tels que :

- $f(\cdot)$  : la fonction objectif à minimiser ou à maximiser,
- $F(\cdot)$  : fonction exprimant le revenu actuel et futur espéré de la fonction objectif.

La résolution de cette équation concrétise la phase d'optimisation, car elle conduit à calculer les lâchers optimaux par exploration de toutes les combinaisons (états du stockage - apports possibles) et à construire une grille de (états du stockage - apports possibles - lâchers optimaux). La phase de simulation permet par la suite de déterminer les lâchers optimaux pour une séquence d'apports bien définie.

La méthode de programmation dynamique stochastique est appliquée sur le réservoir Hammam Boughrara (W. Tlemcen) (Dahmani et al. 2018, Annexes) pour déterminer sa règle de gestion optimale par l'approche ESO. Cependant, les demandes en eau considérées dans l'optimisation par la SDP sont de l'ancienne affectation de la ressource en eau du réservoir. Cette affectation est différente à celle adoptée dans la suite de cette étude. Ce qui empêche la comparaison judicieuse entre les résultats aboutis à la fin des deux travaux.

### **1.7.4 Méthodes de calcul intelligent**

Cette famille de méthodes englobe - entre autres - les trois catégories de techniques du calcul évolutionnaire, à savoir : les systèmes flous, les réseaux de neurones artificiels (ANNs) et les techniques d'essais intelligents (ou "algorithmes d'optimisation métaheuristiques"). Notons que l'expression "algorithmes de calcul évolutionnaire" désigne très souvent la dernière catégorie de méthodes, c-à-d., les algorithmes d'optimisation métaheuristiques. Ces dernières années, ces méthodes ont gagné de popularité dans le domaine de gestion des ressources en eau, y compris l'optimisation de la gestion opérationnelle des réservoirs à cause de leur capacité à surmonter les complexités limitant l'usage des anciennes méthodes telles que LP, NLP, DDP, SDP, ... etc., (Rani and Moreira 2010).

#### **1.7.4.1 Théorie des ensembles flous**

La théorie des ensembles flous (ou les systèmes flous) a été proposée par (Zadeh 1965). Ensuite, en 1970 le concept de l'optimisation floue est proposée par (Bellman and Zadeh 1970). Ce type de méthodes est proposé pour traiter les problèmes de prise de décisions incertaines, qualitatives ou imprécises. L'utilisation des expressions linguistiques pour définir les relations entre entrées et décisions est adoptée au lieu les équations mathématiques complexes.

Des applications réussies de systèmes flous dans l'optimisation de la gestion opérationnelle des réservoirs ont été établies telles que (Russell and Campbell 1996), couplés avec une métaheuristique (algorithme génétique) (Akter and Simonovic 2004), couplés avec les réseaux de neurones artificiels (Chaves and Kojiri 2007). Aussi, selon (Celeste and Billib 2009), les ANNs et les systèmes flous peuvent améliorer significativement les règles de gestion issues de l'approche d'optimisation implicite (ISO).

#### **1.7.4.2 Réseaux de neurones artificiels**

Les réseaux de neurones artificiels (Artificial Neural Networks, ANNs) sont des méthodes imitant artificiellement et de manière très simplifiée le processus de réflexion du cerveau humain. Ils sont appliqués fréquemment dans la prévision de pluviométrie et des apports aux réservoirs. Dans l'optimisation de fonctionnement de ces derniers, on trouve des applications pertinentes des ANNs couplés avec les systèmes flous (Chaves and Kojiri 2007) et les ANNs paramétrés par un algorithme métaheuristique (Algorithme Génétique "GA") (Chaves and Chang 2008). Par exemple, les auteurs de ce dernier article ont développé un système intelligent de réseaux de neurones artificiels évolutionnaires (Evolving Neural Network Intelligent System

"ENNIS") souple, basé sur les ANNs pour la construction d'un modèle d'opération de décisions multiples, et GA pour l'entraînement des modèles ANNs.

#### **1.7.4.3 Algorithmes d'optimisation évolutionnaires**

Les algorithmes d'optimisation évolutionnaires (EOAs) sont de nouvelles méthodes utilisées pour chercher des solutions approchées des problèmes d'optimisation avec un temps de calcul raisonnable et une précision acceptable. Ces méthodes disposent plusieurs avantages leurs permettent de vaincre les limitations des anciennes méthodes. Selon (Rani and Moreira 2010), les EOAs peuvent traiter la non-linéarité, la discontinuité, la discrétisation, l'incertitude et la multi-objectivité des problèmes d'optimisation sans problèmes.

Récemment, plusieurs études d'optimisation du fonctionnement des réservoirs et de systèmes de réservoirs basées sur les approches ISO et P-S-O ont été établies au moyen des algorithmes EOAs. Les résultats aboutis montrent que ces derniers servent à des performances supérieures dans la plupart des cas par rapport aux anciennes méthodes. Les plus utilisés dans ce domaine sont les algorithmes génétiques (Genetic Algorithms "GAs") (Ahmed and Sarma 2005; Jothiprakash and Shanthi 2006; Jothiprakash and Shanthi 2009; Nagesh Kumar et al. 2006; Devisree and Nowshaja 2014; ...), l'algorithme d'optimisation des essaims de particules (Particle Swarm Optimization "PSO", Eberhart and Kennedy 1995) (Nagesh Kumar and Janga Reddy 2007; Reddy and Kumar 2007; Ghimire and Reddy 2013; ...). Le PSO a été couplé avec MODSIM (modèle hybride PSO-MODSIM) pour proposer des tailles optimales des installations de stockage et de transfert d'eau prévues dans le bassin en amont de Sirvan en Iran (Rani and Moreira 2010). D'autres algorithmes sont proposés tels que l'algorithme de recherche gravitationnelle (Gravitational Search Algorithm "GSA") (Bozorg-Haddad et al. 2016; ...), l'algorithme d'optimisation des mauvaises herbes (Weed Optimization Algorithm "WOA") (Karami et al. 2018) ... etc.

Comme évoqué précédemment, quatre algorithmes métaheuristiques (RGA, GSA, HPSOGWO et IGWO) sont choisis pour réaliser (séparément) la phase d'optimisation, c-à-d., la génération des politiques optimales de gestion du réservoir Hammam Boughrara. De même, ils sont utilisés dans des essais d'entraînement et de paramétrage des réseaux de neurones artificiels employés dans la phase de prévisions des apports au réservoir.

## **1.8 Conclusion**

Les réservoirs sont la pierre angulaire dans la gestion des ressources en eau. À travers ce chapitre, nous avons présenté le rôle et la nécessité des barrages-réservoirs dans la société

et pour le développement économique des pays, ainsi la protection de la vie humaine, qui est sans doute n'a pas de prix. Aussi, la gestion opérationnelle de ces ouvrages est discutée avec ses deux volets : 1) les règles de répartition de la ressource en eau entre usagers et 2) les règles de sa distribution temporelle. La confection de ces dernières est retenue comme piste du travail.

Un focus est appliqué sur les approches d'optimisation stochastique de fonctionnement des réservoirs aboutissant à la synthèse des meilleures règles de gestion. L'approche implicite "ISO" est adoptée à cause de sa simplicité. Elle revendique trois phases, la première est la génération stochastique des scénarios d'apports liquides. La deuxième est dérivation des politiques de gestion optimales via un modèle d'optimisation déterministe (toutes les entrées au modèle sont définies explicitement y compris les apports liquides, la consommation, ... etc). Les modèles déterministes emploient des algorithmes d'optimisation déterministes, qui sont plus simples par rapport aux autres stochastiques. La troisième phase consiste à analyser les sorties des deux étapes précédentes pour synthétiser les règles de gestion. Néanmoins, dans ce travail nous concentrons sur les deux premières phases uniquement car elles sont à la source des règles de gestion fiables.

Le modèle d'optimisation sous contraintes appliqué à la gestion d'un réservoir unique est discuté aussi. De même, la fonction objective représentant la finalité voulue d'une opération d'optimisation et les contraintes cadrant cette fonction dans les limites de fonctionnement du réservoir cas d'étude, sont déployées aussi.

Un passage en revue sur les différentes techniques d'optimisation utilisées dans le domaine est présenté. Les algorithmes métaheuristiques sont choisis comme outils d'optimisation dans ce travail. La présentation de cette famille de méthodes fait l'objet du chapitre suivant.



# Chapitre 2

## Les algorithmes métaheuristiques

### 2.1 Introduction

L'homme a toujours été adoré par l'idée de trouver les solutions optimales à ses problèmes (les plus courts chemins, l'usage minimal des ressources conduisant aux zéro perte ou gaspillage, maximum de revenus et bénéfices, ... etc). C'est à partir de ce désir que l'optimisation a vu sa naissance.

À cette époque, l'optimisation trouve ses applications un peu partout, dans le domaine scientifique, technologique, industriel, financier, ... etc. Deux (02) opérations majeures constituent l'optimisation, la minimisation et la maximisation. Dans les cas pratiques, l'optimisation touche fréquemment la minimisation du temps, coûts (frais), risques, défaillances, vulnérabilité, ... etc., ou, la maximisation des profits (bénéfices), qualité, efficacité, fiabilité, résilience, ... etc.

La méthode (ou "l'algorithme") d'optimisation constitue l'un des trois piliers emmenant à la résolution efficace d'un problème d'optimisation, à côté d'une représentation fiable des processus physiques intervenant dans le problème et un simulateur numérique efficace permettant d'évaluer les solutions prises (Yang et al. 2012). En effet, le perfectionnement d'un modèle d'optimisation efficient implique le choix judicieux de telle méthode.

Les méthodes d'optimisation dites classiques, telles que la programmation linéaire (LP) et la programmation dynamique (DP) sont largement explorées et exploitées dans la littérature de l'optimisation du fonctionnement des réservoirs et systèmes de réservoirs. Malgré leurs avantages, elles présentent de multitudes inconvénients, parmi lesquels ils figurent le problème de dimensionnalité, la nécessité de discrétisation des variables, le temps de calcul important, la consommation excessive des ressources d'ordinateurs (mémoire et traitement), ... etc.

Les métaheuristiques sont des nouvelles méthodes conçues dans l'espérance de pallier les désavantages des anciennes. Principalement, elles peuvent produire des solutions de qualité "meilleure" ou "bonne" dans un temps de traitement raisonnable. Durant les trois dernières décades, cette classe de méthodes a gagné beaucoup d'intérêt dans le domaine de l'optimisation en général et l'intelligence artificielle en particulier.

Le présent chapitre est consacré à l'optimisation mathématique en gros et aux métaheuristiques notamment, avec une concentration sur les méthodes dites "algorithmes évolutionnaires", sans oublier les motivations derrière leur rétention dans cette étude. Dans ces derniers contextes, les quatre algorithmes utilisés dans la suite de ce travail sont bien détaillés, à savoir : une version de l'algorithme génétique à codage réel (RGA), l'algorithme de recherche gravitationnelle (GSA), l'optimiseur de Loup Gris amélioré (IGWO) et l'algorithme hybride de l'essaim de particules et l'optimiseur de Loup Gris (HPSOGWO).

## 2.2 Généralités sur l'optimisation

Les travaux sur l'optimisation ont commencé probablement en Grèce. Euclide, au *III<sup>ème</sup>* siècle avant J.-C., avait formulé des problèmes d'optimisation dans son livre "*Les éléments*" (Dodge 2006). Dans le moyen age, Johannes Kepler (1571-1630), mathématicien et astronome allemand (travaillé notamment sur l'optique et la géométrie) a trouvé les dimensions optimales d'un tonneau de vin. Par la suite, le mathématicien anglais Isaac Newton (1642-1727) et l'allemand Gottfried Leibniz (1646-1716) ont formulé les bases de calcul de variation des fonctions, lesquelles ont contribué ensuite, au développement des techniques d'optimisation. En 1917, le mathématicien américain Harris Hancock (1867-1944) a publié le premier livre porté sur l'optimisation "Theory of Maxima and Minima".

Mathématiquement, un problème d'optimisation peut être défini par le couple  $(X, f)$ , avec :

- $X$  : représente l'ensemble des "*solutions faisables*", ou "*l'espace de recherche*";
- La fonction  $f : X \rightarrow \mathfrak{R}$ , définie sur l'ensemble  $X$  les valeurs dans l'ensemble  $\mathfrak{R}$ .

Dans le cas de minimisation, l'optimisation consiste à trouver un élément  $x^*$  de  $X$  tel que :

$$f(x^*) \leq f(x), \forall x \in X. \quad (2.1)$$

Par contre, dans le cas de maximisation, il faut trouver l'élément  $x^*$  de  $X$  tel que :

$$f(x^*) \geq f(x), \forall x \in X. \quad (2.2)$$

Dans ce cas,  $f(x^*)$  est appelé un extremum (minimum ou maximum) global, et l'élément  $x^*$  est la solution optimale (minimale ou maximale) globale. La figure 2.1 démontre un exemple explicatif de la différence entre le minimum global et un autre local.

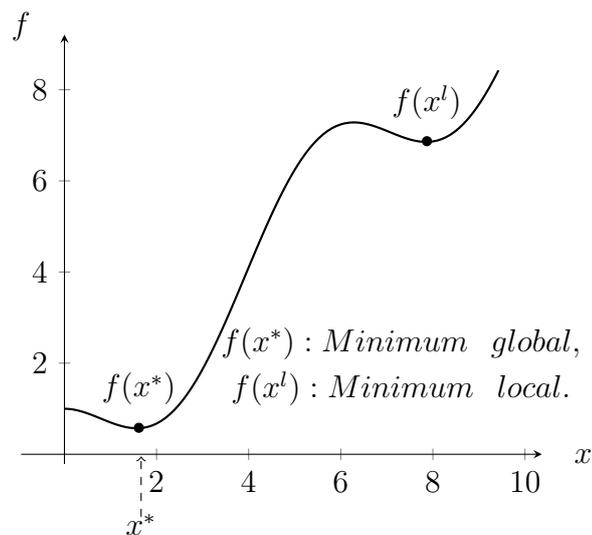


FIGURE 2.1 – Minimum local et global d'une fonction

D'une manière générale, les problèmes pratiques d'optimisation s'expriment par au moins une "fonction objectif" ou "critère d'optimisation" (équation 2.3) traduisant la finalité d'optimisation (trouver l'optimal global). Si  $M = 1$  dans l'équation 2.3, l'optimisation est dite "mono objectif" (ou à "simple objectif" ou encore "mono critère"), dans le cas contraire elle est multiobjectif". La (ou les) fonction objectif est peut-être soumise ou non à une ou plusieurs contraintes (équations 2.4 et 2.5). Si aucune contrainte ne se présente, l'optimisation est dite "libre". Dans le cas contraire, elle est "sous contraintes" (Talbi 2009; Yang et al. 2012).

$$\text{Minimiser (ou maximiser) } f_i(x), \quad i = 1, 2, \dots, M \quad (2.3)$$

Sous contraintes :

$$h_j(x) = 0, \quad j = 1, 2, \dots, J \quad (2.4)$$

et

$$g_k(x) \leq 0, \quad k = 1, 2, \dots, K \quad (2.5)$$

## **2.3 Classification des méthodes d'optimisation**

Les méthodes d'optimisation peuvent être classifiées selon plusieurs points de vue, à savoir (Talbi 2009 ; Yang et al. 2012) :

**§1 À base de gradient et à gradient libre** : Les méthodes à base de gradient (ou à base de dérivation) sont celles qui utilisent l'information sur gradient (restituée de dérivées des fonctions objectifs) dans le processus d'optimisation (telles que la méthode de Gauss-Newton, Levenberg-Marquardt, ... etc). Au contraire, les méthodes à gradient libre (telles que les métaheuristiques) utilisent directement l'information rapportée par la (ou les) fonction objectif.

**§2 Déterministes et stochastiques** : Si l'algorithme d'optimisation n'implémente aucun mécanisme "*interne*" de recherche aléatoire dans son processus de fonctionnement, il est classé comme "*déterministe*". Dans le cas contraire, il est répertorié comme "*stochastique*". Les méthodes classiques telles que la programmation linéaire, la programmation dynamique, ... etc., sont des méthodes déterministes. Par contre la majorité des algorithmes évolutionnaires sont des méthodes stochastiques.

**§3 Exactes et approximatives** (figure 2.2) : Les méthodes exactes obtiennent des solutions optimales et garantissent leur optimalité. Selon la complexité de problème, ces méthodes exigent une consommation excessive (voir parfois énorme ou non envisageable) du temps et des ressources de calcul. Cependant, les méthodes approximatives (ou heuristiques) génèrent des solutions de haute qualité dans un temps de traitement raisonnable pour une utilisation pratique, mais il n'y a aucune garantie de trouver une solution optimale globale (Talbi 2009).

D'autres classifications peuvent être établies aussi, telles que Mono-objectif ou multi-objectif, de recherche locale ou globale, ... etc. Une classification plus détaillée est peut-être trouvée dans (Janga Reddy and Nagesh Kumar 2020).

## **2.4 Choix d'une méthode d'optimisation**

Le choix correct d'un algorithme d'optimisation pour traiter un problème spécifique ("*the right algorithm to the right problem*") est l'une des trois problématiques majeures de l'optimisation axée sur la simulation (à côté de l'efficacité des algorithmes d'optimisation et l'efficacité et la précision des simulateurs) (Yang et al. 2012). Sans oublier qu'il n'existe pas

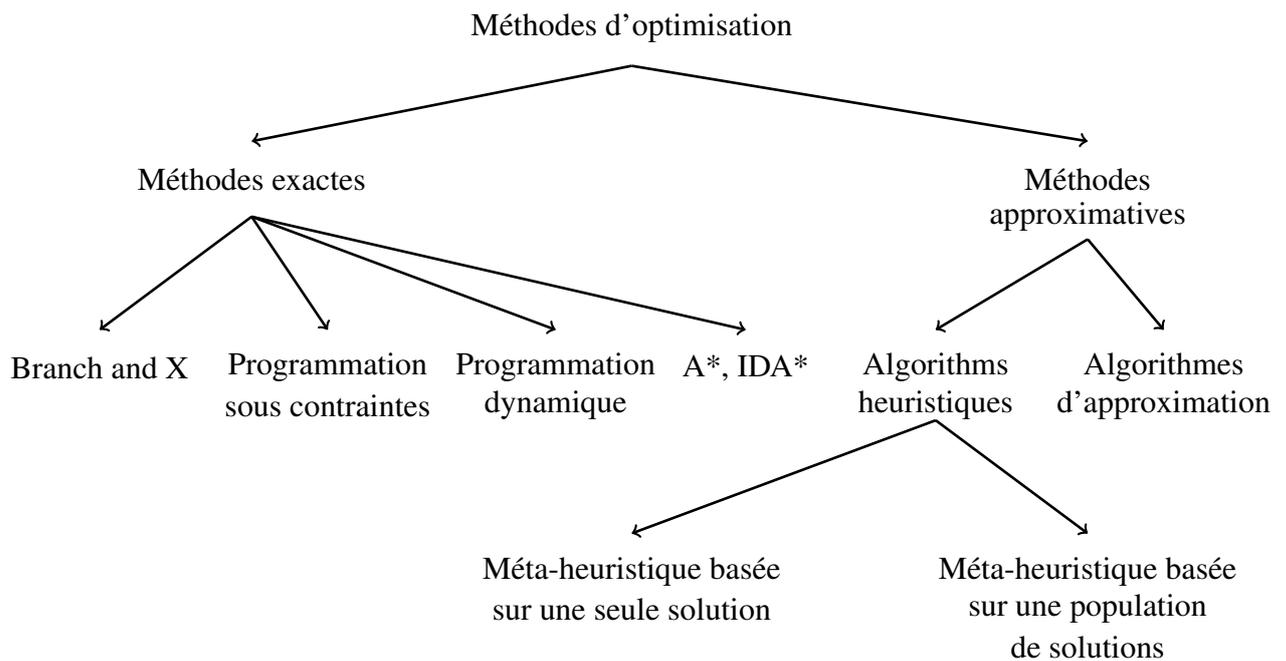


FIGURE 2.2 – Classification des méthodes d'optimisation. Adapté d'après (Talbi 2009)

un algorithme performant dans tous les domaines comme l'indique le théorème "*No free lunch*" proposé par (Wolpert and Macready 1997).

D'une manière générale, selon (Yang et al. 2012) le choix dépend de plusieurs facteurs qui peuvent être axés sur des appuis systématiques et raisonnables. Or, dans certains cas le choix ne se base que sous des préférences personnelles des décideurs ou concepteurs des modèles d'optimisation, à défaut qu'il n'existe pas une procédure de choix générale et bien ordonnée dans la littérature.

Les principaux facteurs du choix peuvent être résumés dans les points suivants :

**§1 La nature du problème et son complexité** : Les problèmes simples à complexité polynomiale, c-à-d, dont la résolution nécessite un temps et un espace (de mémoire) polynomiales en fonction de leurs tailles, peuvent être résolus avec les méthodes exactes (figure 2.2), sans recours aux celles approximatives (Talbi 2009). De même, les problèmes avec fonctions objectifs linaires peuvent être résolus directement par la programmation linéaire ou autre méthode équivalente. Par contre, pour les problèmes non-linéaires, multimodales (dont la fonction objectif disposent de plusieurs minimas locaux) ou de haute dimensionnalité, le recours aux méthodes approchées peut être le seul alternatif (Yang et al. 2012).

**§2 La nature de l'algorithme d'optimisation** : Chaque méthode d'optimisation présente des avantages et des inconvénients, ou même des limites. Par exemple, les méthodes à base

de gradient ne sont pas valables pour les problèmes à objectifs discontinus (Talbi 2009 ; Yang et al. 2012). De même, la programmation linéaire (LP) ne peut pas traiter les problèmes non-linéarisables. Aussi, les deux versions de la programmation dynamique (déterministe et stochastique) souffrent du fléau de dimensionnalité et ne sont pas exploitables pour les problèmes de haute dimensionnalité (à grande échelle), et par conséquent le recours aux métaheuristiques peut constituer le bon choix.

**§3 La qualité de solutions exigée** : La solution exacte (l'optimale globale certaine) est revendiquée pour certains problèmes sans considération de temps et coûts d'optimisation. Dans ce cas, l'appel aux méthodes exactes est le seul choix. Au contraire, il s'est avéré parfois de trouver seulement des solutions de bonne qualité satisfaisant des applications pratiques, de grande échelle (haute dimensionnalité) ou d'autres de fortes incertitudes (telles que l'optimisation de l'allocation de l'eau d'un réservoir pour l'irrigation, ..., recherche d'une configuration économique des réseaux d'alimentation en eau potable "AEP", d'électricité, ..., agencement des circuits électroniques, ... etc). Dans ce cas, le recours aux méthodes approchées peut constituer un bon chemin.

**§4 Temps et ressource de calcul** : Le temps du calcul est souvent un facteur déterminant. Sous l'effet que certains problèmes d'optimisation nécessitent un temps de traitement court ou parfois très court, l'adoption des algorithmes rapides constitue le seul alternatif (e.g., la commande automatique des appareils en temps réel, recherche de plus court chemin par un appareil GPS "Global Positioning System", ... etc). Cependant, si le temps et les ressources du calcul (capacité de traitement du (ou des) processeur(s) et mémoire(s) de l'ordinateur) autorisent l'usage des méthodes exactes, ces dernières constituent le bon choix dans ce cas. Si l'emploi des méthodes exactes implique un temps (qui peut aller jusqu'à plusieurs siècles pour certains problèmes) et des ressources excessives de l'ordinateur (à cause de fléau de dimensionnalité par exemple), le seul choix est accordé aux méthodes approchées telles que les métaheuristiques (Talbi 2009).

D'autres facteurs peuvent être pris en compte dans le choix d'un algorithme d'optimisation parmi autres, tels que le nombre et la nature des contraintes (Janga Reddy and Nagesh Kumar 2020). La disponibilité de l'algorithme (son implémentation en code informatique) contribue considérablement dans sa popularité (Talbi 2009 ; Yang et al. 2012), sans oublier aussi que les utilisateurs tentent habituellement à utiliser les algorithmes qu'ils maîtrisent (Yang et al. 2012). En effet, la richesse bibliographique participe favorablement à l'adoption d'un algorithme dans différents domaines. D'une manière générale, le choix est souvent basé sur multiples facteurs ou une balance entre eux (p.ex., une bonne qualité de solutions avec un temps de calcul raisonnable, la solution exacte dans un temps ne dépassant pas certaines limites ... etc).

## **2.5 Les métaheuristiques**

### **2.5.1 Définition**

Le mot métaheuristique ou "*Metaheuristic, en anglais*" a été utilisé pour la première fois par (Glover (1986)). Le mot est d'origine Grec, et subdivise en deux : "*Meta*" signifie "*Au-delà*" ou "*méthodologie de niveau supérieur*", et "*heuriskein*" signifie "*l'art de découvrir des nouvelles stratégies (règles) pour résoudre un problème*" (Talbi 2009). Le concept du *calcul heuristique* (qui signifie pratiquement de chercher une solution par essai et erreur (Yang et al. 2012)) est introduit la première fois à la fin de la seconde guerre mondiale (1939-1945) par (Pólya 1945).

Les métaheuristiques peuvent être définies comme : "*des méthodes de solution qui orchestrent une interaction entre les procédures d'amélioration locales et les stratégies de niveau supérieur pour créer un processus capable d'échapper aux optima locaux et d'effectuer une recherche robuste dans un espace de solutions. Au fil du temps, ces méthodes sont également venues inclure toutes les procédures qui emploient des stratégies pour surmonter le piège de l'optimalité locale dans l'espace de solutions complexe, en particulier les procédures qui utilisent une ou plusieurs structures de voisinage comme moyen de définir des mouvements admissibles à la transition d'une solution à l'autre, ou de construire ou détruire des solutions dans des processus constructifs et destructeurs*" (Glover and Kochenberger 2006).

Les métaheuristiques (désignés généralement par "algorithmes évolutionnaires d'optimisation, EOAs") constituent une famille de méthodes de calcul approximatif (figure 2.2). Ils sont capables de fournir des solutions "acceptables" ou "bonnes", ou même optimales globales des problèmes difficiles et complexes en science et ingénierie, dans un délai de calcul raisonnable (Talbi 2009). Les métaheuristiques peuvent manœuvrer la non-linéarité, la discontinuité, la discrétisation, l'incertitude et la multi-objectivité des problèmes d'optimisation sans problèmes (Rani and Moreira 2010).

Ceci est derrière leur popularité ces dernières années dans les différents domaines. Au contraire aux méthodes exactes, les métaheuristiques ne garantissent pas la qualité des solutions achevées, et même d'un essai à l'autre les solutions peuvent changer en fonction de stabilité de l'algorithme utilisé. La différence majeure entre les métaheuristiques et les autres méthodes approximatives réside dans le fait que les premiers ne fournissent pas une information sur quel point les solutions obtenues sont proches à celles optimales (Talbi 2009).

### 2.5.2 Principe de fonctionnement des métaheuristiques

Durant le fonctionnement des métaheuristiques, deux opérations se déroulent d'une façon mélangée, mais différentielle d'une méthode à l'autre : *l'exploration (ou diversification)* et *l'exploitation (ou intensification)*. Les deux opérations sont contradictoires (figure 2.3) et doivent être prises en compte dans la conception des telles méthodes (Talbi 2009) :

1. L'exploration de l'espace de recherche, dans laquelle l'algorithme doit visiter toutes les régions de l'espace d'une façon égale et ne se limite pas à un nombre réduit de celles-ci. Cette phase se produit par la diversification des solutions.
2. L'exploitation des régions prometteuses caractérisées par les "bonnes" solutions (déterminées par l'opération précédente). Ces régions sont explorées encore d'une manière intensive dans l'espérance de trouver les "meilleures" solutions.

La classe des algorithmes basées sur populations ("*Population-based metaheuristics*") (figure 2.3) manipulent un ensemble de solutions au cours d'un processus itératif, où ils essaient de les améliorer d'une itération (ou génération) à l'autre. Les méthodes appartenant à cette classe ont une tendance vers la diversification plus qu'à l'intensification. La méthode de recherche aléatoire ("*Random Search*") est une méthode à diversification pure. D'une génération à l'autre, elle génère un tas de solutions aléatoirement dans l'espérance de trouver une meilleure, sans garder une mémoire sur les générations précédentes.

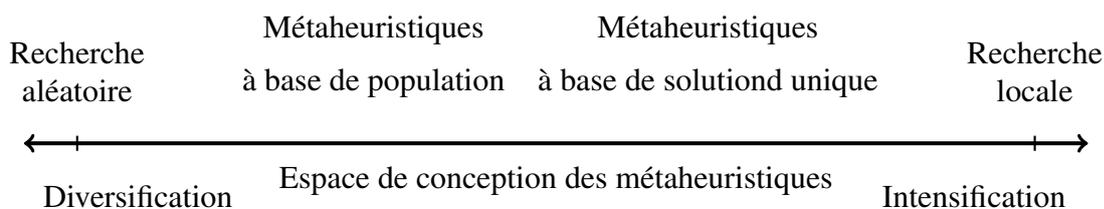


FIGURE 2.3 – Les deux critères conflictuels (la diversification et l'intensification) des algorithmes métaheuristiques. Adapté d'après (Talbi 2009)

Au contraire, la classe des méthodes basées sur une solution unique se caractérisent par une tendance à l'intensification plus qu'à la diversification. La méthode de recherche locale "*Local Search*" essaie d'améliorer une solution unique d'une itération à l'autre en explorant les solutions voisines (Talbi 2002). Malgré la contradiction entre les deux opérations (exploration et exploitation), le balancement entre elles détermine la performance globale d'une métaheuristique.

### 2.5.3 Domaines d'application

Les algorithmes métaheuristiques ont été développés principalement pour la résolution des problèmes d'optimisation complexes et l'apprentissage des modèles d'intelligence artificielle (notamment les réseaux de neurones artificiels et machines à vecteurs de support "SVM") (Yao 1993). Au cours des dernières 20 années, ils ont gagné beaucoup d'intérêt dans le domaine de l'intelligence artificielle (Janga Reddy and Nagesh Kumar 2020) et ont trouvé des applications partout dans les différents soucis d'optimisation, y compris la conception industrielle, l'informatique, l'électronique, l'aérodynamique, la télécommunication, robotique, apprentissage automatique "Machine learning", le contrôle, traitement de signal, logistique, finance, gestion des flux, ... etc., (Talbi 2009).

Le domaine de l'ingénierie des ressources en eau n'a pas échappé. Les premières applications de tels algorithmes ont été proposées par (McKinney and Lin 1994; Ritzel et al. 1994) dans le traitement des problèmes de gestion des eaux souterraines en utilisant les algorithmes génétiques. Par la suite, l'emploi des métaheuristiques a touché pratiquement l'ensemble des filières du domaine, y compris les systèmes de distribution d'eau potable, de drainage urbain et d'assainissement, la gestion opérationnelle des réservoirs et systèmes d'irrigation, la gestion des bassins versants, les systèmes fluviaux et des eaux souterraines, la conception des réseaux de surveillance des systèmes d'eau souterraine, ... etc., (Yang et al. 2012; Mala-Jetmarova et al. 2017); Ayad et al. 2018; Janga Reddy and Nagesh Kumar 2020).

### 2.5.4 Classification des métaheuristiques

La classification des algorithmes métaheuristiques peut être effectuée selon plusieurs critères et points de vue. Les plus conventionnels sont (Talbi 2002) :

**§1 Naturels ou physiques** : Un grand nombre de tels algorithmes sont inspirés depuis des phénomènes et processus biologiques, notamment l'évolution naturelle (algorithmes génétiques "Genetic Algorithms (GAs)", l'évolution différentielle "Differential Evolution DE)", ... etc), et l'imitation des comportements animales (algorithme de colonies d'abeilles "Artificial Bee Colony (ABC)", de fourmis "Ant Colony Optimization (ACO)", loups gris "Grey Wolf Optimizer (GWO)", IGWO, HPSOGWO, ... etc). Cependant, les autres sont inspirés depuis les phénomènes physiques tels que la gravitation (GSA), vague d'eau "Water Wave Algorithm (WWA)", ... etc).

**§2 Algorithmes basés sur une solution unique ou une population de solutions** : Les premiers manipulent une seule solution durant le processus de recherche (e.g., recherche locale "LS", Recherche Tabu "Tabu Search (TS)), tandis que les deuxièmes transforment un ensemble (population) de solutions (e.g., GAs, ABC, ACO, GSA, WWA, PSO, ... etc). Les deux familles présentent des caractères contradictoires mais complémentaires, l'intensification pour les premiers et la diversification pour les deuxièmes. En effet, l'hybridation entre un algorithme à base d'une solution unique et un autre à base de population génère un algorithme fils plus performant que les pères dans la majorité des cas (Talbi 2002). Cette classification est la plus adoptée dans la littérature (Talbi 2009).

Tous les algorithmes utilisés dans cette étude sont basés sur une population de solution "population-based algorithms". Cependant, cette classe est peut-être subdivisée en deux sous classes (Janga Reddy and Nagesh Kumar 2020) :

1. Algorithmes évolutionnaires (e.g., GA, RGA, DE, stratégies évolutionnaires "Evolutionary Strategies "ES", ... etc);
2. Algorithmes d'essaims intelligents (e.g., ABC, ACO, GSA, GWO, IGWO, HPSOGWO, ... etc).

Selon (Talbi 2009), la nomination "*Algorithmes Évolutionnaires d'Optimisation*", en anglais "*Evolutionary Optimization Algorithms (EOAs)*" est souvent utilisée pour désigner les deux sous-classes sans différenciation entre elles. Car, on peut avoir deux algorithmes identiques (effectuent les mêmes opérations durant le processus d'optimisation), mais appartenant aux sous-classes différentes. En effet, la nomination "Algorithmes Évolutionnaires" est adoptée dans la suite de ce travail pour l'ensemble des algorithmes étudiés.

**§3 Algorithmes déterministes ou stochastiques** : Les premiers utilisent des mécanismes de recherche bien définis (pas de caractère aléatoire) d'une itération à l'autre (e.g., méthodes LS, TS, ...etc). En effet, si l'algorithme commence par la même solution initiale, il aboutit à la même solution finale. Par contre, les algorithmes stochastiques utilisent certaines procédures internes de recherche aléatoire. Par conséquent, des solutions finales différentes peuvent être trouvées en commençant par les mêmes solutions initiales. Les algorithmes à base de population sont globalement des méthodes stochastiques (e.g., GAs, ABC, ACO, WWA, PSO, ... etc), y compris les algorithmes utilisés dans cette étude (RGA, GSA, IGWO et HPSOGWO).

D'autres critères de classification moins adoptés dans la bibliographie peuvent être trouvés dans (Talbi 2009).

Comme évoqué précédemment, les métaheuristiques à base de population sont des

méthodes stochastiques à cause de leur caractère de recherche aléatoire intrinsèque. Cependant, ils sont utilisés dans les modèles d'optimisation déterministes, c-à-d., lorsque toutes les variables à l'entrée d'un modèle sont exprimées explicitement.

### 2.5.5 Pourquoi l'usage des métaheuristiques ?

Chaque année, des méthodes d'optimisation nouvelles ou améliorées s'immergent dans la bibliographie. Cependant, chacune présente des avantages, inconvénients, et même des limitations. Le tableau 2.1 résume à titre d'exemple les principaux inconvénients des méthodes classiques utilisées dans le domaine de la gestion opérationnelle des réservoirs. Ceux-ci continuent à pousser les chercheurs à développer des techniques plus performantes ou les améliorer dans le but de surmonter les défaillances des précédentes.

TABLE 2.1 – Inconvénients majeurs des méthodes classiques principales

Méthode	Principaux inconvénients / incapacités
LP*	Fonctions objectifs et contraintes non linéaires ou discontinues
NLP*	Fonctions objectifs et contraintes de non-linéarité élevée ou discontinues, problèmes à grandes échelles (dimensionnalité élevée)
DP*	Fléau de dimensionnalité, nécessité de discrétisation, explosion de nombre d'opérations mathématiques et temps du calcul

\* Programmation Linéaire (LP), Programmation Non-Linéaire (NLP), Programmation Dynamique (DP).

Le jugement de choix d'une méthode ou famille de méthodes d'optimisation dans le cas de l'optimisation de la gestion opérationnelle d'un réservoir est basé sur les facteurs qui l'influencent (cités dans la section 2.4 du présent chapitre), à savoir :

**§1 La nature du problème et son complexité** : Le problème d'optimisation du fonctionnement d'un réservoir est complexe à cause de plusieurs raisons :

- Il peut être mono-objectif comme multi-objectif. La considération de cette dernière catégorie sans simplification nécessitent des méthodes d'optimisation plus complexes que la première ;
- L'optimisation de ce type de problèmes n'est pas libre, elle subit aux différentes contraintes exigées par le fonctionnement physique du réservoir ;
- Les fonctions objectifs à minimiser ou à maximiser sont souvent non linéaires et multimodales (ne possèdent pas un seul optimum facile à l'atteindre) ;

- Le problème est souvent de haute dimensionnalité. Dans le cas d'une optimisation déterministe sur une durée de fonctionnement de  $T = 100$  mois par exemple, le nombre de variables à déterminer est égale à 100 lâchers mensuels ;
- Mathématiquement, la complexité d'un problème est relative à celle de l'algorithme de traitement utilisé et à combien de fois un tel algorithme est appelé pour achever la résolution (Talbi 2002). Dans les approches d'optimisation ISO et P-S-O (présentées dans le chapitre précédent, section 1.4), un algorithme d'optimisation doit être utilisé  $N$  fois pour  $N$  scénarios d'apports probables, sans oublier le nombre d'opérations mathématiques internes à effectuer par l'algorithme lui-même. En effet, le "coût" d'un algorithme en nombre d'opérations, temps de calcul et consommation de mémoire de l'ordinateur, impacte considérablement la performance globale d'un modèle d'optimisation.

Ces caractéristiques de complexité du problème figurent parmi les inconvénients de la majorité des méthodes classiques d'optimisation. Cependant, elles peuvent être franchies par les méthodes évolutionnaires (Rani and Moreira 2010).

**§2 La nature de l'algorithme d'optimisation** : En plus les différences entre les méthodes classiques et les métaheuristiques déjà citées, ces dernières possèdent une grande capacité d'intégration avec différents problèmes à cause de leur caractère générique. La même métaheuristique est utilisable d'un problème à l'autre sans nécessité d'adaptation de sa structure ou de réimplémentation (reprogrammation) de l'algorithme. Dans ce cas, uniquement la (ou les) fonction(s) objectif et les contraintes doivent être reformulées avec l'ajustement de paramétrage de l'algorithme. Dans ce qui suit de ce travail, les mêmes métaheuristiques sont utilisées pour effectuer l'apprentissage des réseaux de neurones artificiels "ANNs" (chapitres 3 et 7), la recherche des réseaux de neurones artificiels évolutionnaires "EANNs" (chapitres 4 et 8) et l'optimisation de fonctionnement du réservoir Hammam Boughrara (chapitre 9).

**§3 La qualité de solutions exigée** : Sous l'effet des incertitudes caractérisant la gestion opérationnelle des réservoirs, notamment celles relatives à la quantification des différents phénomènes hydrologiques en amont (précipitations, apports, évaporation, infiltration, transport solide, ... etc) en premier lieu, à l'estimation fiable des besoins en eau en aval (variation temporelle des demandes en eau) en deuxième lieu, et à l'évaluation des impacts socio-économiques d'une politique de gestion donnée en troisième lieu, les "bonnes" décisions (solutions approchées) peuvent être largement suffisantes. Par conséquent, les algorithmes métaheuristiques évolutionnaires peuvent constituer un choix satisfaisant.

**§4 Temps et ressource de calculs** : Les différentes approches d'optimisation stochastique de fonctionnement d'un réservoir (section 1.4), exigent des calculs intensifs pour l'aboutissement à des règles de gestion optimales. Les approches ISO et P-S-O réclament la génération d'un nombre important de politiques de gestion pour déduire les courbes règles. Cette opération nécessite elle-même la génération d'un nombre conséquent de scénarios d'apports (ou même de consommation en eau), puis l'optimisation de fonctionnement du réservoir en fonction de ceux-ci. Cette dernière tâche impose l'appel d'un algorithme d'optimisation autant de fois que le nombre de scénarios. Si l'algorithme d'optimisation employé est lent et/ou gourmand aux ressources de l'ordinateur (mémoire et traitement "processing"), l'opération entière pourra être longue (requiert plusieurs heures ou jours de calculs) ou non envisageable par fois. En effet, le recours aux métaheuristiques qui sont plus rapides par rapports aux méthodes classiques, participe considérablement à l'économie du temps et des ressources du traitement d'une part et permet la reproduction des calculs si nécessaire.

Vu les inconvénients des méthodes classiques et en considérant les facteurs influençant le choix d'une méthode d'optimisation sur le cas de l'optimisation du fonctionnement des réservoirs multi-usages, les métaheuristiques peuvent constituer un choix judicieux.

### 2.5.6 Inconvénients des métaheuristiques

Les métaheuristiques sont développées dans l'espérance de franchir les limitations des anciennes méthodes. Malgré leurs avantages, elles présentent plusieurs inconvénients. (Karami et al. 2018) résume ces derniers dans les points suivants :

1. Une grande partie de ces algorithmes peuvent rater l'optimal global et être piégés dans un optimal local ;
2. La majorité de ces méthodes ne peuvent pas détecter la nature stochastique de certaines variables (p.ex., les apports, les demandes, ... etc) ;
3. Certaines méthodes connaissent un déséquilibre entre l'exploitation et l'exploration à différents niveaux de recherche ;
4. Les métaheuristiques sont sensibles à leur paramétrage. À titre d'exemple, l'étude de sensibilité présentée dans notre papier (Dahmani and Yebdri 2020) confirme cet inconvénient concernant les algorithmes RGA, GSA et HPSOGWO ;
5. Et finalement, si un algorithme prouve sa performance dans un domaine, ça ne généralise pas sa performance sur d'autres domaines. Ce désavantage est connu sous "*les théorèmes de pas de déjeuner gratuit pour l'optimisation, (No free lunch theorems for optimization)*" introduit par (Wolpert and Macready 1997).

Dans ce contexte, il est nécessaire d'étudier et explorer des nouvelles méthodes qui

pourraient surmonter ces inconvénients et permettre d'obtenir des politiques et règles de gestion optimales des barrages-réservoirs et systèmes de réservoirs.

## **2.6 Les métaheuristiques utilisées dans le cadre de l'étude**

### **2.6.1 Algorithme génétique à codage réel (RGA)**

Les algorithmes génétiques (GAs) constituent une sous-classe des méthodes d'optimisation évolutionnaires (EOAs). Ils combinent des stratégies de la survie des plus forts avec l'échange de l'information d'une manière aléatoire mais orientée (recherche pseudo-aléatoire guidée). Les solutions sont représentées sous forme de génomes (chromosomes constitués d'un ensemble de gènes) codés soit sous forme de vecteurs en valeurs binaires, réelles ou chaînes de caractères selon le cas.

Les origines des algorithmes génétiques remontent au milieu des années 1950 avec les travaux de Barricelli (Barricelli 1957; Barricelli 1963) et Fraser (Fraser 1957). Dans ces travaux, les deux auteurs ont commencé l'utilisation de la simulation assistée par ordinateur pour mieux comprendre les processus naturels de l'héritage biologique étudiés par J.G Mendel (1822 - 1884) au 19<sup>ème</sup> siècle et la théorie de l'évolution génétique proposée par Charles Darwin (1809-1882) et publiée dans son livre « On the origin of species by means of natural selection ». (Bledsoe 1962; Bremermann et al. 1962) ont utilisé des approches évolutives basées sur des génomes binaires (chaînes de chiffres binaires) pour résoudre les inégalités, l'optimisation des fonctions, et la détermination des poids dans les réseaux neuronaux au début des années 1960.

La véritable formulation de l'algorithme génétique utilisée actuellement est introduite par John Holland au milieu des années 1970 (Holland 1975), puis par Goldberg et Holland (Goldberg and Holland 1988). Des améliorations successives ont été effectuées sur la formulation initiale de l'algorithme génétique en créant des variantes de ce dernier. Ce qui a eu comme résultat une famille d'algorithmes appelée « Algorithmes Génétiques ». On cite par exemple, le GA à codage binaire (binary-coded GA), GA à codage réel (Real-coded GA "RGA"), GA à tri non dominant (non-dominating sorting GA version I "NSGA-I", NSGA-II, NSGA-III, ... etc.

(Wright 1991) a introduit l'algorithme génétique à codage réel (RGA), qui est une autre méthode de codage des individus (représentation de solutions) en utilisant des valeurs réelles au-lieu binaires. Le RGA possède beaucoup d'avantages que son homologue à codage binaire, notamment lorsqu'il s'agit de rechercher dans des espaces continus avec de grandes dimensions, et lorsque une grande précision numérique est requise (Mishra and Patnaik 2009). Aussi dans le RGA, chaque gène (élément d'un génome "chromosome") représente une variable

du problème, et la taille du génome est maintenue la même que la longueur de la solution. (Wright 1991 ; Janikow and Michalewicz 1991 ; Chang and Chen 1998) ont comparé les deux versions binaire et réelle et ont conclu que cette dernière (RGA) produit des résultats meilleurs que la première.

Les algorithmes génétiques sont assez robustes et peuvent être appliqués à grande variété de problèmes d'optimisation sans considérer la forme de la fonction objectif comme contrainte. Cependant, pour avoir un fonctionnement optimal des algorithmes génétiques, certains paramètres doivent être réglés soigneusement et rigoureusement, notamment la taille de la population, le nombre de générations (d'itérations), les probabilités de croisement et de mutation, ... etc. Comme autres algorithmes évolutionnaires, les algorithmes génétiques sont sensibles aux valeurs de paramètres, ce qui représente un inconvénient.

Depuis les premières utilisations des algorithmes génétiques dans le domaine de l'ingénierie des ressources en eau (McKinney and Lin 1994 ; Ritzel et al. 1994), ils ont été adoptés par plusieurs auteurs dans ce domaine (Yang et al. 2012). Cette famille d'algorithmes a prouvé une performance similaire ou supérieure aux anciennes méthodes dans branche de la gestion opérationnelle des réservoirs et systèmes de réservoirs, telles que la programmation linéaire "LP" (Nagesh Kumar et al. 2006 ; Devisree and Nowshaja 2014), la programmation dynamique stochastique "SDP" (Ahmed and Sarma 2005 ; Jothiprakash and Shanthi 2009). Actuellement, les algorithmes génétiques sont les métaheuristiques les plus populaires utilisées dans la gestion des ressources en eau (Yang et al. 2012).

Dans ce travail, une version mono-objectif simple de l'algorithme génétique à codage réel (RGA) est utilisée dans le but d'établir une étude comparative avec les autres algorithmes GSA, IGWO et HPSOGWO. La section suivante est consacrée à la présentation de la version utilisée.

### **2.6.1.1 Principe et algorithme**

Pour un problème d'optimisation donné, un ensemble de solutions possibles nommées "individus, génomes, ou encore chromosomes" se génère couramment d'une manière pseudo-aléatoire au départ. Cet ensemble est appelé "population". D'une époque (génération ou itération) à l'autre, la population évolue en formant des générations de solutions en passant par trois étapes : sélection, croisement "crossover", et mutation. Les mécanismes de réalisation de ces dernières diffèrent d'une version GA à l'autre.

Durant la première étape (la sélection), les individus se classent par rapport à leur performance "fitness", laquelle est évaluée au moyen de la fonction objectif. Si l'objectif est

la minimisation d'une fonction, la solution donnant la faible valeur de celle-ci s'attribue par la meilleure performance. Par autre manière, les solutions se classent de "forte" au "faible" par ordre croissant de la valeur correspondante de la fonction objectif. Si l'optimisation s'effectue pour maximisation, l'ordre de classement s'inverse.

Les individus présentant les meilleures performances sont sélectionnés pour la reproduction. Cette dernière s'effectue par le croisement entre chaque couple de solutions choisies arbitrairement parmi les meilleures. Puis, une mutation aléatoire s'effectue sur certains individus dans le but de diversifier encore les solutions. Les individus de faibles performances s'éliminent, et une génération plus performante que la précédente se formule. Ce processus se répète jusqu'à l'atteint de certain critère d'arrêt (figure 2.4).

La version RGA utilisée dans ce travail s'exécute selon le processus suivant :

1. Initialisation de la population de recherche :

Une population de solutions (individus) possibles est initialisée, de sorte que les valeurs (gènes) sont générées d'une façon pseudo-aléatoire dans les plages de recherche faisables. La taille de la population est assumée égale à  $N$ .

$$X_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^{D_s}) \quad i = 1, 2, \dots, N \quad (2.6)$$

Tels que :

- $X_i$  : individu (ou génome ou encore chromosome) d'indice  $i$ , représente une solution possible du problème d'optimisation ;
- $x_i^d$  : gène (valeur numérique réelle le cas du RGA) d'indice  $d$  appartenant à l'individu d'indice  $i$  ;
- $D_s$  : dimension de l'espace de recherche "search space dimension", égale au nombre de variables de décision.

Dans le cas du problème d'optimisation de la gestion opérationnelle d'un réservoir, dont les lâchers ( $\vec{R} = \{R_t\}$ ,  $t = 1, 2, \dots, T$ ) correspondent aux variables de décision sur une durée du temps  $T$ . L'équation 2.6 peut être réexprimée comme suit, où la dimension de l'espace de recherche  $D_s = T$  :

$$X_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^{D_s}) = \vec{R}_i = (R_i^1, R_i^2, \dots, R_i^t, \dots, R_i^T) \quad (2.7)$$

Les valeurs  $R_i^t$  doivent être faisables, c-à-d.,  $R_{min} \leq R_i^t \leq R_{max}$ .

2. Évaluation des individus :

Chaque individu est passé comme argument à la fonction objectif (à minimiser ou à maximiser) pour évaluer sa performance "fitness" (équation 2.8). Par la suite, la population est triée par ordre croissant (cas de minimisation) ou décroissant (cas de maximisation) à la base de la performance de chaque individu.

$$fit_i^k(X_i) = F_{obj}(X_i) \quad (2.8)$$

Tels que :

- $fit_i^k(X_i)$  : la performance de l'individu  $X_i$  à l'itération  $k$  ;
- $F_{obj}$  : la fonction objectif.

3. Croisement des individus :

Dans cette version RGA, la technique de croisement d'un seul point aléatoire ("random single point crossover") est utilisé. Elle consiste à générer arbitrairement un indice  $l$  (tel que  $0 < l < D_s$ ), puis échanger les gènes entre deux individus "Parents" aléatoirement sélectionnés de l'indice  $l$  jusqu'à  $D_s$  comme suit :

$$Parent\ 1 : X_1 = (x_1^1, \dots, x_1^l, \dots, x_1^{D_s}), \quad Parent\ 2 : X_2 = (x_2^1, \dots, x_2^l, \dots, x_2^{D_s}) \quad (2.9)$$

Après croisement on obtient :

$$fils\ 1 : X_1' = (x_1^1, \dots, x_2^l, \dots, x_1^{D_s}), \quad fils\ 2 : X_2' = (x_2^1, \dots, x_1^l, \dots, x_2^{D_s}) \quad (2.10)$$

4. Mutation aléatoire des individus :

La technique de mutation aléatoire à un seul point ("random single-point mutation") est utilisée. Elle consiste à générer aléatoirement un indice  $m$  (tel que  $0 < m < D_s$ ), puis sélectionner arbitrairement (par une probabilité faible) les individus à muter, dans lesquels le gène  $x_i^m$  d'indice  $m$  est remplacé par un autre  $y_i^m$  ( $y_i^m$  est une valeur réelle faisable le cas du RGA) (équation 2.11) :

$$X_i = (x_i^1, \dots, x_i^m, \dots, x_i^{D_s}), \quad mutation : X_i = (x_i^1, \dots, y_i^m, \dots, x_i^{D_s}) \quad (2.11)$$

5. Sélection des individus pour la génération suivante :

À la fin de l'étape précédente (mutation), il résulte un ensemble de taille  $2 \times N$  d'individus (parents et fils). L'évaluation de performance de chacun, puis le tri de l'ensemble permettent de sélectionner les meilleurs individus pour la génération suivante.

6. Les étapes de 2 à 5 sont répétées jusqu'à atteindre une condition d'arrêt prédéfinie. Cette dernière peut être un nombre maximal de générations (itérations)  $K_{max}$ , ou autre telle que la stabilité de la meilleure performance pendant un nombre d'itérations donné, ... etc.

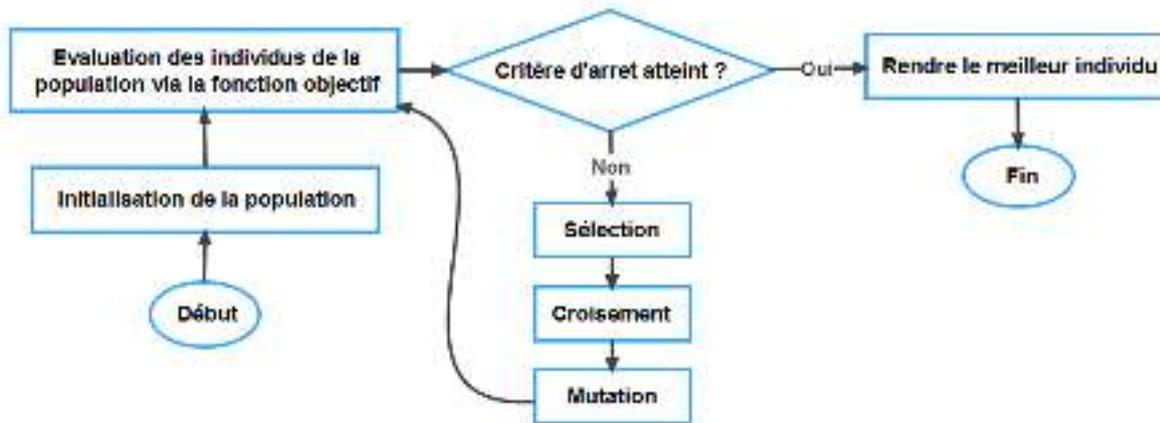


FIGURE 2.4 – Organigramme explicatif du fonctionnement de l'algorithme génétique à codage réel (RGA)

À la fin du processus, le meilleur individu est considéré comme la meilleure solution du problème et sa valeur de la fonction objectif comme la meilleure performance "the best fitness".

### 2.6.2 Algorithme de recherche gravitationnelle (GSA)

GSA (Gravitational Search Algorithm) est l'un des nouveaux algorithmes metaheuristiques basés sur la population de solutions dans leurs processus. Depuis son développement par (Rashedi et al. 2009), il a été utilisé dans plusieurs domaines, y compris l'ingénierie de l'énergie, la télécommunication, le génie mécanique, le traitement d'image, la classification, le génie civil, ... etc. (Sabri et al. 2013) a présenté une révision sur le GSA et les domaines dans lesquels il a été appliqué.

Les auteurs de l'algorithme (Rashedi et al. 2009) ont introduit des tests de comparaison avec une version RGA, PSO et l'algorithme d'optimisation de force centrale "central force optimisation (CFO)". Les tests ont été effectués sur des fonctions standards de référence "benchmark functions". Les résultats ont montré la supériorité du GSA dans la plupart des cas. Dans l'optimisation de la gestion des réservoirs et systèmes de réservoirs, (Bozorg-Haddad et al. 2016) a montré la supériorité du GSA contre l'algorithme génétique (une version RGA), l'algorithme de chauve-souris "Bat Algorithm (BA)", l'algorithme de cycle de l'eau "Water Cycle Algorithm (WCA)" et l'optimisation basée sur la biogéographie "biogeography-based optimisation (BBO)".

Au contraire aux algorithmes GAs et PSO, le GSA n'a pas été exploité intensivement dans le domaine de l'ingénierie des ressources en eau. Le manuscrit (Ahmad et al. 2014) a

revendiqué plus d'exploration de la capacité de cet algorithme, ce qui nous a encouragé à sa prise en considération dans cette étude.

### 2.6.2.1 Principe et algorithme

La population en GSA est un ensemble d'agents de recherche "Searcher agents" considérés comme des masses qui interagissent les uns avec les autres par les lois de la gravité et du mouvement de Newton (Rashedi et al. (2009)). La loi de gravité (Équation 2.12, figure 2.5) indique que chaque objet attire un autre par une force gravitationnelle  $F$  proportionnelle au produit de leurs masses  $M_1, M_2$  et constante gravitationnelle  $G$  et inversement proportionnelle au carré de distance entre eux  $R_G$ . La loi de mouvement (Équation 2.13) dénote que l'accélération  $a$  d'un objet dépend seulement de la force appliquée  $F$  et de la masse d'objet  $M$ .

$$F = G \frac{M_1 M_2}{R^2} \quad (2.12)$$

$$a = \frac{F}{M} \quad (2.13)$$

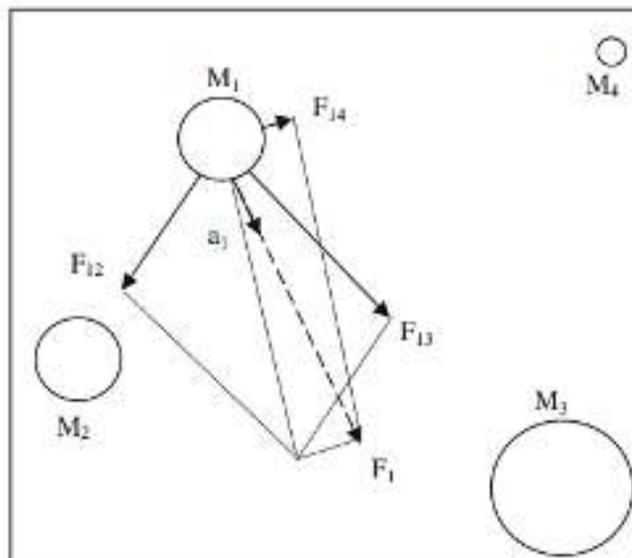


FIGURE 2.5 – Principe du GSA. Chaque masse accélère vers la force résultant de l'agissement des autres masses (Rashedi et al. 2009)

Dans le GSA, les masses d'objets (solutions) sont proportionnelles à leurs valeurs de la fonction objectif (coûts). Les masses les plus lourdes ont les forces les plus attractives. Par conséquent, ces masses qui sont probablement proches de l'optimum global, attirent les autres masses en fonction de leurs distances.

L'algorithme de recherche gravitationnelle est exécuté en suivant les étapes ci-après :

1. Initialisation de la population :

Le processus du GSA commence par l'initialisation d'un ensemble de solutions "population" en considérant un système de  $N$  agents  $X_i$  :

$$X_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^{D_s}) \quad i = 1, 2, \dots, N \quad (2.14)$$

La solution  $X_i$  est défini dans l'espace de recherche avec une dimension  $D_s$ . La  $i^{\text{ème}}$  position  $x_i^d$  de l'agent  $X_i$  dans  $d^{\text{ème}}$  dimension est définie aléatoirement avec une valeur faisable. Ensuite, la valeur correspondante de la fonction objectif est calculée pour chaque agent (ou masse).

2. Calcul de la meilleure solution de l'itération courante :

La performance "fitness" de chaque agent est calculée au moyen de la fonction objectif. Pour les problèmes de minimisation ou de maximisation, la meilleure et la pire valeur de performance à chaque itération  $k$  sont exécutées comme suit :

— Pour les problèmes de maximisation :

$$Best(k) = Max(fit_i(k)) \quad i \in 1, 2, \dots, N \quad (2.15)$$

$$Worst(k) = Min(fit_i(k)) \quad i \in 1, 2, \dots, N \quad (2.16)$$

— Pour les problèmes de minimisation :

$$Best(k) = Min(fit_i(k)) \quad i \in 1, 2, \dots, N \quad (2.17)$$

$$Worst(k) = Max(fit_i(k)) \quad i \in 1, 2, \dots, N \quad (2.18)$$

Tels que :

- $fit_i(k)$  : la valeur de performance de l'agent  $i$  à l'itération  $k$  ;
- $Best(k)$  : la meilleure performance à l'itération  $k$  ;
- $Worst(k)$  : la pire performance à l'itération  $k$ .

3. Calcul des masses des agents :

À chaque itération  $k$ , les masses gravitationnelles et d'inertie sont calculées pour chaque agent.

$$m_i(k) = \frac{fit_i(k) - Worst(k)}{Best(k) - Worst(k)} \quad (2.19)$$

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^N m_j(k)} \quad (2.20)$$

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad (2.21)$$

Avec :

- $m_{ai}$  : la masse gravitationnelle active de l'agent d'indice  $i$  ;
- $M_{pi}$  : la masse gravitationnelle passive l'agent  $i$  ;
- $M_{ii}$  : indique la masse d'inertie de l'agent  $i$ .

4. Calcul de la constante gravitationnelle  $G$  :

À chaque itération  $k$ , la constante gravitationnelle  $G$  est mise à jour en utilisant la formule suivante (Équation 9.2) (Rashedi et al. 2009).  $G$  sera réduit avec l'itération  $k$ .

$$G(k) = G_0 e^{\left(\frac{-\alpha_g k}{K_{max}}\right)} \quad (2.22)$$

Tels que :

- $G_0$  et  $\alpha_g$  : sont des constantes initialisées au début des itérations. Ils correspondent aux paramètres intrinsèques de l'algorithme GSA ;
- $K_{max}$  : le nombre d'itérations.

5. Calcul des accélérations des agents :

À chaque itération  $k$ , l'accélération de l'agent  $i$  est calculée. Premièrement, la force totale  $F_i^d(k)$  agissant sur l'agent  $i$  est déterminée au moyen de l'équation suivante.

$$F_i^d(k) = \sum_{j \in K_{best}, j \neq i} rand_j \cdot F_{ij}(k) \quad (2.23)$$

Tels que :

- $rand_j$  : un nombre aléatoire dans l'intervalle  $[0, 1]$  ;
- $K_{best}$  : ensemble des premiers agents  $K$  avec les plus grandes masses et les meilleures valeurs de performance.  $K_{best}$  diminuera linéairement, et à la fin il n'y aura qu'un agent appliquant la force sur les autres.

$F_{ij}^d(k)$  est la force agissant par l'agent  $j$  sur l'agent  $i$  à l'itération  $k$  dans la dimension  $d$ . Elle est donnée par l'équation suivante :

$$F_{ij}^d(k) = G(k) \frac{M_{pi}(k) \cdot M_{aj}(k)}{R_{ij}(k) + \epsilon_g} (x_j^d(k) - x_i^d(k)) \quad (2.24)$$

$R_{Gij}(k)$  est la distance euclidienne entre deux agents  $i$  et  $j$  à l'itération  $k$  ;  $G(k)$  est la constante gravitationnelle calculée à l'étape précédente en utilisant l'équation (9.2), tandis que  $\epsilon$  est une petite constante mise pour éviter la division sur zéro.

Enfin, l'accélération  $a_i^d$  de l'agent  $i$  dans la dimension  $d$  à l'itération  $k$  est calculée.

$$a_i^d(k) = \frac{F_i^d(k)}{M_{ii}(k)} \quad (2.25)$$

6. Mise à jour de la vitesse et positions des agents :

Pour chaque agent  $i$ , la vitesse et la position à l'itération suivante ( $k + 1$ ) dans la dimension  $d$  sont calculées en fonction de la vitesse, la position et de l'accélération de l'itération courante comme suit :

$$v_i^l(k + 1) = rand_i \cdot v_i^l(k) + a_i^l(k) \quad (2.26)$$

$$x_i^l(k + 1) = x_i^l(k) + v_i^l(k + 1) \quad (2.27)$$

Avec :

—  $Rand_i$  : valeur aléatoire dans l'intervalle  $[0,1]$ . Ce nombre aléatoire est utilisé pour donner une caractéristique aléatoire à la recherche.

Les étapes 2 à 6 sont répétées jusqu'à ce que les itérations atteignent leur limite maximale  $K_{max}$ . La meilleure valeur de performance à l'itération finale représente la performance globale, tandis que la position de l'agent correspondant représente la solution globale du problème.

La figure 2.6 illustre l'organigramme général de l'algorithme de recherche gravitationnelle (GSA).

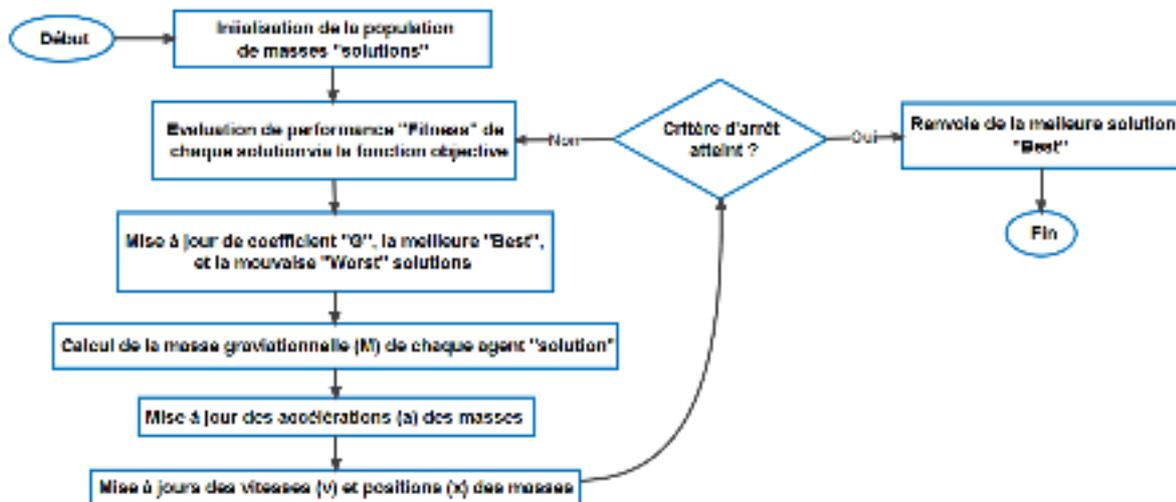


FIGURE 2.6 – Organigramme général de processus d'optimisation du GSA.

### **2.6.3 Algorithme d'optimiseur de Loup Gris amélioré (IGWO)**

L'optimiseur de Loup Gris amélioré "Improved Grey Wolf Optimizer (IGWO)" est un nouvel algorithme évolutionnaire proposé par (Long et al. 2017). Comme son nom indique, l'IGWO constitue une amélioration de l'algorithme GWO "Grey Wolf Optimizer" développé par (Mirjalili et al. 2014). Les algorithmes GWO et IGWO (ainsi le HPSOGWO, section 2.6.4) imitent le comportement de chasse des Loups Gris (canis-lupus) dans la nature et partagent la quasi-totalité du processus d'optimisation.

Depuis la publication du GWO, il a été utilisé avec succès dans divers domaines, tels que l'apprentissage automatique, le traitement d'image, l'entraînement des réseaux de neurones artificiels, la planification, ... etc., (Faris et al. 2018). L'algorithme GWO présente plusieurs avantages qui ont encouragé son adoption par les chercheurs, à savoir (Singh and Singh 2017; Faris et al. 2018) :

1. Structure simple et facilité d'implémentation ;
2. Peu de paramètres à ajuster ;
3. Nombre réduit d'opérations mathématiques, donc une rapidité d'exécution.

En revanche, comme tout algorithme d'optimisation le GWO a présenté certains désavantages comme le déséquilibre entre l'exploration et l'exploitation pour certains problèmes. En effet, plusieurs recherches ont été faites pour améliorer sa performance (exploration et exploitation) et qui ont surgi d'autres versions, telles que le GWO modifié "mGWO" (Mittal et al. 2016), IGWO, HPSOGWO (Singh and Singh 2017), ... etc.

À la limite de connaissance de l'auteur de la thèse, les algorithmes GWO, IGWO et HPSOGWO n'ont pas été utilisés et testés dans le domaine de l'ingénierie des ressources en eau auparavant, et vu les avantages caractérisant ces méthodes, nous étions encouragés à inclure l'IGWO et le HPSOGWO dans cette présente étude et ce dernier dans notre papier (Dahmani and Yebdri 2020).

#### **2.6.3.1 Principe et algorithme**

Comme évoqué précédemment, les algorithmes GWO, IGWO et HPSOGWO sont basés sur les essaims intelligents imitant le comportement de chasse des Loups Gris. Ces derniers vivent en groupes de 5 à 12 membres avec une hiérarchie stricte, comme le montre la figure 2.7. Le membre classé en tête, appelé Alpha ( $\alpha$ ), est considéré comme le dominant d'un groupe. Il est chargé de prendre des décisions sur le comportement social du groupe en particulier la

chasse. Les membres Beta ( $\beta$ ) et Delta ( $\delta$ ) sont subordonnés à Alpha, mais ils peuvent contrôler le reste du groupe qui s'appelle Omega ( $\omega$ ).

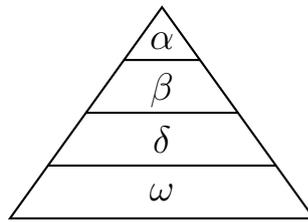


FIGURE 2.7 – La hiérarchie sociale des groupes de Loups Gris (Mirjalili et al. 2014)

Le mécanisme de chasse des loups gris comporte des étapes spécifiques (Muro et al. 2011) : d'abord, suivre, chasser et approcher la proie ; deuxièmement, poursuivre, encercler et harceler la proie jusqu'à ce qu'elle cesse de bouger ; Troisièmement, attaquer vers la proie.

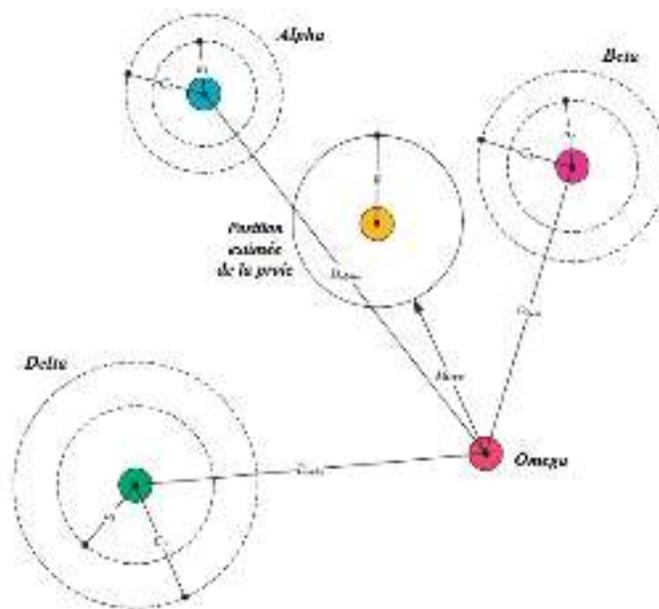


FIGURE 2.8 – Mise à jour des positions des éléments de recherche dans le GWO et IGWO. Adapté d'après (Mirjalili et al. 2014)

Dans le modèle mathématique, les étapes de chasse relatives aux GWO et IGWO sont imitées d'une manière identique sauf le mécanisme de mise à jour du paramètre intrinsèque ( $\vec{a}$ , équations 2.30 et 2.31) :

1. Premièrement,  $N$  Loups Gris (agents de recherche "ou solutions") sont définis et placés au hasard (d'une façon pseudo-aléatoire) dans l'espace de recherche comme suit :

$$X_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^{D_s}) \quad i = 1, 2, \dots, N \quad (2.28)$$

Avec :

- $X_i$  : agent de recherche (Loup Gris) d'indice  $i$  ;
- $x_i^d$  : la position de l' $i^{me}$  Loup Gris dans la dimension  $d$  ;
- $D_s$  : la dimension de l'espace de recherche.

2. Deuxièmement, calcul des valeurs de performance pour l'ensemble des loups en utilisant la fonction objective. Ensuite, le classement descendant (du plus fort au plus faible) des agents de recherche en fonction de leurs performances et attribution des trois premiers meilleurs agents à  $X_\alpha$ ,  $X_\beta$ , et  $X_\delta$  respectivement.

$$\begin{aligned} \vec{X}_\alpha &= \vec{X}_1 \\ \vec{X}_\beta &= \vec{X}_2 \\ \vec{X}_\delta &= \vec{X}_3 \end{aligned} \quad (2.29)$$

3. Troisièmement, mise à jour des vecteurs des paramètres de convergence  $\vec{a}$ ,  $\vec{A}$  et  $\vec{C}$  par les équations suivantes. Indiquant ici que les coefficients de vecteur  $\vec{a}$  diminuent linéairement de 2 à 0 à la fin du processus pour le GWO.

$$\vec{a} = 2\left(1 - \frac{k}{K_{max}}\right) \quad \text{pour GWO} \quad (2.30)$$

Cependant,  $\vec{a}$  se calcule comme suit le cas de l'IGWO :

$$\vec{a} = \left(1 - \frac{k}{K_{max}}\right)\left(1 - \mu_{IGWO} \frac{k}{K_{max}}\right)^{-1} \quad \text{pour IGWO} \quad (2.31)$$

$$\vec{A} = 2\vec{a} \cdot r_1 - \vec{a} \quad (2.32)$$

$$\vec{C} = 2 \cdot r_2 \quad (2.33)$$

Avec :  $k$  et  $K_{max}$  indiquent l'itération courante et le nombre maximal d'itérations respectivement ;  $r_1$  et  $r_2$  sont deux vecteurs de valeurs générées aléatoirement dans l'intervalle  $[0,1]$  ;  $\mu_{IGWO}$  paramètre intrinsèque de l'algorithme IGWO.

4. Quatrièmement, en raison que la position de proie (l'optimum global) est inconnue, il est considéré que les loups  $\alpha$ ,  $\beta$  et  $\delta$  sont les plus proches de lui (figure 2.8). Ainsi, les comportements d'encerclement et de chasse peuvent être imités respectivement comme suit :

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{aligned} \quad (2.34)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \quad (2.35)$$

5. Cinquièmement, la mise à jour des positions des autres agents de recherches selon les trois meilleures solutions comme suit :

$$X_{k+1}^{\rightarrow} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (2.36)$$

6. Sixièmement, le processus de l'étape 2 à 5 est répété jusqu'à ce que le nombre maximal d'itérations  $K_{max}$  soit atteint. À la fin, les agents de recherche cessent bouger et l'agent de recherche  $X_\alpha$  est considérée comme la meilleure solution.

La figure 2.9 montre l'organigramme explicatif de déroulement des algorithmes d'optimisation GWO et IGWO.

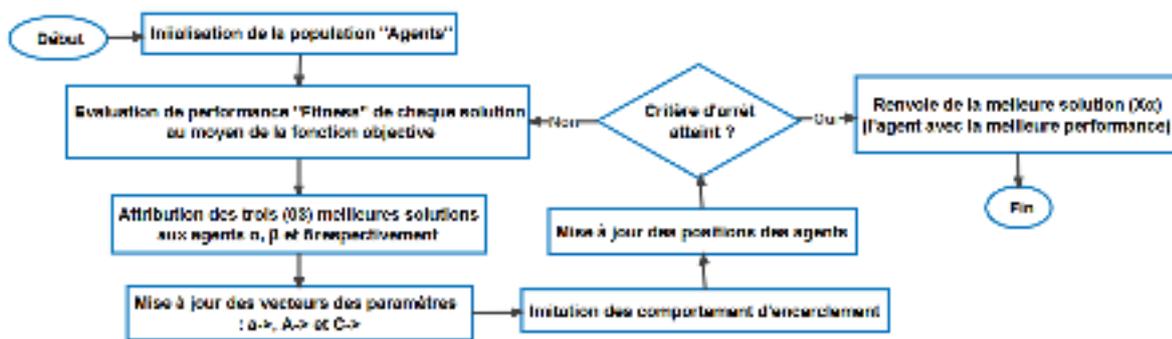


FIGURE 2.9 – Organigramme général de processus d'optimisation des algorithmes GWO, IGWO et HPSOGWO

### 2.6.4 Algorithme hybride HPSOGWO

HPSOGWO est un nouvel algorithme métahéuristique hybride, dans lequel certaines fonctionnalités de PSO "Particle Swarm Optimization" ont été fusionnées en GWO pour améliorer la capacité d'exploration de ce dernier et celle d'exploitation du PSO (Singh and Singh 2017).

L'hybridation est une opération courante dans la discipline d'optimisation. Elle consiste de combiner les fonctionnalités (ou certaines) des algorithmes "pères" pour obtenir d'autres "fils" plus performants. Au cours de ces dernières années, cette opération a connu un intérêt considérable, car les meilleures solutions des problèmes classiques ou de la vie quotidienne sont

obtenues à partir des algorithmes hybrides. D'une manière générale, l'hybridation est peut-être réalisée par (Talbi et al. 2013) :

1. Combiner une métaheuristique avec une autre (complémentaire) ;
2. Combiner la métaheuristique avec des méthodes exactes de la programmation mathématique, approches qui sont principalement utilisées dans la recherche opérationnelle ;
3. Combiner la métaheuristique et les approches de programmation par contraintes développées dans la communauté de l'intelligence artificielle ;
4. Combiner la métaheuristique avec les techniques d'apprentissage automatique "machine learning" et d'exploration de données "data mining".

Selon la taxonomie d'hybridation proposée par (Talbi 2002), l'opération d'hybridation peut être effectuée comme suit :

1. De bas niveau "low-level" ou de haut niveau "high-level" : l'hybridation de bas niveau porte sur la composition d'une seule méthode d'optimisation, dans laquelle une fonctionnalité donnée d'une métaheuristique est remplacée par une autre (relative à une autre métaheuristique). En revanche, dans les algorithmes hybrides de haut niveau, les différents métaheuristiques sont autonomes (les fonctionnalités internes ne sont pas mélangées) ;
2. De relais "relay" ou coopérative "teamwork" : dans l'hybridation de relais, les métaheuristiques sont appliquées l'une après l'autre, chacune utilise les sorties de précédente comme entrée. Par contre, l'hybridation du travail d'équipe représente des modèles d'optimisation coopératifs, dans lesquels nous avons de nombreux agents parallèles "coopératifs", où chaque agent effectue une recherche dans un espace de solution.

Vu les classifications précédentes, l'algorithme HPSOGWO est une hybridation entre deux métaheuristiques. Aussi, l'hybridation est de niveau bas, car les fonctionnalités sont mélangées. Et, elle est coopérative car les deux algorithmes pères "PSO" et "GWO" ne fonctionnent pas en séquentielle (l'un après l'autre) (Singh and Singh 2017).

La motivation derrière l'adoption de l'algorithme HPSOGWO dans cette étude est justifiée par plusieurs raisons. Premièrement, il est quasi-similaire au GWO, donc il conserve les majeurs avantages de ce dernier (simplicité de structure, facilité de codage, nombre d'opérations mathématiques faible et rapidité d'exécution). Deuxièmement, il est hybride et a prouvé une performance meilleure que ses "pères" (Singh and Singh 2017). Troisièmement, il n'a pas été exploité et évalué dans l'ingénierie des ressources en eau (à la limite de notre connaissance).

### 2.6.4.1 Principe et algorithme

L'algorithme HPSOGWO hérite du GWO l'imitation du comportement de chasse des Loups Gris dans la nature. Cependant, la mise à jour des positions des éléments de recherche (Loups Gris) est héritée depuis l'algorithme PSO. Les auteurs du HPSOGWO ont introduit un coefficient  $w$  pour paramétrer l'imitation l'encerclement de la proie (l'optimum global) par les loups (agents de recherche) (Équation 2.37). Ce coefficient permet de régler la phase d'exploration de l'espace de recherche. La phase d'exploitation (intensification de la recherche autour des solutions prometteuses) est inspirée de l'algorithme PSO (Equations 2.39 et 2.40).

Dans le processus d'exécution, le HPSOGWO partage avec le GWO et l'IGWO les trois (03) premières étapes :

- Les étapes 1, 2 et 3 : Les même équations 2.28, 2.29, 2.30, 2.32 et 2.33 de la section précédente sont gardées.

Pour les étapes suivantes :

4. Les processus d'encerclement et de chasse sont imités comme suit :

$$\begin{aligned}\vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - w * \vec{X}|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - w * \vec{X}|, \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - w * \vec{X}|\end{aligned}\quad (2.37)$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta\end{aligned}\quad (2.38)$$

Telle que  $w$  est la constante d'inertie, générée aléatoirement dans l'intervalle  $[0,1]$ .

5. Mise à jour des vitesses  $v_i$  et positions  $x_i$  des agents de recherche dans chaque dimension par l'approche de l'algorithme PSO comme suit :

$$v_i^{k+1} = w * (v_i^k + c_1 r_1 (x_1 - x_i^k) + c_2 r_2 (x_2 - x_i^k) + c_3 r_3 (x_3 - x_i^k)) \quad (2.39)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.40)$$

Tels que :

- $k$  : l'itération courante ;
- $r_3$  : valeur générée d'une manière aléatoire dans l'intervalle  $[0,1]$  ;
- $c_1, c_2$ , et  $c_3$  : paramètres de l'algorithme et doivent être fixés au départ.

6. Le processus de l'étape 2 à 5 est répété jusqu'à atteindre le critère d'arrêt (nombre maximal d'itération  $K_{max}$  ou l'absence d'amélioration de la meilleure solution trouvée pendant certain nombre d'itérations par exemple). Identiquement aux algorithmes GWO et IGWO, l'élément de recherche  $X_\alpha$  est considéré comme solution optimale à la fin du traitement.

La métaheuristique HPSOGWO partage avec le GWO et IGWO le même processus général d'optimisation (figure 2.9), en considérant les différences principales déjà mentionnées.

## **2.7 Conclusion**

Ce chapitre a été réservé à la présentation générale des algorithmes métaheuristiques évolutionnaires. En premier lieu, la définition mathématique de l'optimisation, la classification des méthodes utilisées, et les facteurs de choix entre elles sont abordés. En deuxième lieu, les métaheuristiques, leurs mécanismes du fonctionnement et dans quelles situations le recours à ces méthodes est préférable sont dépliés. En dernier lieu, les quatre algorithmes évolutionnaires (RGA, GSA, HPSOGWO et IGWO) sont mis en détails.

À travers ce chapitre, les points suivants peuvent être ressortis :

- Les méthodes d'optimisation constituent une pierre angulaire dans les modèles d'optimisation, à côté de la modélisation fiable et les simulateurs performants ;
- Le choix d'une méthode d'optimisation pour traiter un problème donné doit se faire à la base de plusieurs considérations, parmi lesquelles : la nature et la complexité du problème, la qualité des solutions ciblées, le temps et les ressources de computation offerts, la disponibilité de la méthode et sa maîtrise, ... etc ;
- Le problème d'optimisation du fonctionnement d'un réservoir unique ou d'un système de réservoirs est complexe à cause de plusieurs raisons, telles que : fonctions objectifs souvent non-linéaires, multimodales et multiples, contraintes du fonctionnement physique réel, haute dimensionnalité, incertitudes dues aux différents phénomènes naturels et socio-économiques, ... etc ;
- Les méthodes d'optimisation classiques présentent plusieurs inconvénients limitant leur emploi à grandes échelles et qui figurant parmi les sources de complexité du problème d'optimisation de fonctionnement des réservoirs. Les métaheuristiques sont des nouvelles méthodes développées dans l'expectative de surmonter les désavantages des anciennes. En effet, le recours à ces méthodes récentes est peut-être justifié par leurs capacités et avantages d'un côté, et d'autre côté par la nature et les caractéristiques de problème d'optimisation en question ;

- La nature des métaheuristiques évolutionnaires tant que des méthodes de calcul stochastique approché et leurs inconvénients, notamment la non-généralité de performance, impliquent l'examen de chaque algorithme de cette famille dans chaque discipline.

Ce dernier point s'inscrit parmi les objectifs de cette étude. En effet, quatre algorithmes évolutionnaires sont choisis pour traiter les problèmes relatifs à l'optimisation de la gestion opérationnelle d'un réservoir. Le premier est une version simple de l'algorithme génétique à codage réel (RGA). Il est choisi à cause de l'efficacité prouvée des algorithmes génétiques dans l'optimisation de tels problèmes. Le deuxième est l'algorithme de recherche gravitationnelle (GSA), dont la performance et la stabilité ont été attestées dans plusieurs domaines, mais il n'est pas largement exploités dans le traitement de la problématique en question. Le troisième et le quatrième sont l'algorithme de Loup Gris amélioré (IGWO) et l'algorithme hybride de l'essaim des particules et l'optimiseur de Loup Gris (HPSOGWO). Ces derniers sont des méthodes nouvelles dotées de plusieurs avantages, et l'examen de leurs facultés dans la matière de l'optimisation du fonctionnement des réservoirs représente une opportunité de recherche.

Dans la suite de ce travail, ces méthodes sont utilisées pour effectuer trois tâches principales à savoir :

1. L'entraînement des réseaux de neurones artificiels (ANNs) dans l'essai de prévision des apports du réservoir Hammam Boughrara (W. Tlemcen);
2. Recherche de réseaux de neurones artificiels évolutionnaires (EANNs) pour réaliser le même essai que les ANNs;
3. La dérivation des politiques optimales de la gestion opérationnelle du réservoir à travers une étude comparative.

Les quatre algorithmes discutés dans ce chapitre sont reprogrammés à la base des codes sources originales en langages Matlab, C# ou Python (selon le cas) et à la base des articles relatifs. La réimplémentation en langage de programmation *VisualBasic.NET (VB.NET)* est disponible librement pour vérification ou réutilisation dans le site d'hébergement et de gestion de développement des logiciels "*Github.com*", sous le lien "*https://github.com/SaadDAHMANI/RMOSS/tree/master/EvolutionaryAlgorithms*".

# Chapitre 3

## Les réseaux de neurones artificiels

### 3.1 Introduction

La puissance des réseaux de neurones artificiels "RNAs" (Artificial Neural Networks, "ANNs") a été prouvée dans divers domaines tels que l'approximation des fonctions, le traitement du signal, le traitement d'images, la reconnaissance vocale, la prévision, l'aide à la décision, l'auto-pilotage, la robotique, ... etc. Actuellement, les RNAs sont au cœur de l'intelligence artificielle (IA) explosée cette dernière décennie.

Les méthodes classiques de résolution des problèmes sont basées sur l'algorithmique, c-à-d, des procédures qui doivent être décrites en détails en connaissant les relations entre les différentes variables intervenants selon le cas. Une arborescence de possibilités à traiter peut s'éclater notamment si le nombre de variables est important. Les informations (données) sur les variables doivent être complètes et utilisées pour garantir la résolution et assurer la qualité des solutions. Au regard de ça, l'automatisation de traitement via des programmes informatiques devient complexe, laborieuse et fragile, où les bugs peuvent nuire ces programmes.

Au contraire à l'approche précédente, les réseaux de neurones artificiels constituent un alternatif pouvant échapper ces trappes. En premier lieu, les RNAs sont des boîtes noires disposant une grande capacité de généralisation. Une fois entraînés, ils peuvent étendre la capacité de reconnaissance même pour des cas n'ont pas figurés auparavant en entraînement. En deuxième lieu, les RNAs peuvent traiter des problèmes avec une partielle connaissance d'informations, l'une des habilités derrière la puissance du cerveau humain, imitée (ou plus exact, partiellement imitée) par les RNAs. En dernier lieu, la complexité reliant les différentes variables gérant un problème n'a pas de grand effet sur la faculté des RNAs tant que ces variables sont énumérés et y en introduites.

A cause de ces aptitudes, les RNAs sont largement adoptés dans diverses disciplines, y compris l'hydrologie distinguée par ses phénomènes de grandes ampleurs, incertitudes et complexités. La prévision des débits des cours d'eau n'échappe pas à la règle. Elle est de grande importance, notamment pour les gestionnaires des barrages réservoirs et les services liés, ainsi aux concepteurs des ouvrages de protection contre inondations.

Comme rapporté précédemment, les différentes approches d'optimisation stochastique du fonctionnement des réservoirs (ISO, ESO et P-S-O) requièrent des modèles mathématiques efficaces de prévision des apports, dont les RNAs peuvent constituer un bon choix. Ce chapitre est inscrit dans ce contexte. Il cible de traiter les points suivants concernant ces modèles :

- La présentation des RNAs et les notions de base de leur fonctionnement, avec la concentration sur ceux de structure multicouche ;
- Le mécanisme de l'apprentissage des RNAs et les méthodes dites "classiques" d'apprentissage (la rétropropagation de gradient de l'erreur "Backpropagation, BP" et Levenberg-Marquardt "LM").
- La présentation de la manière d'incorporer les métaheuristiques dans l'apprentissage des RNAs. Deux (02) algorithmes métaheuristiques énoncés auparavant (chapitre 2) sont sélectionnés, le RGA et le HPSOGWO. La comparaison entre ces derniers et les méthodes classiques dans la prévision des apports du réservoir Hammam Boughrara est fait l'objet du chapitre 7.

## **3.2 Historique**

Les RNA ont commencé à voir le jour dès 1890 avec les travaux de William James (1842-1910), un célèbre psychologue et philosophe américain qui a introduit le concept de mémoire associative, où il a proposé une loi de fonctionnement de l'apprentissage sur les réseaux de neurones. En 1943, W. McCulloch et W. Pitts, deux bio-physiciens de l'université de Chicago (États-Unis), ont proposé un passage effectif des observations neurophysiologiques et anatomiques au modèle formel des neurones, appelé "Modèle de McCulloch et Pitts". La figure (3.1) démontre le schéma explicatif de ce modèle. Aussi, ils ont démontré qu'un réseau de neurones discret peut représenter n'importe quelle fonction booléenne (principe de base de fonctionnement des processeurs électroniques). Malgré sa simplicité qui ne correspond pas au fonctionnement réel et entier d'un neurone biologique, le modèle de McCulloch et Pitts reste à l'heure actuelle un élément de base des réseaux de neurones artificiels.

*Le neurone reçoit les entrées  $(x_1, x_2, \dots, x_n)$  et calcule le potentiel  $O$  par pondération :*

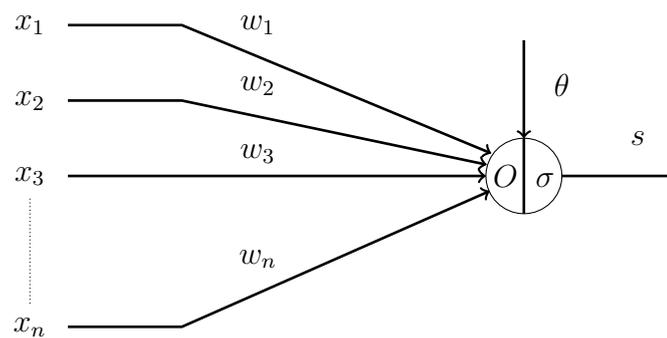


FIGURE 3.1 – Modèle formel d'un neurone proposé par W. McCulloch et W. Pitts

$O = w_1x_1 + w_2x_2 + \dots + w_nx_n$ . Puis, une décision est prise sur la sortie  $S_r$  en fonction de la valeur de  $O$  et du seuil  $\theta$ ; si  $O > \theta$ , alors  $s = +1$ , sinon  $s = -1$ . Cette dernière expression représente la fonction de décision  $\sigma$ , donnant la valeur de la sortie  $S_r$  en fonction de  $O$  :  $s = \sigma(O)$ .

En 1949, grâce aux travaux de D. Hebb (physiologiste américain) sur le conditionnement chez les animaux où il a établi une fameuse règle connue sous l'appellation de "Règle de Hebb". Cette règle permet de modifier la valeur des coefficients synaptiques (connexions entre neurones) en fonction de l'activité des éléments qu'ils relient. Par une autre manière, la règle suggère d'accroître la valeur des coefficients synaptiques entre les neurones formels qui ont une activité synchronisée, et de ne rien modifier si ce n'est pas le cas (Blayo and Verleysen 1996).

En 1958, Rosenblatt proposa le premier système artificiel inspiré du système visuel capable d'apprendre par expérience, nommé "Perceptron" (Rosenblatt 1958). Le perceptron est composé de deux couches neurones, la première de perception et la deuxième de prise de décision. Ce réseau a pu calculer quelques fonctions logiques et reconnaître quelques formes simples.

En 1960, B. Widrow et T. Hoff, deux chercheurs de l'université Stanford (États-Unis) ont développé un réseau de neurones à simple couche sur la base du neurone formel de McCulloch et Pitts. Ce réseau est appelé "ADALINE (Adaptive Linear Neuron)", et sert à la base des modèles des RNAs multicouches par la suite. Parallèlement, en 1961, E. Caianielle a développé une théorie de traitement de l'information à la base des équations neuroniques (Blayo and Verleysen 1996).

Au début des années 70, le domaine des RNAs a connu un désintéressement des chercheurs. La publication du livre "Perceptrons : an introduction to computational geometry" en 1969 par les deux chercheurs M. Minsky and S. Papert est l'une des causes de ce désintéressement. Les auteurs du livre ont argumenté la limitation du perceptron (réseaux de neurones à une seule couche), et en particulier l'incapacité de résoudre les problèmes non linéairement sé-

parables. Heureusement, au début des années 80 et notamment avec les travaux de Hopfield en 1982 et parallèlement de Werbos, le développement des réseaux de neurones est rétabli. Hopfield a proposé les réseaux de neurones associatifs, tandis que Werbos a proposé un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron.

En 1986, D.E Rumelhart, G.E Hinton et R.J Williams ont publié une nouvelle approche d'apprentissage des RNAs dite "rétropropagation de gradient de l'erreur" (notée couramment "Backpropagation"). L'approche propose d'ajouter des couches "cachées" entre les couches d'entrée et de sortie, ainsi d'ajuster les poids des connexions neuronales (les valeurs  $w_i$  dans la figure 3.1) permettant de réduire les différences entre les sorties calculées par l'RNA et les valeurs désirées (Rumelhart et al. 1988). À partir de cette année, les recherches sur les RNAs ont repris ses ampleurs, et le développement des approches et des architectures n'a pas cessé à l'heure actuelle.

### 3.3 Principe du fonctionnement d'un neurone formel

Le neurone formel est un modèle mathématique du neurone biologique. En toute rigueur, le terme formel est utilisé à la suite du mot "neurone" afin de distinguer l'objet mathématique de la cellule biologique. Après avoir les entrées ( $x_j$ ,  $j = 1, 2, \dots, n$ ), les poids  $w_{ij}$  (où  $i$  indique l'indice de neurone dans un modèle multicouche) et leur pondération  $O_i$  sont calculés. Puis, au moyen d'une non-linéarité  $f$  appelée fonction d'activation ou fonction de seuil, une décision " $s_i$ " est prise. Notant que les entrées au neurone peuvent être externes ou calculées par autres neurones. La figure (3.2) démontre le schéma explicatif d'un neurone formel.

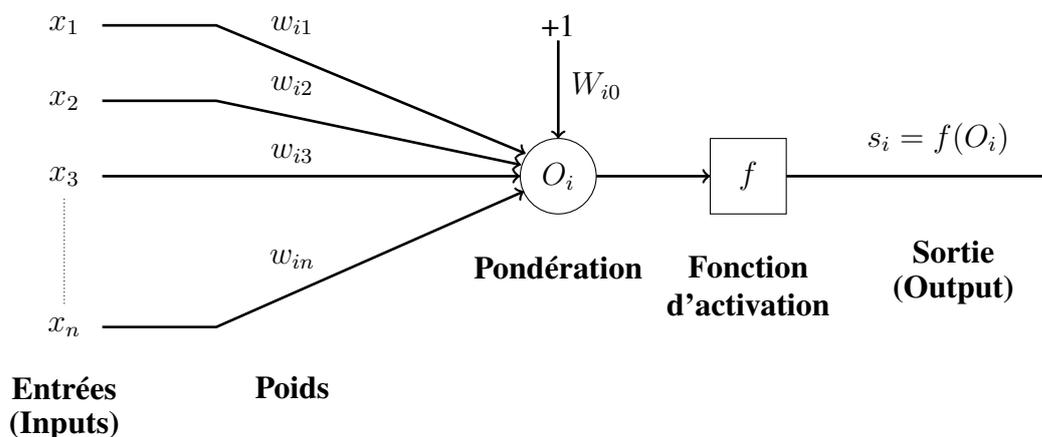


FIGURE 3.2 – Schéma d'un neurone formel

La sortie  $s_i$  de neurone  $i$  est donnée par l'équation suivante :

$$s_i = f(O_i) \quad (3.1)$$

Tel que :

$$O_i = \sum_{j=1}^n x_j w_{ij} - (+1 \times w_{i0}) \quad (3.2)$$

Avec :

- $O_i$  : le potentiel du neurone d'indice  $i$  ;
- $x_{ij}$  : entrée d'indice  $j$  au neurone  $i$  ;
- $w_{ij}$  : poids (ou coefficient synaptique) reliant l'entrée  $j$  avec le neurone  $i$  ;
- $w_{i0}$  : poids "ou biais" d'un paramètre additionnel considéré comme la valeur du seuil interne du neurone. Ce seuil est égale à (+1) dans ce cas.

Il est important de noter que les RNAs peuvent avoir une ou plusieurs sorties finales (dans la couche terminale). Dans ce qui suit, on s'intéresse notamment aux modèles dotés d'une seule sortie dans la couche finale.

### 3.3.1 La fonction d'activation

La fonction d'activation (ou fonction de seuil) est une fonction mathématique présentée généralement par une non-linéarité. Elle permet de définir l'état interne de neurone en fonction de son entrée totale. Plusieurs fonctions sont proposées, telles que les fonctions : linéaire, sigmoïde, tangente hyperbolique (bipolaire sigmoïde), Heaviside, Gaussienne, ... etc. Le choix de la fonction d'activation dépend de l'application (Darras (2015)). Les trois (03) premières fonctions (linéaire, sigmoïde et tangente hyperbolique) (tableau 3.1, figure 3 (Annexes)) sont utilisées dans cette étude. Ces fonctions présentent l'intérêt d'être continues, dérivables et bornées et sont fréquemment utilisées en hydrologie (Thirumalaiah and Deo 1998; Darras 2015).

TABLE 3.1 – Fonctions d'activation utilisées dans cette étude

Nomination	Fonction	Bornes
Linéaire	$f(L) = \alpha L$ , si : $A < \alpha L < B$ $f(L) = A$ , si : $\alpha L \leq A$ $f(L) = B$ , si : $\alpha L \geq B$	[A, B]
Sigmoïde (logistique ou marche douce)	$f(L) = \frac{1}{1+e^{-\alpha L}}$	]0, 1[
Bipolaire sigmoïde (tangente hyperbolique)	$f(L) = \frac{2}{1+e^{-\alpha L}} - 1$	] -1, 1[

### 3.4 Réseau de neurones multicouche

Un réseau de neurones est un ensemble de neurones agencés selon une architecture adaptée à une application souhaitée. Une grande variété d'architectures peut-être établie. Cependant, le RNA multicouche comporte en général au moins trois (03) couches connectées entre elles. Une couche d'entrée, une ou plusieurs couche(s) cachée(s), et une couche de sortie (Borne et al. 2007). De plus, une entrée constante  $x_0$  est connectée à chacun des neurones des couches cachées et de la couche de sortie. Cette entrée dont la valeur est fixée par l'opérateur, permet d'effectuer une transformation affine du signal estimé en sortie. En effet, lorsque toutes les entrées sont nulles, il peut être nécessaire que la sortie ne le soit pas. Bien que cette entrée constante ne figure pas en général sur les schémas représentant les réseaux de neurones, elle est systématiquement introduite (Darras 2015). La figure (3.3) montre un schéma explicatif d'un RNA multicouche, tandis que l'équation (3.3) exprime son fonctionnement mathématique. L'architecture multicouche est celle utilisée dans cette étude.

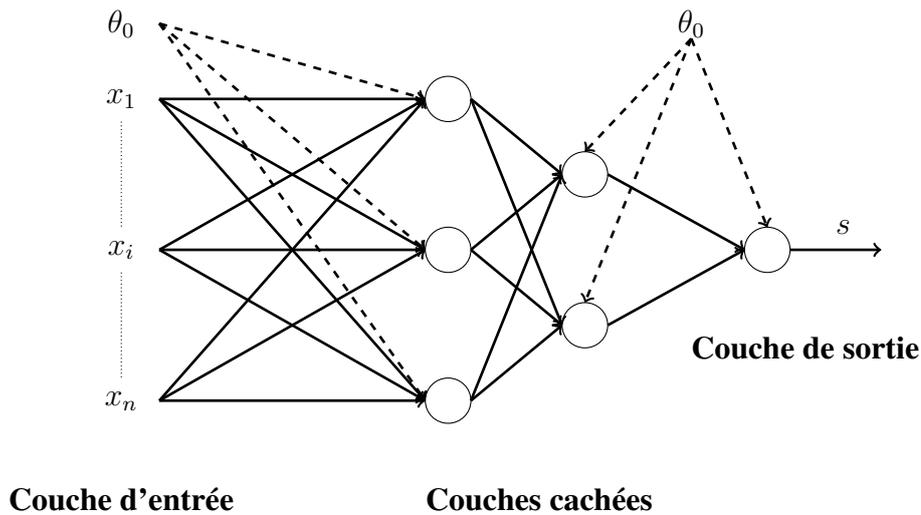


FIGURE 3.3 – Schéma explicatif de structure d'un RNA multicouche

Mathématiquement, la sortie finale  $s$  se calcule comme suit :

$$s = w_{s0}\theta_0 + \sum_{i=1}^{N_c} \left( w_{si} \left( f \left( \sum_{j=0}^n (w_{ij}x_j) \right) \right) \right) \quad (3.3)$$

Tels que :

- $s$  : la sortie du réseau de neurones ;
- $\theta_0$  : l'entrée constante du réseau ;
- $x_j$  ( $j = 1, \dots, n$ ) : les variables d'entrée ;
- $w_{s0}$  : le poids synaptique "biais" reliant l'entrée constante au neurone de sortie ;
- $N_c$  : le nombre de neurones cachés ;

- $w_{si}$  : le poids synaptique reliant le neurone d'indice  $i$  de la couche cachée au neurone de sortie ;
- $f(\cdot)$  : la fonction d'activation des neurones de la couche cachée ;
- $w_{ij}$  : le poids synaptique reliant la variable d'entrée d'indice  $j$  au neurone d'indice  $i$  de la couche cachée.

### **3.4.1 Principaux avantages des RNAs multicouche**

Les principaux avantages encourageant l'utilisation d'un RNA multicouche résument dans les points suivants :

**§1 Approximation universelle** : Selon (Hornik et al. 1989), toute fonction bornée suffisamment régulière est peut-être approchée uniformément avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation bornée, et un neurone de sortie linéaire. Encore, selon (Cybenko 1989), le perceptron multicouche (MLP : Multi-Layer Perceptron en anglais), est capable d'approcher n'importe quelle fonction non-linéaire bornée et dérivable en tout point. Aussi, il a été prouvé qu'un MLP à 3 couches (entrée-cachée-sortie) permet la résolution de n'importe quel problème non-linéaire (Jordan 2007). D'une manière générale, il a été prouvé à travers diverses études que les réseaux de neurones possèdent une forte capacité de généralisation, c-à-d., une fois qu'ils sont entraînés, ils sont capables de générer des résultats précis, même pour les cas qu'ils n'ont jamais vu auparavant (Kişi 2004).

**§2 Architecture compacte** : Le perceptron multicouche dont la sortie dépend non linéairement de ses paramètres est plus économique qu'un modèle dont la sortie dépend linéairement de ses paramètres (Darras 2015). Encore, selon (Wilamowski and Irwin 2016), les réseaux de neurones artificiels limités en nombre de couches cachées permettent une généralisation plus importante que ceux dotés d'un nombre de couches important. Par autre manière, dans certains cas, l'augmentation de nombre de couches cachées d'un réseau de neurones artificiels permet d'avoir des bons résultats d'apprentissage au détriment de la généralisation du réseau.

## **3.5 Architectures des réseaux de neurones artificiels**

Généralement, il existe deux catégories d'architectures, les réseaux statiques (non récurrents, "feedforward en Anglais") et les réseaux dynamiques (récurrents, "recurrent en Anglais").

### 3.5.1 Les réseaux de neurones statiques

Cette famille des RNA se caractérise par la circulation de l'information dans un sens unique, de l'entrée vers la sortie de réseau. Par autre manière, dans aucun cas, la sortie du réseau de neurones ne se réinjecte jamais comme entrée du réseau ni d'une façon directe ou au moyen d'un neurone quelconque. A cet effet, l'information sortante d'un RNA au temps  $t$  (itération  $k$ ) n'a aucun effet sur l'information sortante au temps  $t + 1$  (itération  $k + 1$ ). Ces réseaux sont nommés aussi "RNA non-bouclés". La figure 3.4 montre un exemple des réseaux statiques.

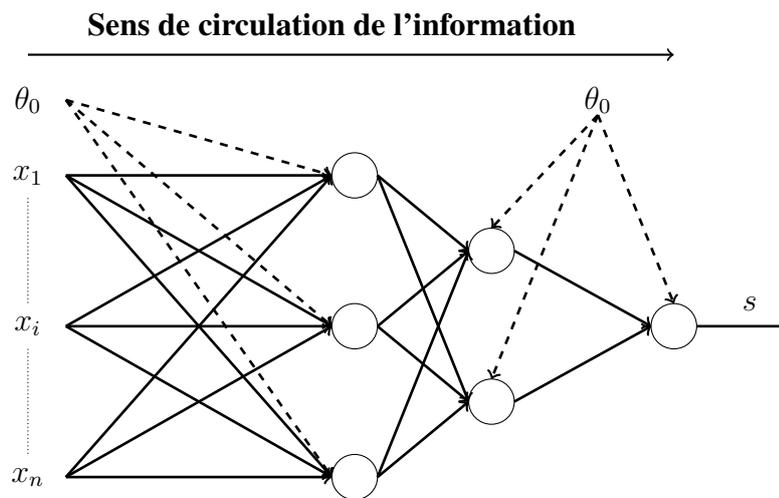


FIGURE 3.4 – Exemple d'un réseau de neurones artificiels statique "non-récurrent"

### 3.5.2 Les réseaux de neurones dynamiques

Contrairement à l'architecture statique, les RNAs de cette famille sont caractérisés par une circulation récurrente de l'information. Cette dernière peut-être soit issue de l'RNA lui-même (RNA récurrent ou bouclé) (figure 3.5), ou mesurée sur un processus physique puis réintroduite dans l'RNA (apprentissage dirigé par les observations) (Darras 2015). Dans le premier cas, les sorties à un temps donné (exprimé par l'itération  $k$ ) ne dépendent pas uniquement aux entrées, dites *exogènes* dans ce cas, mais aussi au/aux sortie(s) estimée(s) par le réseau lui-même.

## 3.6 Modèle mathématique des réseaux de neurones

Avant d'aborder l'apprentissage des RNA, les équations de base reliant les entrées et les poids d'une part, et les sorties d'un RNA d'autre part sont à définir. Le modèle considéré est un réseau de neurones artificiels multicouche non récurrent.

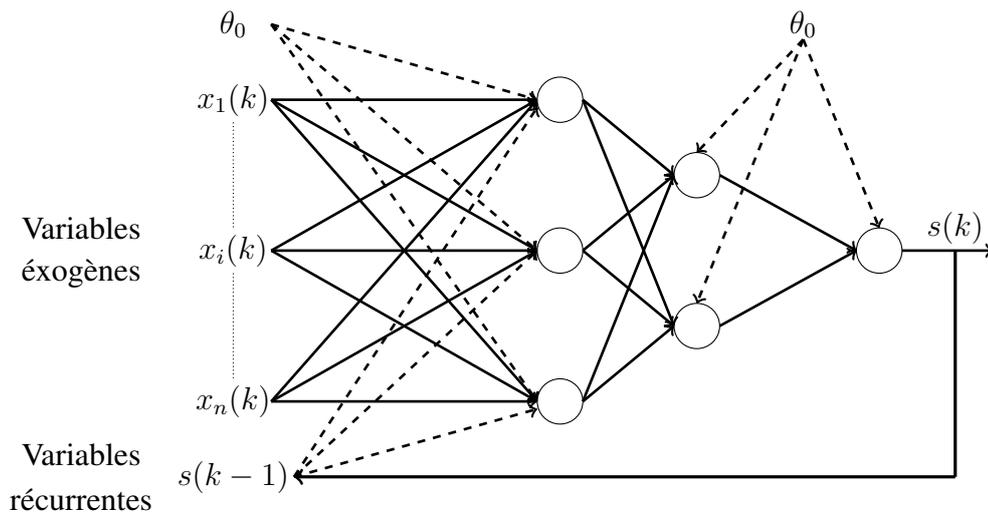


FIGURE 3.5 – Exemple d'un réseau de neurones artificiels dynamique "récurrent"

Dans le cas général, étant donné un ensemble de couples entrée(s) [A] /sortie(s) [B] observés de dimensions  $(P \times M)$  et  $(P \times N)$  respectivement, définis comme suit :

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2m} & \dots & x_{2M} \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ x_{p1} & x_{p2} & \dots & x_{pm} & \dots & x_{pM} \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ x_{P1} & x_{P2} & \dots & x_{Pm} & \dots & x_{PM} \end{bmatrix}$$

$$B = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} & \dots & y_{1N} \\ y_{21} & y_{22} & \dots & y_{2n} & \dots & y_{2N} \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ y_{p1} & y_{p2} & \dots & y_{pn} & \dots & y_{pN} \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ y_{P1} & y_{P2} & \dots & y_{Pn} & \dots & y_{PN} \end{bmatrix}$$

Dans ce cas, ces données peuvent être utilisées successivement (de  $p = 1$  jusqu'à  $p = P$ ) pour entraîner un RNA comme suit (Figure 3.6) :

Le calcul des sorties de la couche " $l$ " contenant " $n_l$ " neurones en fonction des sorties de la couche précédente " $l - 1$ " contenant " $n_{l-1}$ " neurones se fait par l'équation suivante :

$$s_i^l = f \left( \sum_{j=0}^{n_{l-1}} w_{ij} s_j^{l-1}(k) \right) \quad \text{pour } i = 1, 2, \dots, n_l; \quad l = 1, 2, \dots, L \quad (3.4)$$

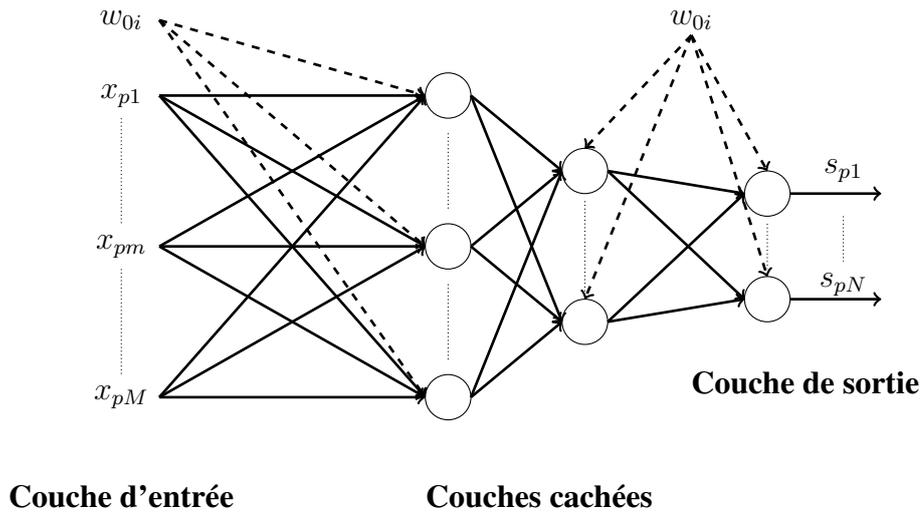


FIGURE 3.6 – RNA multicouche

Avec :

$$s_0^l(k) = 1 \quad l = 0, 1, 2, \dots, L \quad (3.5)$$

$$s_i^0(k) = x_{pi}(k) \quad i = 1, 2, \dots, M; \quad p = 1, 2, \dots, P \quad (3.6)$$

$$s_i(k) = s_i^L(k) \quad i = 1, 2, \dots, N \quad (3.7)$$

Donc, les sorties de la première couche sont calculées pour chaque vecteur d'entrées  $\vec{X}_p$  comme suit :

$$s_{pi}^1 = f \left( \sum_{j=0}^M w_{ij} x_{pj}(k) \right) \quad i = 1, 2, \dots, n_1; \quad p = 1, 2, \dots, P \quad (3.8)$$

Les sorties finales du réseau pour chaque vecteur d'entrée  $\vec{X}_p$  sont calculées comme suit :

$$s_{pi} = s_{pi}^L = f \left( \sum_{j=0}^{n_{L-1}} w_{ij} s_{pj}^{L-1}(k) \right) \quad i = 1, 2, \dots, N; \quad p = 1, 2, \dots, P \quad (3.9)$$

Les paramètres des équations précédentes sont définis comme suit :

- $s_i^l$  : la sortie de neurone  $i$  de la couche  $l$ ;
- $f(\cdot)$  : la fonction d'activation;
- $w_{ij}$  : le poids synaptique reliant le neurone (ou l'entrée)  $j$  de la couche  $l-1$  avec le neurone  $i$  de la couche  $l$ ;
- $s_j^{l-1}$  : la sortie de neurone  $j$  appartenant à la couche  $l-1$ ;
- $n_l$  : indique le nombre de neurones (nœuds) dans la couche  $l$ ;
- $L$  : le nombre total de couches;
- $x_{pi}$  : l'entrée  $i$  de vecteur  $\vec{X}_p$  (Figure 3.6);
- $w_{0i}$  : le seuil adaptable du neurone  $i$ ;
- $k$  : indique l'itération durant le processus d'apprentissage.

### **3.7 Apprentissage des réseaux de neurones**

Les réseaux de neurones artificiels sont des modèles de boîtes noires contenant un ensemble de paramètres permettant la transformation des entrées en sorties. Cependant au départ, un réseau de neurones ne contient aucune information et ne permet pas l'obtention des sorties désirées. Pour cette raison, il doit subir une phase dite "d'apprentissage" ou "d'entraînement" ("*learning*" ou "*training*" en anglais) qui consiste à ajuster ses paramètres afin d'obtenir les sorties désirées. L'entraînement des RNA ressemble au calage (ou calibration) des modèles mathématiques (Kişi 2004). Durant cette phase, un algorithme dit d'apprentissage "ou d'entraînement" met à jour les poids synaptiques et les biais "biases ou thresholds en anglais"  $\vec{W}$  du réseau en fonction d'un jeu de données. Donc, le RNA "apprend" en utilisant des exemples. Et, si cette phase est réalisée correctement, le RNA acquiert une mémoire permettant de générer des sorties proche (ou même identiques) de celles observées pour les mêmes entrées.

Il est important de noter qu'il est facile d'entraîner un RNA multicouche doté d'un nombre important de neurones. Néanmoins, plusieurs études ont montré qu'un RNA de structure compacte (faible nombre de neurones) assure une meilleure capacité de généralisation. Cela signifie qu'il répondra correctement pour les exemples (couples entrées/sorties) non utilisés pour l'entraînement. Si trop de neurones sont utilisés, alors le réseau peut être surentraîné sur les exemples d'apprentissage, mais il échouera sur ceux jamais utilisés. Avec un plus petit nombre de neurones, le réseau ne peut pas être formé à de très petites erreurs, mais il peut produire des approximations beaucoup mieux pour de nouveaux exemples (Wilamowski 2010; Wilamowski and Irwin 2016).

En général, il existe deux modes d'apprentissage (ou d'entraînement), supervisé et non supervisé.

#### **3.7.1 Apprentissage non supervisé**

Dans ce cas, des exemples ou "prototypes" ou "patrons" présentés au réseau qu'on laisse s'auto-organiser au moyen de lois locales qui régissent l'évolution des poids synaptiques. Ce mode d'apprentissage est appelé aussi "apprentissage par compétition" (Borne et al. 2007).

#### **3.7.2 Apprentissage supervisé**

Dans ce type d'apprentissage, on cherche à imposer au réseau un fonctionnement donné de façon qu'à partir des entrées présentées, les sorties du réseau à prendre des valeurs

données en modifiant les poids synaptiques de l'RNA. Le réseau se comporte alors comme un filtre dont les paramètres de transfert sont ajustés à partir des couples entrées/sorties présentés (Borne et al. 2007). En résumé, l'apprentissage supervisé consiste à chercher une fonction paramétrée  $S_r(\vec{X}, \vec{W})$  réalisée à l'aide d'un réseau de neurones, pour laquelle la fonction de coût "l'erreur totale  $E(\vec{W})$ " est minimisée. Plusieurs fonctions d'erreur d'apprentissage sont proposées, la plus utilisée est connue sous le nom de "la règle de Widrow–Hoff", donnée par l'équation (3.10) (Wilamowski and Irwin 2016) :

$$\text{Minimiser } E(W) = \sum_{p=1}^P \left[ \vec{Y}_p - S_r(\vec{X}_p, \vec{W}) \right]^2 \quad (3.10)$$

Avec,  $\vec{X}_p, \vec{Y}_p$  sont des vecteurs d'entrées et de sorties observées respectivement, d'un échantillon d'exemples de taille  $P$ .  $\vec{W}$  est le vecteur des poids synaptiques de l'RNA à entraîner. Le RNA est représenté par la fonction  $S_r(\vec{X}_p, \vec{W})$ , dont elle renvoie des sorties calculées.

L'adaptation des paramètres du réseau s'effectue au moyen d'un algorithme d'optimisation, l'initialisation des poids synaptiques étant le plus souvent aléatoire (Borne et al. 2007). Les valeurs des poids sont définies dans l'intervalle  $[-1, 1]$ . On s'intéresse dans ce qui suit au mode d'apprentissage supervisé.

### 3.7.3 Principe d'apprentissage supervisé des réseaux de neurones

Prenant un neurone formel simple (Figure 3.7). En général, les poids synaptiques du neurone sont ajustés comme suit :

$$w_{ij}^{k+1} = w_{ij}^k + C \cdot (y_j - s_j^k) \cdot In_i \quad (3.11)$$

Tels que :

- $w_{ij}^k$  : poids synaptique de l'entrée d'indice  $i$  pour le neurone d'indice  $j$  à l'itération  $k$  ;
- $C$  : le taux d'apprentissage compris entre 0 et 1 ;
- $y_j$  : la sortie observée ;
- $s_j^k$  : la sortie (output) calculée par le neurone d'indice  $j$  à l'itération  $k$ , et peut être l'entrée d'un autre neurone (succédant) ;
- $In_i$  : entrée (input) d'indice  $i$ , et peut être la sortie d'un autre neurone (appartenant à la couche précédente).

Au cours de l'apprentissage d'un RNA, les poids synaptiques et les biais  $W_{ij}$  sont

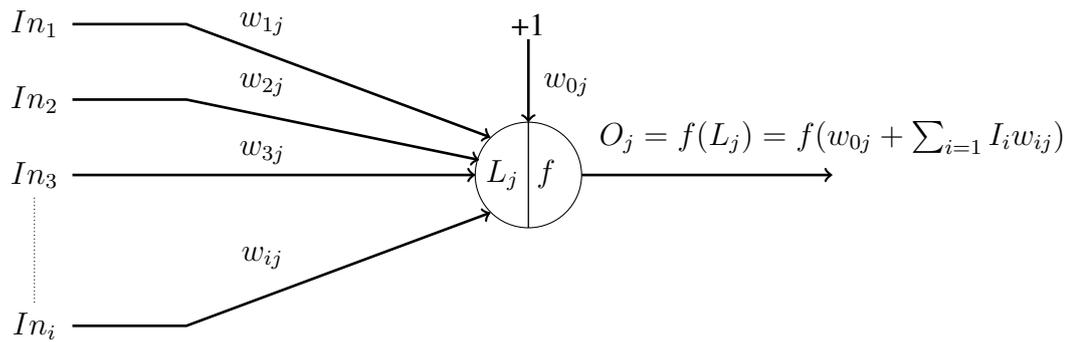


FIGURE 3.7 – Schéma d'un neurone formel, élément de base d'un RNA multicouche

modifiés dans le but de **minimiser l'erreur globale**. Par conséquent, l'équation (3.10) peut être réécrite comme suit :

$$\text{Minimiser } E = \sum_{p=1}^P E_p \quad (3.12)$$

$$E_p = \sum_{n=1}^N (y_{pn} - s_{pn})^2 \quad (3.13)$$

Tels que :

- $E$  : l'erreur globale d'entraînement ;
- $P$  : le nombre total des paires (couples) d'entraînement (entrées-sorties) ;
- $E_p$  : l'erreur d'entraînement de couple entrée(s)/sortie(s) d'indice  $p$  ;
- $N$  : le nombre des nœuds de sortie ;
- $s_{pn}$  : la valeur calculée par le RNA au nœud de sortie d'indice  $n$  pour le couple d'entraînement d'indice  $p$  ;
- $y_n$  : la valeur observée d'indice  $n$ .

Si le RNA dispose une seule sortie ( $N = 1$ ) dans la couche finale (Figure 3.8), les équations 3.12 et 3.13 peuvent être simplifiées comme suit :

$$E = \sum_{p=1}^P E_p = \sum_{p=1}^P (y_p - s_p)^2 \quad (3.14)$$

### 3.8 Méthodes classiques d'apprentissage des ANNs

Il existe une large variété d'algorithmes d'apprentissage des réseaux de neurones artificiels. Ils sont des méthodes d'optimisation utilisées pour minimiser la fonction d'erreur (équation 3.14) fréquemment utilisée. Parmi les plus utilisés, on cite : l'algorithme de rétropropagation de l'erreur (Back-propagation), le gradient conjugué (Conjugate Gradient), Gauss–Newton, la corrélation cascade (Cascade correlation), Levenberg–Marquardt, ... etc.

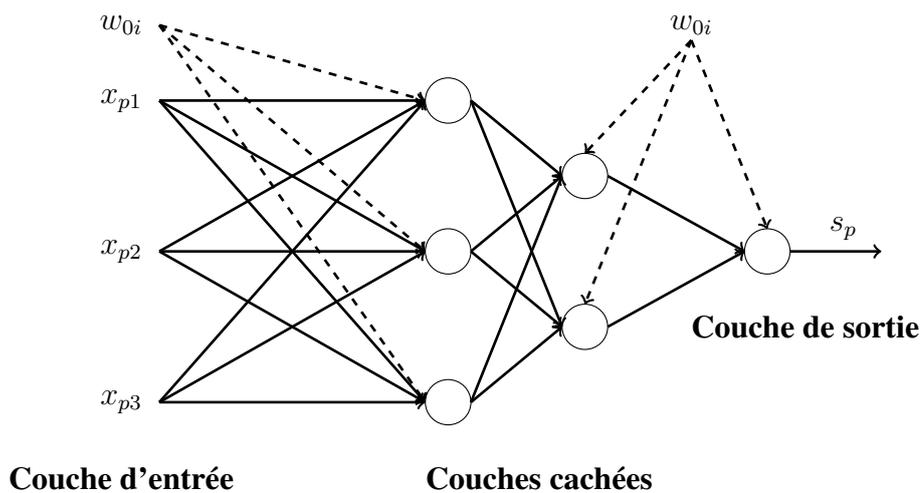


FIGURE 3.8 – RNA multicouche (01 d'entrées, 01 de sortie, 02 couches cachées)

Dans la suite, on s'intéresse aux méthodes de rétropropagation du gradient de l'erreur "BP" et Levenberg-Marquardt "LM".

### 3.8.1 Méthode de rétropropagation du gradient de l'erreur (Back-propagation)

Elle est proposée par (Rumelhart et al. 1988). C'est la méthode la plus utilisée pour l'entraînement des RNA multicouche à propagation avant (Figure 3.8) (Kişi 2007). Elle pourrait être considérée comme l'une des percées les plus significatives dans l'entraînement des réseaux neuronaux (Wilamowski and Irwin 2016). Aussi, il offre une convergence rapide vers une erreur minimale dans la majorité des cas (Jordan 2007). Tandis que, cette méthode peut présenter une convergence lente ou un comportement oscillatoire dans certains cas. De ce fait, les chercheurs ont développé les méthodes de gradient conjugué et la corrélation cascade qui peuvent surmonter ces problèmes (Thirumalaiah and Deo 2000).

**§1 Principe :** Dans le schéma de rétro-propagation, les poids et les biais  $\vec{W}$  sont ajustés par le déplacement de gradient de la fonction d'erreur (équation 3.12) dans le sens négatif du gradient jusqu'à l'achèvement de convergence. A ce niveau, une erreur minimale "Erreur d'apprentissage" fixée au départ est atteinte.

On commence l'entraînement par un choix aléatoire des valeurs initiales des poids. On présente le premier vecteur d'entrée. Une fois la sortie du réseau est calculée par les équations (3.4)-(3.8), l'erreur correspondante  $E$  et le gradient de l'erreur par rapport à tous les poids sont calculées. Ensuite, les poids synaptiques sont ajustés dans la direction opposée à celle du gradient de l'erreur. On refait la même procédure pour tous les exemples d'apprentissage. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches à celles

désirées.

L'algorithme de rétro-propagation est présenté dans la section (Algorithme 1, Annexes).

### **3.8.2 Méthode de Levenberg-Marquardt**

Elle est développée indépendamment par Kenneth Levenberg and Donald Marquardt (Marquardt 1963). Cette méthode est très utilisée pour minimiser des fonctions souvent non-linéaires à cause de sa convergence rapide. Dans le domaine des réseaux de neurones, elle est fréquemment exploitée pour entraîner les RNA de taille faible ou moyenne, car elle est consommatrice de mémoire notamment si une précision élevée d'apprentissage est requise. Ce dernier problème est due aux calcul et stockage de la matrice Jacobienne de la fonction d'erreur d'apprentissage.

Au cours de ces dernières années, l'algorithme de Levenberg-Marquardt est utilisé dans la discipline de prévision par réseaux de neurones, notamment en hydrologie. Plusieurs travaux de prévision des débits des cours d'eau et apports des réservoirs ont été publiés tels que (Thirumalaiah and Deo 2000 ; Kişi 2004 ; Kişi 2007 ; etc).

La méthode de Levenberg-Marquardt mélange la méthode de descente de gradient et celle de Gauss-Newton. Heureusement, il hérite l'avantage de vitesse de l'algorithme Gauss-Newton et la stabilité de la méthode de descente de gradient. Elle est plus robuste que la méthode de Gauss-Newton, parce que dans de nombreux cas, elle peut bien converger même si la surface d'erreur est beaucoup plus complexe que la situation quadratique. Bien que la méthode de Levenberg-Marquardt a la tendance d'être un peu plus lente que l'algorithme Gauss-Newton (en situation convergente), elle converge beaucoup plus vite que la méthode de descente de gradient (Wilamowski and Irwin 2016 ; Haykin 2009).

La version utilisée dans cette étude est celle décrite par (Roweis 1996). Elle est implémentée dans la plateforme de source ouverte "*Accord.NET Framework*" (Souza et al. 2014), disponible sous : <http://accord-framework.net/>, ou <https://github.com/accord-net/framework>. Le principe et l'algorithme de la méthode de Levenberg-Marquardt sont présentés dans les sections (32 et algorithme 2, Annexes).

### **3.8.3 Inconvénients des méthodes classiques d'apprentissage**

Parmi les inconvénients des méthodes classiques d'entraînement des réseaux de neurones artificiels, telles que la rétro-propagation (BP) et Lenvenberg-Marquard (LM), on cite les

points suivants (Yao 1993 ; Wilamowski and Irwin 2016) :

1. Elles peuvent manipuler seulement des problèmes avec des modèles relativement petits parce que la taille de la matrice Jacobienne est proportionnelle au nombre de modèles (exemples de données) d'entrée ;
2. La méthode LM a été écrite uniquement pour les réseaux MLP, qui ne sont pas les meilleures architectures pour les réseaux neuronaux ;
3. Le changement de la fonction objective d'entraînement (par exemple, l'équation 3.10) par une autre, nécessite la réécriture d'une partie importante du code d'exécution de ces méthodes. La réécriture touche notamment la procédure de calcul de la matrice Jacobienne. Sans oublier que si la fonction à minimiser n'est pas quadratique, le calcul de la matrice Hessienne est nécessaire ;
4. Ces méthodes sont sensibles aux paramètres intrinsèques de minimisation ;
5. La méthode de rétropropagation d'erreur "BP" nécessite un nombre important d'itération pour atteindre une erreur d'apprentissage suffisamment faible ;
6. La BP est souvent prise au piège dans un minimum local de la fonction d'erreur. Et, elle est très inefficace dans la recherche d'un minimum global si la fonction d'erreur est multimodale et non différentiable ;
7. Les avantages de l'algorithme LM diminuent à mesure que le nombre de paramètres (poids) du réseau augmente.

### **3.9 Entraînement des RNAs par les métaheuristiques**

Comme évoqué précédemment, l'apprentissage supervisé consiste par définition à chercher l'ensemble des poids et biais  $\vec{W}$  d'un réseau de neurones artificiels dans l'objectif de réduire le décalage entre l'ensemble des valeurs observées  $[Y]$  et celles calculées  $[S_r]$  par le réseau. Cette opération s'exprime généralement par l'équation 3.10, rétablie ci-dessous :

$$\text{Minimiser } E(W) = \sum_{p=1}^P \left[ Y_p - S_r(\vec{X}_p, \vec{W}) \right]^2$$

En effet, l'entraînement des RNAs n'est qu'un problème de minimisation de la fonction d'erreur  $E(W)$ . Par autre manière, est un problème d'optimisation dont l'objectif est de trouver un ensemble de poids synaptiques  $\vec{W}$  minimisant la fonction d'erreur (équation 3.10). À partir de là, il y avait l'idée d'utiliser les métaheuristiques tant que méthodes générales d'optimisation pour entraîner les RNAs.

L'avantage principal des algorithmes métaheuristiques évolutionnaires réside au niveau de l'exécution indépendante à la fonction objective à optimiser. Un algorithme de cette famille peut minimiser ou maximiser une fonction sans nécessité d'adaptation de sa structure ou changement de son implémentation (reprogrammation). Le bloc du programme qui doit être changé est uniquement la fonction objective à traiter. L'applicabilité générale de l'approche évolutive permet d'économiser beaucoup d'efforts humains dans le développement de différents algorithmes d'entraînement pour différents types de réseau. Aussi, selon (Yao 1993) les algorithmes évolutionnaires sont particulièrement bons pour traiter de grands espaces complexes qui contiennent de nombreux optima locaux. Ils sont moins susceptibles d'être piégés dans un mauvais minimum local que les algorithmes de recherche traditionnels basés sur le gradient (tels que BP et LM). De plus, ils ne dépendent pas de l'information sur le gradient de sorte qu'ils sont tout à fait adaptés aux problèmes où ces informations ne sont pas disponibles. Ils peuvent même faire face à des problèmes où aucune fonction objective explicite et/ou exacte n'est disponible. Ces fonctionnalités les rendent beaucoup plus robustes que les autres algorithmes de recherche.

Dans les sections suivantes, le mécanisme de l'exploitation des algorithmes évolutionnaires (RGA et HPSOGWO) dans l'apprentissage des RNAs est détaillé.

### **3.9.1 Entraînement des RNAs par l'algorithme génétique RGA**

À cause de la performance prouvée des algorithmes génétiques dans divers domaines (y compris l'entraînement des RNAs, Bledsoe 1962; Bremermann et al. 1962; Goldberg and Holland 1988), la version à codage réel (RGA) déjà présentée est choisie pour entraîner des réseaux de neurones de diverses structures. On veut par structure, le nombre de couches cachées, et le nombre de neurones (nœuds) dans chaque couche. Notre objectif ultime est de prévoir les apports du réservoir Hammam Boughrara. En effet, dans la suite de l'explication, les débits de la série chronologique des apports sont utilisés pour exprimer l'apprentissage des RNAs.

Pour un réseau de neurones donné, l'apprentissage par l'algorithme génétique "RGA" (ou par autre algorithme évolutionnaire) commence par le calcul du nombre de poids et de biais synaptiques  $\vec{W}$  du réseau. Ce nombre sera attribué à la dimension  $D_s$  de l'espace de recherche exploré par l'algorithme. Puis, un ensemble de taille  $N$  de vecteurs "génomés ou chromosomes" est créé, dans lequel chaque vecteur est de longueur  $D_s$  (équation 3.15). Ce qui signifie que chaque chromosome contient l'ensemble des poids et biais de l'RNA à entraîner. Les valeurs des poids synaptiques  $[w_{ij}]$  sont initialisées aléatoirement dans l'intervalle  $[-1, 1]$ .

$$\vec{W}_i = w_i^1, w_i^2, \dots, w_i^j, \dots, w_i^{D_s} \quad \text{tels que : } i = 1, \dots, N \text{ et } j = 1, \dots, D_s \quad (3.15)$$

$$[w_i^j] = \begin{bmatrix} w_1^1 & w_1^2 & \dots & w_1^j & \dots & w_1^{D_s} \\ w_2^1 & w_2^2 & \dots & w_2^j & \dots & w_2^{D_s} \\ \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ w_i^1 & w_i^2 & \dots & w_i^j & \dots & w_i^{D_s} \\ \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ w_N^1 & w_N^2 & \dots & w_N^j & \dots & w_N^{D_s} \end{bmatrix}$$

Le génome le plus performant est celui qui minimise la fonction d'erreur (équation 3.10) mieux que les autres. L'évaluation de performance "fitness" de chaque génome  $\vec{W}_i$  doit passer par l'injection de leurs éléments  $w_i^j$  tant que poids et biais dans l'RNA à entraîner en premier lieu. Puis, en utilisant les données d'entraînement (les entrées observées "mesurées"), les sorties du réseau sont calculées et comparées aux données observées. Par une autre manière plus détaillée, pour chaque couple (entrées-sortie) d'indice  $p$ , l'erreur d'apprentissage  $e_p^i$  est calculée par l'équation 3.16.

$$e_p^i = (Qobs_p - Qc_p^i)^2 \quad (3.16)$$

Avec,  $Qobs_p$  est la valeur de débit observé standardisée d'indice  $p$ , et  $Qc_p^i$  est le débit calculé par le RNA en utilisant les poids synaptiques  $\vec{W}_i$  du chromosome d'indice  $i$ .

À la fin, l'erreur totale d'apprentissage  $E_i$  correspondant au chromosome d'indice  $i$  est calculée comme suit (équation 3.17) :

$$E_i = \sum_{p=1}^P e_p^i = \sum_{p=1}^P (Qobs_p - Qc_p^i)^2 \quad (3.17)$$

L'erreur totale  $E_i$  correspondant à la performance "ou fitness" du chromosome  $i$ , est évaluée par l'ensemble des données d'entraînement de  $p = 1, \dots, P$ . Cette étape exprime l'évaluation d'un chromosome au moyen de la fonction objectif.

L'étape suivante consiste à classer les chromosomes en ordre croissant en fonction de l'erreur totale d'apprentissage correspondante  $E_i$ . Le meilleur chromosome  $\vec{W}_i$  est celui présentant la plus faible erreur d'apprentissage. Si un chromosome prouve une erreur d'apprentissage inférieure ou égale à celle exigée (c-à-d., lorsque  $E_i \leq E_{app}$ ), le processus itératif s'arrête, et le RNA est considéré entraîné dans ce cas. Dans le cas contraire (c-à-d., lorsque  $E_i > E_{app}$ ,  $\forall i = 1, \dots, N$ ), l'algorithme RGA s'exécute d'une manière ordinaire. En effet, les opérations de sélection, croisement et de mutation des chromosomes s'effectuent selon les approches adoptées. Dans la version RGA utilisée, l'opération de sélection est faite par élitisme, qui consiste à choisir les meilleurs chromosomes d'une itération "génération" donnée pour la

suite de la procédure itérative. Le croisement et la mutation sont réalisés conformément aux approches adoptées et expliquées (point aléatoire unique "random single-point"), avec les probabilités de 0.75 et 0.25 respectivement. Aussi, lors de la mutation, les poids sont changés par des valeurs appartenant à l'intervalle  $[-0.5, 0.5]$ .

Finalement, si la condition d'arrêt n'est pas réalisée, le processus itératif se termine lorsqu'il achève le nombre maximal d'itérations  $K_{max}$  fixé au départ. Le déroulement de l'apprentissage par l'algorithme RGA est détaillé dans la figure 4 (Annexes).

### **3.9.2 Entraînement des RNA par l'algorithme HPSOGWO**

Dans cette thèse, on présente un essai d'entraînement des réseaux de neurones artificiels par l'algorithme hybride d'essaim de particules et l'optimiseur de Loups Gris "HPSOGWO". À la limite de notre connaissance, aucun travail n'a été publié dans ce contexte.

De même que l'RGA (ou autre algorithme évolutionnaire), la reimplémentation de cet algorithme n'est pas nécessaire pour entraîner un RNA. Dans ce cas, le changement du scripte (le programme) de l'algorithme ne touche que la définition de la fonction objective. Le reste de l'algorithme s'exécute d'une manière ordinaire.

Au départ, le nombre  $D_s$  de poids synaptiques et de biais doit être calculé en fonction de la structure et la taille de réseau de neurones à entraîner. Ensuite,  $N$  vecteurs ou "agents" (ou encore "loups" selon l'appellation de l'algorithme)  $\vec{W}_i$ , ( $i = 1, \dots, N$ ) doivent être créés et placés aléatoirement dans l'espace de recherche de dimension  $D_s$ , conformément à l'équation 3.15. Cette étape constitue l'initialisation des agents de recherche de l'algorithme.

L'algorithme évalue les agents de recherche via le réseau de neurones à entraîner et au moyen des données d'entraînement. Les éléments  $w_{ij}$  d'un vecteur  $\vec{W}_i$  sont injectés dans le réseau comme des poids synaptiques. Puis, au moyen des données d'apprentissage et les équations 3.16 et 3.17, les erreurs élémentaires  $e_p^i$ , et l'erreur totale  $E_i$  sont estimées respectivement. À ce niveau, la procédure itérative est peut-être arrêtée si l'erreur d'apprentissage permise  $E_{app}$  est atteinte. Dans le cas contraire, le traitement continue.

Par la suite, le déroulement de l'algorithme continue d'une façon habituelle conformément à son processus décrit dans la section 2.6.4.1 (Chapitre 2), avec  $\vec{W}_i = \vec{X}_i$ . Néanmoins, il peut être arrêté si le nombre maximal d'itérations  $K_{max}$  est épuisé en évitant les boucles infinies même si l'erreur d'apprentissage  $E_{app}$  n'est pas atteinte.

La figure 5 (Annexes) représente l'organigramme de déroulement de l'apprentissage par l'algorithme hybride "HPSOGWO".

### **3.10 Conclusion**

Ce chapitre a été consacré à la présentation des réseaux de neurones artificiels, utilisés fréquemment pour imiter le comportement des systèmes complexes y compris les bassins hydrographiques.

Le neurone formel sert à la base de l'architecture multicouche des réseaux de neurones artificiels, a été détaillé avec son mode de fonctionnement. Encore, une large partie du chapitre a été réservée à l'apprentissage supervisé des réseaux de neurones, dans laquelle quatre (04) méthodes d'apprentissage sont exposées. Les deux (02) premières sont des méthodes classiques, très utilisées et devenues un standard dans la matière. Elles sont la rétropropagation du gradient de l'erreur (Backpropagation, "BP") et Levenberg-Marquardt "LM". Les deux (02) autres méthodes sont des métaheuristiques évolutionnaires : une version des algorithmes génétiques à codage réel "RGA", et l'algorithme hybride "HPSOGWO". La manière de profiter les capacités génériques d'optimisation de ces dernières méthodes dans le but d'entraîner les RNA a été décrite.

La finalité de ce chapitre été d'exposer les réseaux de neurones artificiels et quelques méthodes d'entraînement, servent comme outil de prévision des apports des barrages-réservoirs. Sans oublier que l'opération de prévision constitue une pierre de coin dans l'optimisation de la gestion des réservoirs dans un avenir incertain. Les résultats de comparaison entre ces méthodes d'apprentissage appliquées sur le cas d'étude (réservoir Hammam Boughrara) sont présentés dans le chapitre 7.

# Chapitre 4

## Les réseaux de neurones artificiels évolutifs

### 4.1 Introduction

Les réseaux de neurones artificiels sont tellement puissants ont séduit une communauté scientifique assez large à les explorer, développer et à les utiliser. En conséquence, ces dernières décennies, une grande variété de problèmes a été résolue au moyen de ces modèles mathématiques. Dans le chapitre précédent, nous avons abordé l'apprentissage supervisé au moyen des méthodes classiques, notamment la rétropropagation du gradient de l'erreur et Lvenberg-Marquardt, ainsi quelques métaheuristiques évolutives telles que l'algorithme génétique à codage réel et l'algorithme hybride d'essaim de particules et l'optimiseur de Loups Gris. C'était toujours dans le but de fixer les poids et les biais intrinsèques d'un RNA et accomplir son apprentissage.

Toujours dans le cadre de chercher des outils permettant de concrétiser la première phase de l'optimisation stochastique de la gestion opérationnelle d'un réservoir. La phase qui consiste à générer de multiples scénarios d'apports, et qui exige des modèles de prévision assurant certaine fiabilité. Les réseaux de neurones artificiels exposés dans le chapitre précédent, sont des modèles de "boite noire" exigeant un choix judicieux de leurs paramètres corrigés (poids synaptiques et biais) que spécifiques de structure (nombre de noeuds à l'entrée, nombre de couches et neurones cachés, fonction d'activation, ... etc) pour produire des résultats de qualité réclamée.

Malheureusement, les méthodes d'apprentissage des RNAs ne permettent de régler que les poids et biais synaptiques. Les paramètres de structure sont le plus souvent fixés manuellement via une procédure d'essai-erreur, malgré qu'ils peuvent impacter considérablement

la qualité de résultats.

Ce chapitre porte sur la problématique de recherche d'un réseau de neurones "bon", proche-optimal ou quasi-optimal, sous la limite que la recherche d'un RNA optimal global (c-à-d., le meilleur "avec évidence" parmi tous les RNAs possibles) implique l'exploration d'un nombre de possibilités infini ou quasi-infini. Dans cet objectif, le paramétrage des structures des RNAs est fait d'une manière automatique au lieu manuelle à travers l'exploitation des algorithmes métaheuristiques dans l'ambition d'améliorer la performance des RNAs. Cette classe des RNAs est appelée "réseaux de neurones artificiels évolutionnaires, Evolutionary Artificial neural Networks (EANNs)".

Le chapitre est organisé en sections suivantes :

- Problématique : dans laquelle le défi de paramétrage des RNAs est exposé ;
- Recherche d'un RNA proche-optimal : dans cette partie une étude bibliographique succincte de la problématique en question est déployée ;
- Les métaheuristiques pour la recherche d'un EANN proche-optimal : dans laquelle l'approche d'exploitation des algorithmes évolutionnaires pour traiter la problématique en question est exposée.

## **4.2 Problématique**

Même avec le succès largement prouvé dans la littérature et par les applications dans les domaines commerciaux et industriels, la recherche d'un réseau de neurones artificiels (RNA) bien adapté (optimal ou quasi-optimal) à un problème spécifique reste un défi (Almeida and Ludermir 2008b).

Un RNA quasi-optimal se caractérise par le choix des paramètres spécifiques (structure du réseau, nombre de neurones, la fonction d'activation à utiliser, ... etc) et corrigés (poids synaptiques et biais) pour un problème donné, produisant ainsi une performance satisfaisante (Almeida and Ludermir 2008a).

Face à cette situation, on se trouve devant plusieurs alternatives :

- Soit faire un choix trivial avec "essai et erreur", ou faire un balayage partiel de l'espace de recherche qui ne garantissent pas la qualité supérieure des résultats. De plus, ce mode de réglage (ou de choix) manuel des paramètres pour résoudre certains problèmes est considéré comme une tâche fastidieuse, moins productive et sujette à l'erreur (Almeida and Ludermir 2008b ; Kiranyaz et al. 2009 ; Panahi et al. 2019).

- Explorer toutes les combinaisons possibles par balayage de l'espace de recherche. Malheureusement, cette opération est pratiquement "impossible", car elle conduit à l'évaluation d'un nombre exponentiel de paramètres, et par conséquent, un temps de calcul excessif et une consommation abusive des ressources de l'ordinateur.
- Soit développer des outils de "recherche intelligente" permettant d'explorer un nombre prometteur de possibilités avec une consommation raisonnable du temps et de ressources de computation.

De plus que la construction de configurations quasi optimales des RNAs implique des difficultés telles que le nombre exponentiel de paramètres qui doivent être ajustés ; elle requiert dans certains cas, une connaissance préalable du domaine problématique et de l'exploitation et la présence d'un expert lorsqu'une telle connaissance fait défaut (Almeida and Ludermir 2008a). Selon (Yao 1993) la conception des architectures des RNAs reste encore un travail d'expert humain. Cela dépend fortement de la maîtrise de l'expert et d'un processus fastidieux d'essais et d'erreurs.

### **4.3 Recherche d'un RNA proche-optimal**

En commençant par l'évidence qu'il existe au moins un (01) réseau de neurones artificiels qui peut traiter un problème bien déterminé mieux que les autres RNAs. Ce réseau représente le RNA optimum global dans ce cas. Sous l'effet que la recherche de ce réseau de neurones est une opération très couteuse, on se limite à un RNA dite proche-optimal ou "quasi-optimal" présentant une performance suffisante ou requise, ou encore "acceptable".

La recherche d'un tel RNA conduit directement à la résolution d'un problème d'optimisation, dont l'objectif principal est de réduire l'erreur totale dans la phase d'apprentissage sans perdre de capacité de généralisation à l'effet de sur-apprentissage ("Overfitting" en anglais).

Les deux majeures applications des algorithmes évolutionnaires sont l'optimisation et l'apprentissage des réseaux de neurones artificiels (Yao 1993). Selon (Yao 1993 ; Almeida and Ludermir 2008a ; Mirjalili et al. 2012) il existe trois approches majeures d'utilisation d'un algorithme d'optimisation pour l'entraînement des réseaux de neurones artificiels (Figure 4.1) :

1. La première approche consiste à utiliser un algorithme d'optimisation pour trouver une combinaison de poids et biais fournissant l'erreur minimale pour un RNA. Dans cette approche, les méthodes basées sur le gradient décent (la rétropropagation, Gauss, Newton, Levenberg-marquardt, ..., etc.) et les métaheuristiques peuvent être utilisées.

2. Concernant la deuxième approche, un algorithme d'optimisation est utilisé comme un moyen de trouver une structure appropriée d'un RNA dans un problème particulier. On sait que l'architecture d'un RNA détermine sa capacité de traitement de l'information. Les métaheuristiques et les méthodes dites de "construction" ou de "destruction" sont utilisées (Yao 1993). Ces dernières consistent à bâtir ou à détruire un RNA au fur et à mesure avec l'évaluation de sa performance.
3. La dernière approche consiste à l'utiliser pour ajuster les paramètres d'un algorithme d'apprentissage tels que le taux d'apprentissage, la fonction d'activation et ses paramètres ... etc. L'algorithme d'optimisation peut être couplé dans ce cas avec un autre basé sur le gradient (méthode de rétropropagation, Gauss-Newton, Levenberg-Marquardt, ... etc).

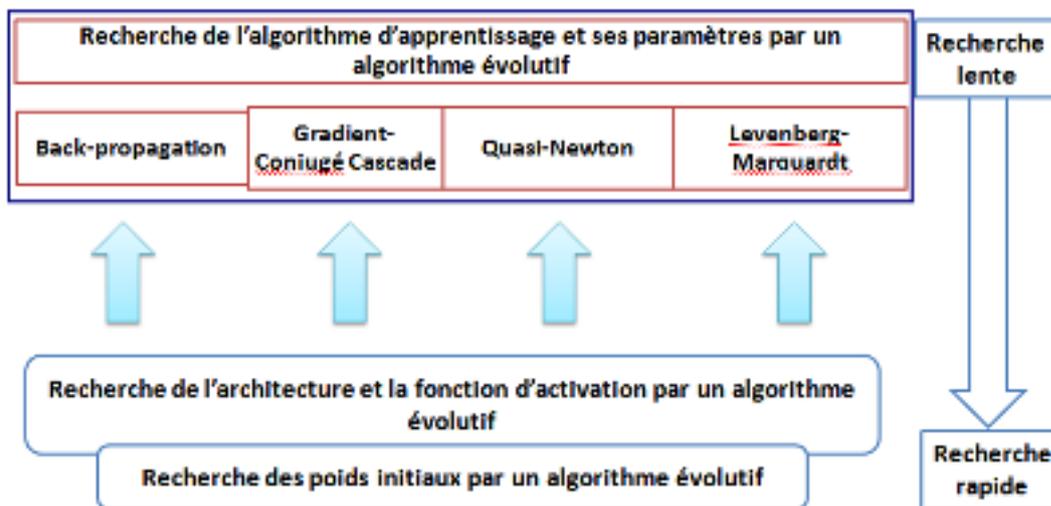


FIGURE 4.1 – Couches de recherche évolutionnaire des paramètres des RNAs (adaptée d'après Almeida and Ludermir 2008a)

Selon (Yao 1993), les réseaux de neurones artificiels issus de l'optimisation par des algorithmes métaheuristiques sont classés dans la catégorie de "réseaux de neurones artificiels évolutionnaires (Evolutionary artificial neural networks, EANNs)".

### 4.3.1 Les métaheuristiques pour la recherche d'un RNA proche-optimal

Dans cette étude, nous optons pour une approche combinée en utilisant deux (02) algorithmes d'optimisation à la fois, servant respectivement à :

1. Le premier algorithme est un métaheuristique sert à *l'optimisation* :
  - (a) Chercher les meilleurs paramètres de l'algorithme d'apprentissage utilisé, y compris le taux d'apprentissage, le nombre maximal d'itérations à appliquer,

- l'erreur d'apprentissage, ... etc ;
- (b) Chercher la fonction d'activation qui convient parmi les fonctions (sigmoïde et sigmoïde bipolaire, ...) avec ses paramètres ( $\alpha_{activ}$ , ...);
  - (c) Chercher la structure du réseau, y compris le nombre de couches cachées et le nombre de neurones de chaque couche.
2. Le deuxième algorithme sert à *l'apprentissage* des réseaux RNAs créés par le premier, c-à-d., le paramétrage des poids et biais synaptiques. Ce 2<sup>ème</sup> algorithme peut être une métaheuristique ou autre basé sur le gradient (rétropropagation ou Levenberg-Marquardt, ...).

La figure 4.2 montre un schéma explicatif de l'approche adoptée.

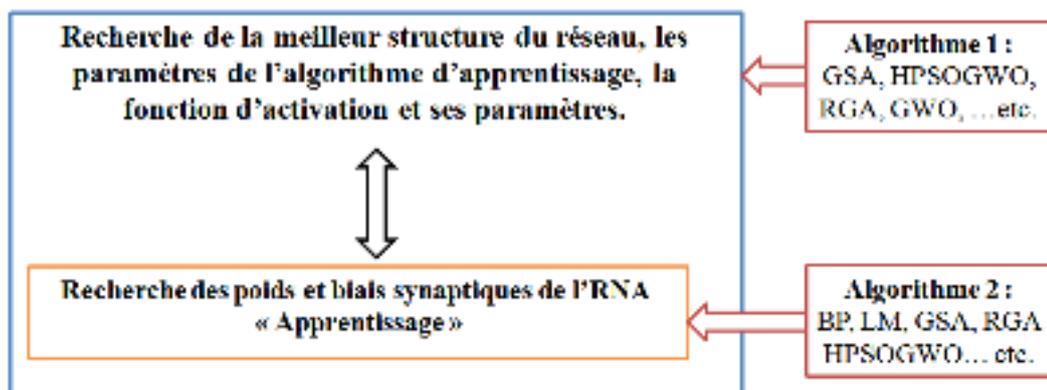


FIGURE 4.2 – Approche adoptée de recherche d'un RNA proche-optimal

L'avantage principal des métaheuristicques réside au niveau de la flexibilité quand elles traitent des problèmes de natures différentes. C'est pourquoi que la modification ou la réimplémentation de ce genre d'algorithmes n'est pas nécessaire.

La figure 4.3 montre la procédure générale de d'exploitation d'un algorithme métaheuristique à base de population pour chercher un réseau de neurones artificiels quasi-optimal.

L'implémentation des réseaux de neurones artificiels ainsi les méthodes classiques d'apprentissage (rétropropagation du gradient d'erreur "BP" et Levenberg-Marquardt "LM") utilisés dans cette étude, sont de la plateforme d'apprentissage machine "Machine-Learning" libre et ouverte "Open-source" *Accord.NET Framework* (Souza et al. 2014) disponible sous les liens : <http://accord-framework.net/>, ou <https://github.com/accord-net/framework>.

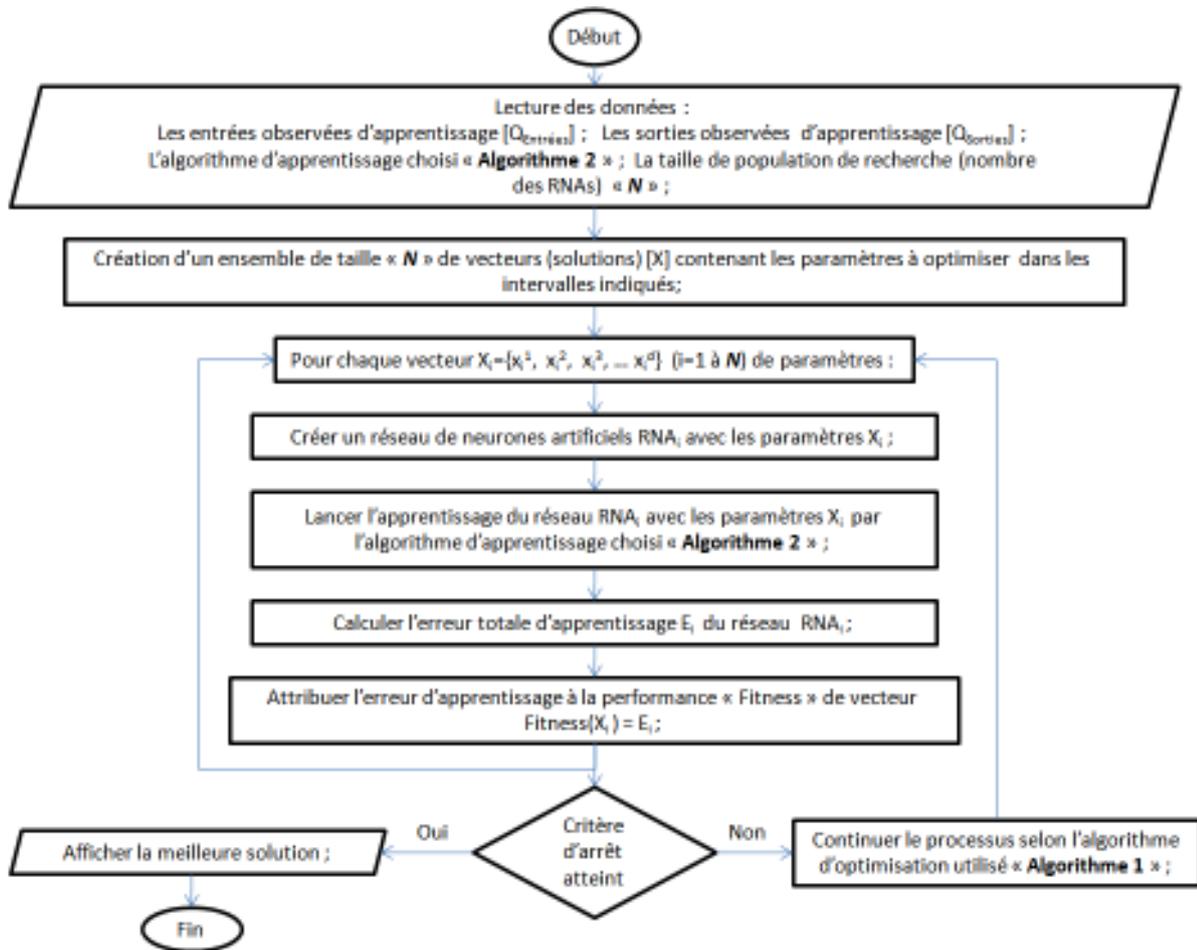


FIGURE 4.3 – Organigramme de recherche d'un RNA proche-optimal selon l'approche adoptée

### 4.3.2 Fonction objectif

Dans cette étude et à fin de garder la généralisation des réseaux de neurones issus de l'opération d'optimisation, on limite le nombre de couches cachées à cinq (05), c-à-d., des réseaux de neurones à sept (7) couches en totalité. De même pour le nombre de neurones cachés dans chaque couche, qui sont arrêtés à quinze (15). La fonction objective d'apprentissage est conservée en principe comme fonction objectif d'optimisation. Elle vise de minimiser le décalage entre les valeurs observées (mesurées) et celles sorties d'un réseau de neurones. A fin de forcer l'algorithme d'optimisation à explorer plus les réseaux de neurones artificiels à petites tailles, une pénalisation est appliquée sur la performance "l'erreur d'apprentissage" en fonction du nombre de couches cachées. Selon (Yao 1993) cette approche de pénalisation est utilisée pour diminuer la complexité des EANNs et améliorer leurs capacités de généralisation. L'équation 4.1 est utilisée pour concrétiser cette fonction objectif.

$$Fobj_2 = Minimiser(E, N_c) = (1 + \frac{N_c}{100})E = (1 + \frac{N_c}{100}) \sum_{i=1}^P (y_i^{obs} - y_i^{clac})^2 \quad (4.1)$$

Tels que :

- $E$  : l'erreur globale d'apprentissage ;
- $N_c$  : nombre de couches cachées de l'RNA ;
- $y^{obs}$  : sortie observée du système physique ;
- $y^{clac}$  : sortie calculée par le réseau de neurones artificiels.

### 4.3.3 Codage des solutions

Comme évoqué auparavant, l'approche décrite ici, consiste à utiliser deux (02) algorithmes en même temps. Le premier est une métaheuristique évolutionnaire à base de population. Il sert à paramétrer l'algorithme d'apprentissage, choisir la fonction d'activation et son (ou ses) paramètre(s), et finalement choisir le nombre de couches cachées et le nombre de neurones de chaque couche. En conséquence, la population résultante est un groupe de vecteurs comprenant l'ensemble de paramètres. Le codage des solutions est présenté dans les figures 4.4 (ci-dessous) et 6 (Annexes). La longueur des vecteurs n'est pas constante car le nombre de paramètres des algorithmes d'apprentissage se diffère l'un de l'autre, le codage des solutions diffère en longueur de même.

(1,2, 3)	1	ID Fonction d'activation $f(\cdot)$	(1,2, 3)	1	ID Fonction d'activation $f(\cdot)$
[0,1, 5]	2	Valeur de coefficient $\alpha_{\text{act}}$	[0,1, 5]	2	Valeur de coefficient $\alpha_{\text{act}}$
[0,1, 1]	3	Taux d'apprentissage $\rho$	[0,1, 1]	3	Taux d'apprentissage $\rho$
[0,005, 0,05]	4	Erreur d'apprentissage $E_{\text{app}}$	[0,005, 0,05]	4	Erreur d'apprentissage $E_{\text{app}}$
[1000, 20000]	5	Nombre maximal d'itérations $K_{\text{iter}}$	[10, 99]	5	Nombre maximal d'itérations $K_{\text{iter}}$
[1, 5]	6	Nombre de couches cachées $N_c$	[1, 5]	6	Nombre de couches cachées $N_c$
[1, 15]	7	Nombre de neurones de la couche cachée $N^{\circ 1}$	[1, 15]	7	Nombre de neurones de la couche cachée $N^{\circ 1}$
[1, 15]	8	Nombre de neurones de la couche cachée $N^{\circ 2}$	[1, 15]	8	Nombre de neurones de la couche cachée $N^{\circ 2}$
[1, 15]	9	Nombre de neurones de la couche cachée $N^{\circ 3}$	[1, 15]	9	Nombre de neurones de la couche cachée $N^{\circ 3}$
[1, 15]	10	Nombre de neurones de la couche cachée $N^{\circ 4}$	[1, 15]	10	Nombre de neurones de la couche cachée $N^{\circ 4}$
[1, 15]	11	Nombre de neurones de la couche cachée $N^{\circ 5}$	[1, 15]	11	Nombre de neurones de la couche cachée $N^{\circ 5}$

(a)
(b)

FIGURE 4.4 – Codage des solutions et intervalles de définition des variables selon l'algorithme d'apprentissage : (a) Back-propagation, (b) Levenberg-Marquardt

Dans ce qui suit de cette étude, les algorithmes utilisés pour réaliser l'opération d'optimisation sont le GSA et le HPSOGWO. Cependant, seules les méthodes classiques (BP et LM) qui sont utilisées pour entraîner les réseaux de neurones artificiels. Car, selon les tests effectués, il est trouvé que le couplage de deux (02) algorithmes évolutionnaires (le 1<sup>er</sup> pour l'optimisation et le 2<sup>ème</sup> pour l'apprentissage) implique un temps de traitement assez long, dépassant  $46^H$  pour chaque test. Ce qui dévalorise l'utilisation de ce couplage dans ce cas.

## **4.4 Conclusion**

À travers ce chapitre, nous avons présenté une approche de recherche des réseaux de neurones artificiels proches-optimaux. L'approche consiste à exploiter les métaheuristiques évolutionnaires à base de population tant que boîtes noires et sans modifications de ses structures pour optimiser le choix de trois (03) catégories de paramètres des RNAs, à savoir :

1. La fonction d'activation à utiliser par les RNAs et ses paramètres ( $\alpha_{activ}$  pour la sigmoïde et la bipolaire-sigmoïde);
2. Les paramètres de l'algorithme d'apprentissage "la méthode d'entraînement" (le taux et l'erreur d'apprentissage, ainsi le nombre d'itérations);
3. La structure des réseaux de neurones en termes de nombre de couches cachées et nombre de neurones de chaque couche.

Ces paramètres sont groupés pour former des vecteurs. Ces derniers sont multipliés pour créer une population de "solutions" à améliorer par la suite via un algorithme métaheuristique. À travers le processus d'optimisation, des réseaux de neurones artificiels sont créés et paramétrés par les valeurs de ces vecteurs. L'évaluation de l'efficacité (la performance) d'un RNA est effectuée après l'apprentissage de ceci. À ce niveau, un algorithme d'apprentissage classique (Rétropropagation du gradient d'erreur, Levenberg-Marquardt, ...) ou un métaheuristique est peut-être utilisé.

L'examen de l'utilité de cette approche est opéré sur la prévision des apports du réservoir Hammam Boughrara, et fait l'objet du chapitre 8.

## **Deuxième partie**

# **Optimisation de la gestion opérationnelle d'un réservoir à usages multiples : Cas du Barrage Hammam Boughrara**



# Chapitre 5

## Présentation de la zone d'étude

### 5.1 Introduction

La gestion des ressources en eau souterraines et superficielles ainsi les risques associés aux différents phénomènes hydrologiques (e.g., inondations, sécheresse, érosion, glissement de terrains, ... etc) sont parmi les préoccupations majeures de la société actuellement. Les bassins versants constituent des unités géographiques relativement homogènes favorables à l'étude de plusieurs phénomènes notamment les écoulements de surface, la source principale de l'alimentation des réservoirs naturels et artificiels.

Vu l'importance des ressources en eau qui les offre, le bassin versant de la Tafna est toujours considéré comme étant le château d'eau de toute la région hydrographique de l'Oranie (nord-ouest de l'Algérie). Depuis les années 1930 à l'heure actuelle, pas mal d'ouvrages de mobilisation de la ressource en eau superficielle ont été réalisés, entre autres, les réservoirs de Béni Bahdel, Mefrouche, El-Izdihar "Sidi Abdelli", Hammam Boughrara et Sikkak. L'objectif toujours est, d'une part, répondre à la demande en eau qui ne cesse d'augmenter avec le développement démographique et urbain de la région, et d'autre part, pallier les périodes de sécheresse qui frappent de temps en temps le nord algérien.

Le barrage Hammam Boughrara est le plus important dans l'ouest algérien avec une capacité de  $177Mm^3$  (Million de mètres cubes). Il est construit dans un contexte multi-objectifs, alimentation en eau potable, industrielle, et irrigation. Le réservoir du barrage est alimenté principalement par le sous bassin d'oued Mouilah partagé entre les deux territoires algérien et marocain.

Ce chapitre est consacré premièrement à la présentation grossière du bassin versant de la Tafna, et deuxièmement à la description de sous bassin d'oued Mouilah.

## 5.2 Présentation du bassin versant de la Tafna

### 5.2.1 Situation géographique

Le bassin versant de la Tafna se situe au nord-ouest du territoire algérien. Il appartient à la région hydrographique de l'Oranie-Chott Chergui (Figure 5.1), et selon la classification des bassins versant de l'Agence Nationale des Ressources Hydriques (ANRH), il est indexé par le numéro 16. Le bassin s'étend sur une superficie de  $(7245\text{km}^2)$ , qui représente 77% de surface de la wilaya de Tlemcen, et déborde d'un tiers de sa superficie sur le territoire marocain. Géographiquement, le bassin est délimité par le principal relief des monts de Tlemcen au sud, entre la méditerranée et les hautes plaines oranaises, et relayé à l'Ouest par le moyen Atlas marocain, et à l'Est par les monts de Saida (Figure 5.2). Du sud vers le nord, la Tafna reçoit plusieurs affluents dont les plus importants sont, l'Oued Isser et l'Oued Mouilah, les autres sont moins importants mais permanents et ne tarissent jamais, tels que oued Khemis, Sikkak, et Chouly. D'autres sont temporaires, non alimentés par des sources et ne dépendant que des précipitations. Leurs apports sont toutefois considérables quand ils sont en crue (Ghenim 2008).

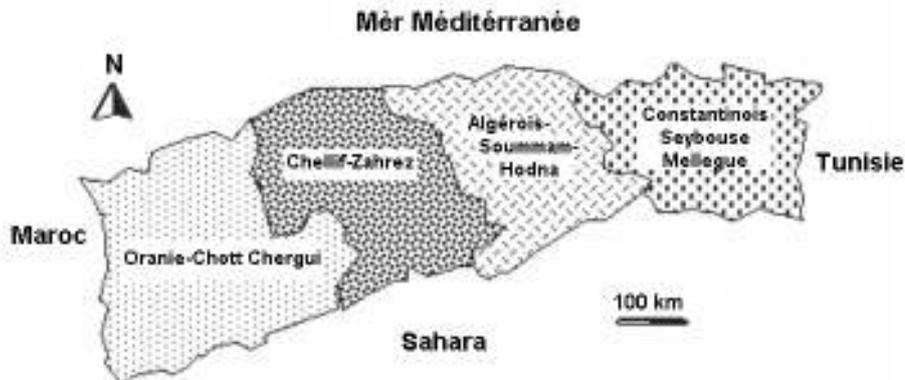


FIGURE 5.1 – Découpage de l'Algérie du nord en quatre régions (Remini 2005)

Le bassin de la Tafna est composé de huit (08) sous bassins à savoir (Figure 5.3) :

1. Sous bassin d'oued Isser (Benskrne) ;
2. Sous bassin d'oued Isser (Remchi) ;
3. Sous bassin d'oued Lakhdar (Chouly) ;
4. Sous bassin d'oued Sekak ;
5. Sous bassin d'oued En nachef ;
6. Sous bassin d'oued Ouardefou ;
7. Sous bassin d'oued Isser Boukiou ;
8. Sous bassin d'oued Mouilah.

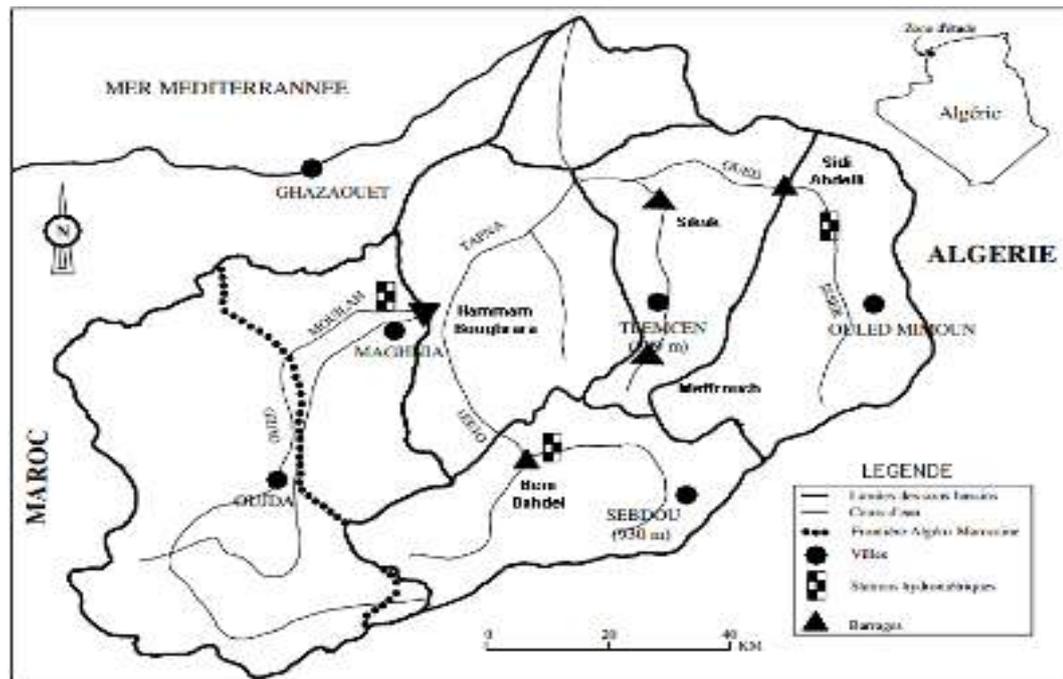


FIGURE 5.2 – Situation géographique du bassin versant de la Tafna (Ghenim 2008)



FIGURE 5.3 – Les sous bassins de la Tafna (Bounoua 2014)

### 5.2.2 Les barrages situés dans le bassin de la Tafna

Plusieurs ouvrages de mobilisation de la ressource en eau superficielle sont réalisés dans le bassin de la Tafna. Les plus importants sont listés ci-dessous. La figure 5.2 montre leurs distributions spatiales sur le bassin.

§1 **Barrage Beni Bahdel** : Réalisé sur la confluence des oueds Tafna et Khémis à 28 Km, au sud-ouest de Tlemcen et mis en eau en 1946. Ce barrage est de capacité de  $63 Mm^3$  et reçoit

de l'eau d'un bassin de superficie de  $1016 \text{ km}^2$ . Conçu initialement pour assurer l'irrigation du périmètre de Maghnia, il a été décidé par la suite de l'utiliser également pour l'alimentation de la ville d'Oran et la production de l'énergie hydroélectrique. Le volume du réservoir fut alors porté à  $63 \text{ Mm}^3$  en élevant la hauteur du barrage (Institut Méditerranéen de l'Eau 2010).

**§2 Barrage Meffrouch :** Situé à environ 4 Km au sud-ouest de la ville de Tlemcen. Ce barrage est construit entre 1957 et 1963 sur l'Oued El-Mefrouche (Sikkak supérieur) drainant un bassin versant de  $90 \text{ km}^2$  environ. Le barrage se caractérise par une capacité de  $15 \text{ Mm}^3$ , destinée à l'alimentation en eau de la ville de Tlemcen et l'irrigation d'un petit périmètre à proximité de la ville. Deux stations de traitement sont implantées en aval du barrage avec des capacités de  $38400 \text{ m}^3/\text{j}$  et  $15840 \text{ m}^3/\text{j}$  respectivement (Bounoua 2014).

**§3 Barrage Sidi Abdelli :** Construit à la wilaya de Tlemcen et mis en exploitation en 1988. Ce barrage est de capacité de  $110 \text{ Mm}^3$  (Institut Méditerranéen de l'Eau 2010), alimenté par un bassin versant de  $1137 \text{ km}^2$ . Il est utilisé actuellement pour alimenter la ville de Sidi Bel Abbes en eau potable et pour soutenir les débits prélevés au niveau de la prise de la Tafna (Bounoua 2014).

**§4 Barrage Hammam Bouhrara :** Le barrage Hammam Bouhrara est un ouvrage stratégique à caractère régional, d'une capacité de  $177 \text{ Mm}^3$ . Il est mis en exploitation la fin de 1998, dont ses objectifs sont l'alimentation en eau potable et l'irrigation.

**§5 Barrage Sikkak :** Situé sur l'Oued Sikkak, à l'aval du barrage Meffrouch (à 1 km environ de la localité d'Ain Youcef). Ce barrage est de capacité de  $27 \text{ Mm}^3$ , alimenté par un bassin versant de  $326 \text{ km}^2$ . Il est mis en exploitation en 2004, et destiné à l'irrigation des périmètres agricoles situés en aval et à l'alimentation du couloir Ain Youcef-Tlemcen (Bounoua 2014).

### **5.3 Le bassin versant d'Oued Mouilah**

Étant donné que le cas d'étude dans ce travail est le barrage Hammam Bouhrara, les prochaines sections seront consacrées à ce barrage et son bassin versant.

### 5.3.1 Localisation géographique de bassin versant d'Oued Mouilah

L'Oued Mouilah débute au sud-ouest de la ville d'Oujda (Maroc) sous le nom d'Oued Bounaim. Ensuite, il franchit le territoire algérien au niveau de la commune de Maghnia, à l'extrême ouest de la wilaya de Tlemcen pour croiser l'Oued Tafna en formant le bassin versant d'Oued Mouilah. La confluence d'Oued Mouilah avec celui de Tafna se situe à environ 13 km sud de la ville de Maghnia, à 285 m d'altitude dans les plaines portant le nom de la ville "i.e, plaines de Maghnia".

Le bassin versant d'Oued Mouilah s'étale sur une superficie de  $2650\text{km}^2$ , limitée par un périmètre de  $230\text{km}$  environ. Une partie majoritaire du bassin se trouve sur le territoire marocain. La figure (5.4) représente la localisation géographique du bassin. Dès la fin de l'année 1998, le bassin est régularisé par la construction du barrage Hammam Boughrara (capacité de  $177\text{Mm}^3$ ).

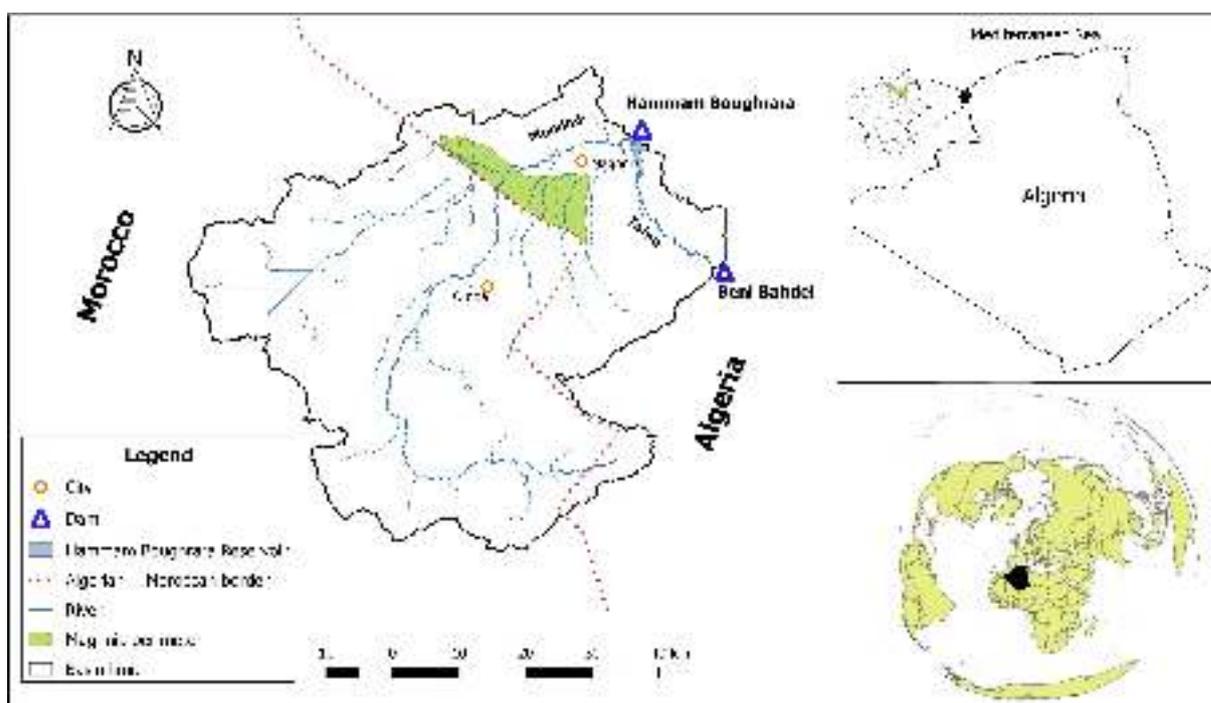


FIGURE 5.4 – Situation géographique du bassin versant d'Oued Mouilah (Dahmani and Yebdri 2020)

### 5.3.2 Morphologie de bassin versant d'Oued Mouilah

Le bassin versant d'Oued Mouilah est constitué par des zones très hétérogènes. On trouve au sud les plaines de Maghnia et des Angad qui représentent des vallées encaissées entre les monts de Tlemcen au sud et les monts des Traras au nord-ouest. Le relief est assez varié constitué des zones montagneuses, de plaines et de vallées (Bounoua 2014).

Les caractéristiques de forme du bassin versant pouvant être résumées en ce qui suit.

### 5.3.2.1 Indice de compacité de Gravelius

La forme d'un bassin versant influence sur l'écoulement et sur l'allure de l'hydrogramme de crue à l'exutoire du bassin. En effet, il existe différents indices morphologiques permettant de caractériser la forme, et aussi de comparer les bassins versants entre eux. Parmi ces indices, on trouve celui de *Gravelius* " $K_c$ " qui caractérise la compacité du bassin versant. Cet indice se définit comme suit :

$$K_c = \frac{P}{P_a} = \frac{P}{2\sqrt{\pi S}} \cong 0.282 \frac{P}{\sqrt{S}} \quad (5.1)$$

Tels que :

- $P$  : périmètre du bassin versant ( $km$ );
- $P_a$  : périmètre du cercle ( $km$ ) ayant même surface que le bassin;
- $S$  : surface du bassin ( $km^2$ ).

Pour un bassin circulaire l'indice de Gravelius  $K_c$  est égal à 1 et augmente autant que la compacité diminue. Si le bassin est carré, le  $K_c$  est égale à 1.128, et peut atteindre la valeur 3 pour les bassins très allongés. Les paramètres de forme de bassin d'Oued Mouilah sont données dans le tableau 5.1. L'indice de Gravelius indique une forme relativement allongée du bassin.

TABLE 5.1 – Paramètres de forme de bassin versant d'Oued Mouilah

Bassin versant	Surface ( $Km^2$ )	Périmètre ( $Km$ )	Indice de compacité ( $K_c$ )	Forme
Oued Mouilah	2650	230	1.25	Allongée

### 5.3.2.2 Rectangle équivalent :

Ce paramètre caractérise l'influence de la forme de bassin sur l'écoulement. Il s'agit d'une transformation purement géométrique dans laquelle le contour du bassin devient un rectangle de même périmètre, les courbes de niveau sont des droites parallèles à la largeur du rectangle, et l'exutoire est l'un des petits côtés du rectangle. La longueur et la largeur du rectangle sont données par les formules (5.2) et (5.3) respectivement.

$$L_r = \frac{K_c \sqrt{S}}{1.12} \left( 1 + \sqrt{1 - \left( \frac{1.128}{K_c} \right)^2} \right) \quad (5.2)$$

$$l_r = \frac{K_c \sqrt{S}}{1.12} \left(1 - \sqrt{1 - \left(\frac{1.128}{K_c}\right)^2}\right) \quad (5.3)$$

Tels que :

- $L_r$  : longueur du rectangle équivalent ( $Km$ );
- $l_r$  : largeur du rectangle ( $Km$ );
- $S$  : superficie du bassin versant ( $Km^2$ );
- $K_c$  : indice de compacité.

Les dimensions du rectangle équivalent sont données dans le tableau 5.2.

TABLE 5.2 – Paramètres de forme et dimensions du rectangle équivalent

Sous bassin	Superficie ( $Km^2$ )	Périmètre ( $Km$ )	$K_c$	$L_r$ ( $Km$ )	$l_r$ ( $Km$ )
Oued Mouilah	2650	230	1.25	82.96	31.94

### 5.3.2.3 Le relief

Le relief est un facteur primordial affectant la majorité des phénomènes naturels. En effet, l'hydrologie et la climatologie d'un bassin versant sont directement liées à l'altitude et sa répartition.

**§1 Courbe hypsométrique :** La représentation de la courbe hypsométrique (donnée aussi sous forme d'histogramme) est la courbe des fréquences altimétriques exprimant la distribution des altitudes en fonction de surface du bassin. En effet, elle exprime ainsi la superficie du bassin ou le pourcentage de superficie, au-delà d'une certaine altitude. Les courbes hypsométriques demeurent un outil pratique pour comparer plusieurs bassins entre eux, et même les diverses sections d'un seul bassin. Elles peuvent en outre servir à la détermination de la pluie moyenne sur un bassin versant et donnent des indications quant au comportement hydrologique et hydraulique du bassin et de son système de drainage (Bounoua 2014). Les altitudes du bassin varient entre 285m à 1460m avec une altitude moyenne de 682.79m environ. Le thalweg principal est de 124km de longueur. La figure (5.5) représentent la courbe hypsométrique du bassin versant d'Oued Mouilah, tandis que la figure (5.6) démontre la distribution des altitudes sur la surface du bassin.

La courbe hypsométrique montre que les surfaces sont accentuées vers les basses altitudes reflétant la présence des plaines de Maghnia, et de même vers les hautes altitudes

reflétant la présence des chaînes montagneuses des Angades au nord-ouest et de Tlemcen au sud.

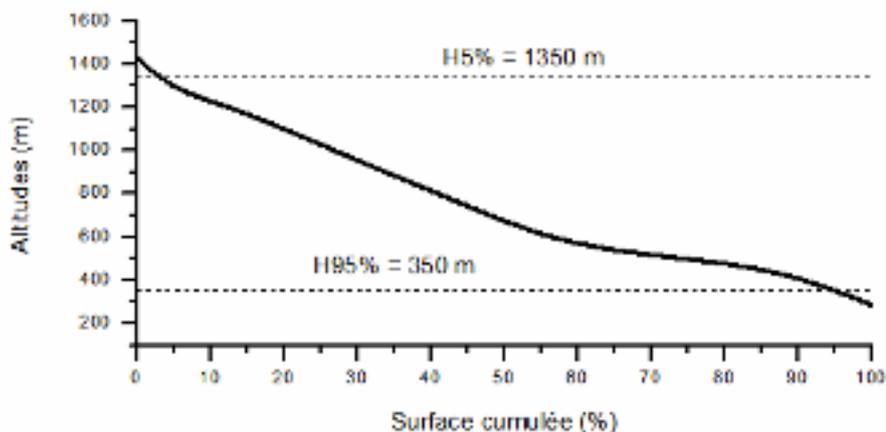


FIGURE 5.5 – La courbe hypsométrique du bassin versant d’Oued Mouilah

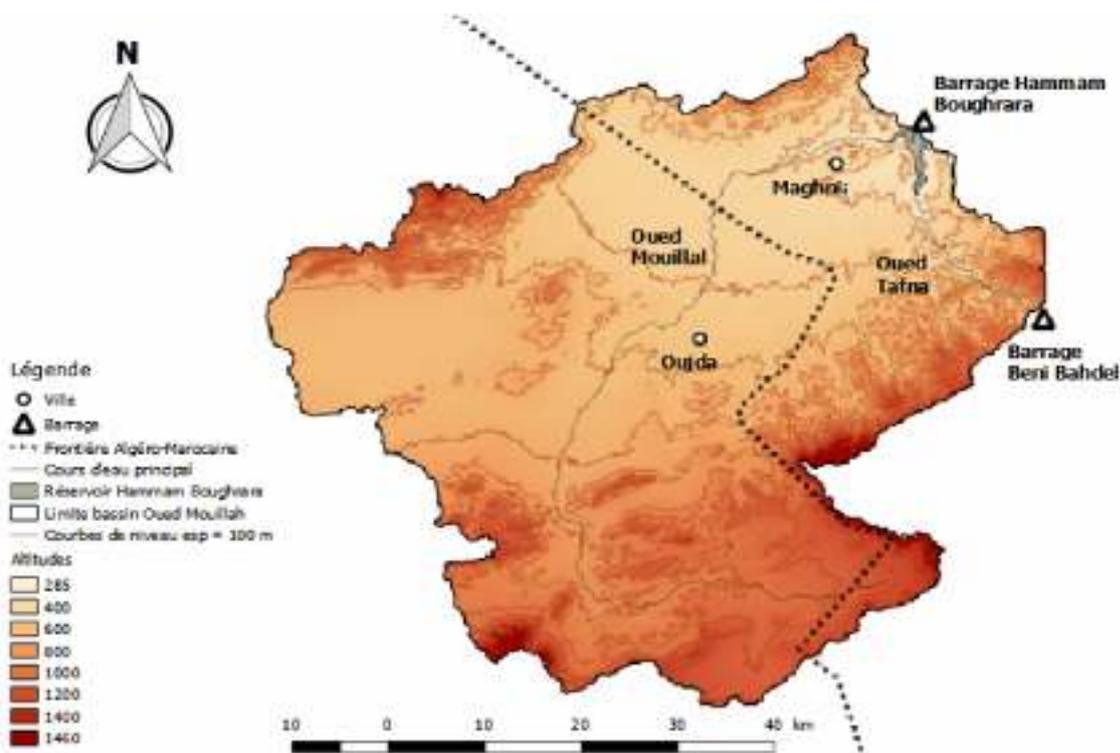


FIGURE 5.6 – Reliefs du bassin d’Oued Mouilah

**§2 Les pentes de terrain :** À partir les figures (5.5 et 5.6), il est claire que l’accentuation de la pente de la courbe vers les basses altitudes indique que la partie aval du bassin du Mouilah est bien occupée par la plaine importante de Maghnia et Angades. Vers les hautes altitudes, le relief accentué au nord-ouest et au sud du bassin est bien mis en évidence par la forme de la courbe hypsométrique.

Les pentes sont en général très accentuées aux zones montagneuses, dépassant les 20%, et plus douces (entre 0 et 10%) de part et d'autre du parcours du cours d'eau et au niveau de la plaine de Maghnia. La figure 5.7 montre la distribution des pentes sur la superficie du bassin.

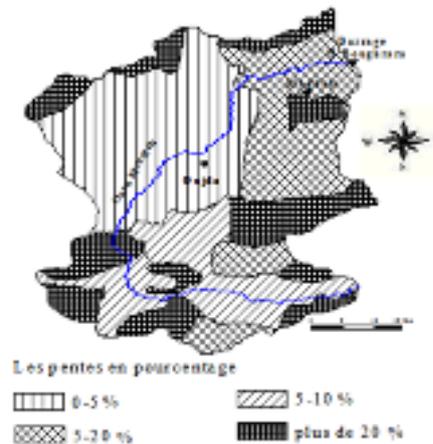


FIGURE 5.7 – Carte de distribution des pentes du bassin versant d'Oued Mouilah (Bounoua 2014)

Les tableaux 5.3 et 5.4 résument respectivement les indices de pente et les caractéristiques morphologiques du bassin.

TABLE 5.3 – Indices de pente relatifs au bassin versant d'oued Mouilah

Indice de pente	Formule	Valeur -bassin O.Mouilah-	Observation
Roche <sup>a</sup>	$I_p = \frac{1}{\sqrt{L_r}} \sum_{i=1}^n \sqrt{S_i(a_i - a_{i-1})}$	1.15	-
Globale <sup>b</sup>	$I_g = \frac{D}{L} = \frac{(H_{5\%} - H_{95\%})}{L_r}$	12.05m/km	relief modéré (R4)
Moyenne <sup>c</sup>	$I_m = \frac{\Delta H}{L_r} = \frac{H_{max} - H_{min}}{L_r}$	1.416%	-
Déniv. Spécifique <sup>d</sup>	$D_s = I_g \sqrt{S}$	620.312	Relief très fort R7

<sup>a</sup>  $a_i$  : cote de courbe de niveau d'indice  $i$ .  $S_i$  : surface du bassin versant comprise entre les courbes de niveau  $i$  et  $i - 1$ .  $L_r$  : longueur du rectangle équivalent égale à 82.96 Km.

<sup>b</sup> D'après la classification de l'IRD (Institut de recherche pour le développement, France) (ex, ORSTOM), donnée au tableau 2 (Annexes).

<sup>c</sup>  $H_{min} = 285m$ ,  $H_{max} = 1460m$ .

<sup>d</sup> Dénivelée spécifique selon la classification IRD (tableau (3, Annexes).

TABLE 5.4 – Caractéristiques morphologiques du bassin d'Oued Mouilah

Paramètre	Valeur	Paramètre	Valeur	Paramètre	Valeur
$S$ (Km <sup>2</sup> )	2650	$Kc$	1.25	$H5\%$ (m)	1350
$P$ (km)	230	$H95\%$ (m)	350	$I_p$	1.15
$L_r$ (Km)	82.96	$l_r$ (Km)	31.94	Altitude moy (m)	682.79
$I_m$ (%)	1.42	$I_g$ (m/Km)	12.05 (modéré)	$D_s$ (m)	620.312 (très fort)

### 5.3.3 Facteurs physiographiques du bassin versant

#### 5.3.3.1 Densité de drainage

Elle est définie comme le rapport entre la longueur totale de tous les cours d'eau et la superficie du bassin versant (Équation 5.4).

$$D_d = \frac{\sum L_i}{S} \quad (5.4)$$

Tels que :

- $D_d$  : la densité de drainage  $km/km^2$ ;
- $L_i$  : les longueurs de tous les thalwegs y compris le thalweg principal ;
- $S$  : la superficie du bassin versant ( $km^2$ ).

Les caractéristiques physiographiques du bassin sont données dans le tableau (5.5).

#### 5.3.3.2 Rapport de confluence

Ce rapport s'exprime par l'équation suivante :

$$R_c = \frac{N_i}{N_{i+1}} \quad (5.5)$$

Où,  $N_i$  est le nombre des cours d'eau d'ordre  $i$ ,  $i = 1, 2, \dots$ . Si on admet que, pour un bassin parfaitement organisé  $R_c = 2$ ; les sous bassins de la Tafna sont mal hiérarchisés (Bounoua 2014).

#### 5.3.3.3 Rapport des longueurs

Ce rapport s'exprime par l'équation suivante :

$$R_L = \frac{L_{i+1}}{L_i} \quad (5.6)$$

Où,  $L_i$  est la longueur de cours d'eau d'ordre  $i$ . Les valeurs de ce rapport sont comprises entre 1.82 et 3.16, elles sont approximatives pour la plupart des cours d'eau.

#### 5.3.3.4 Coefficient de torrentialité

C'est un coefficient qui tient compte à la fois de la densité des thalwegs élémentaires par la densité de drainage :

$$C_t = D_d \cdot F_1 = D_d \frac{N_1}{S} \quad (5.7)$$

Tels que :

- $D_d$  : densité de drainage ;
- $F_1$  : fréquence des thalwegs élémentaires  $F_1 = N_1/S$  ;
- $N_1$  : nombre de cours d'eau d'ordre 1.

#### 5.3.3.5 Temps de concentration

C'est le temps que met une particule d'eau provenant de la partie du bassin la plus éloignée pour parvenir à l'exutoire, pour son calcul, nous faisons appel à la formule de "Gian-dotti" (Laborde 2000) :

$$T_c = 4 \frac{\sqrt{S + 1.5L}}{0.8\sqrt{H_{moy} - H_{min}}} \quad (5.8)$$

Tels que :

- $T_c$  : temps de concentration (heures) ;
- $S$  : superficie du bassin (km<sup>2</sup>) ;
- $L$  : longueur du thalweg principal (km) ;
- $H_{moy}$  : altitude moyenne (m) ;
- $H_{min}$  : altitude minimale (m).

Les paramètres physiographiques du bassin d'oued Mouilah sont résumés dans le tableau (5.5) :

TABLE 5.5 – Paramètres physiographiques du bassin d'Oued Mouilah

Sous bassin	$D_d$	$R_c$	$R_L$	Coef. de torrentialité ( $C_t$ )	Temps de concentration ( $T_c$ )
Oued Mouilah	0.16	3.88	2.34	0.0032	20 <sup>h</sup> 30 <sup>mn</sup>

#### 5.3.4 La géologie de la région

Le bassin versant de l'Oued Mouilah est dominé par les sols calcaires qui longent son thalweg principal et se prolongent au nord-est des monts des Traras et aux piémonts des monts

de Tlemcen. Il comporte aussi des formations calcaïques peu profondes (luvisols) et des terrains alluviaux (fluvisols) développés dans la partie nord de la plaine de Maghnia. La partie sud de la plaine comprend des sols rouges (arénoles) à encroûtement formés de marnes salifères du Miocène (Hudson 1990 ; Ghenim 2008).

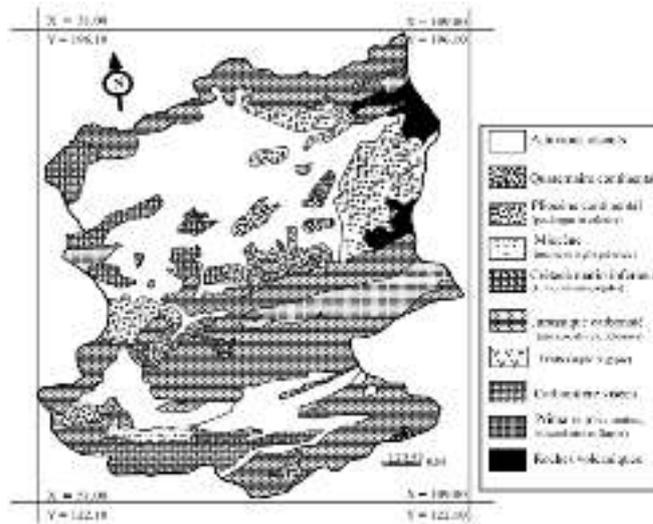


FIGURE 5.8 – La carte géologique du bassin versant de l'Oued Mouilah (Ghenim 2008)

### **5.3.5 Caractéristiques climatiques de la région**

Le climat joue un rôle primordial dans le comportement de bassin versant notamment l'écoulement au niveau des cours d'eau.

#### **5.3.5.1 La pluviométrie**

La pluviométrie représente sans doute le facteur le plus important dans le comportement hydraulique d'un bassin versant. Elle influe sur l'écoulement des cours d'eau, le remplissage des nappes, la distribution de couvert végétal et même l'érosion des sols. De plus, la pluviométrie constitue l'élément le plus accessible par les mesures dans diverses régions du monde à partir duquel plusieurs variables peuvent être déduites et étudiées notamment les débits des cours d'eau.

La figure 5.9 montre la répartition spatiale de la pluviométrie moyenne du bassin de la Tafna. A partir de cette figure, on observe clairement la diminution de la pluviométrie à mesure que l'on s'éloigne du littoral à cause de l'appauvrissement progressif de l'atmosphère en vapeur d'eau lors du passage des courants aériens qui abandonnent leur pluie en franchissant les chaînes montagneuses. En effet, la station de Maghnia située à une altitude de 395m ne

reçoit que  $357.4mm$ , alors qu'à  $30km$  au Nord, la station de Nedroma est plus arrosée et reçoit  $462.3mm$  environ. Les chaînes montagneuses représentent des obstacles de passage de nuage en provoquant la diminution des précipitations sur les versant opposés.

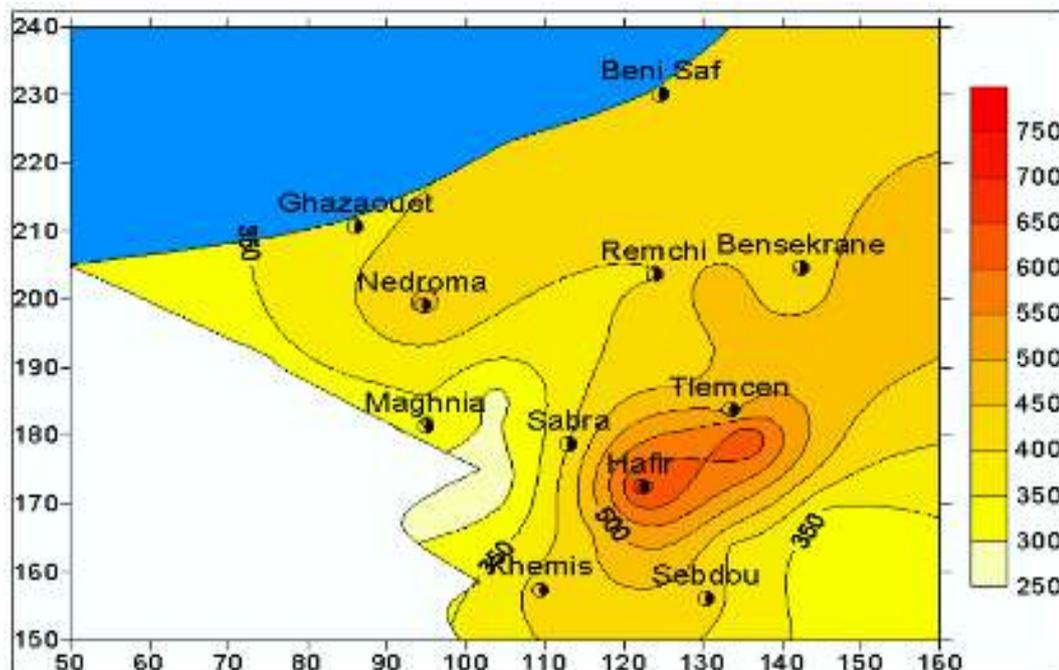


FIGURE 5.9 – Carte des isohyètes du bassin versant de Tafna (Bounoua 2014)

La figure 5.10a montre les précipitations moyennes mensuelles de la région, établies à la base de la série pluviométrique enregistrée à la station de Maghnia avant l'année 1990 et entre janvier 1997 au décembre 2016. Les valeurs de la nouvelle série montrent clairement l'existence de deux (02) saisons ; la première est humide allant du mois d'octobre à avril ; la deuxième est sèche s'étendant du mois de Mai au septembre. Le mois le plus arrosé est celui de novembre avec une moyenne de  $46.7mm$ , alors que le mois le plus sec est celui de juillet avec une moyenne de  $1.0mm$  environ. Toujours sur la figure 5.10a, les précipitations moyennes mensuelles présentent ces dernières décennies une diminution de 05% à 30% environ aux mois de décembre au juin, face à une légère augmentation aux mois d'août et septembre par rapport à celles enregistrées avant l'année 1990. Les fréquences de précipitation sont données dans la figure 5.10b.

### 5.3.5.2 La température de la région

Le climat de la région se caractérise par un été chaud et sec, et un hiver relativement doux. La température moyenne annuelle varie entre  $10.4^{\circ}C$  et  $27.8^{\circ}C$  dans cette région. Le mois le plus froid est celui de janvier (la température moyenne oscille entre  $4.1$  et  $16.6^{\circ}C$ ) ; le mois le plus chaud est celui d'Aout (température moyenne entre  $20.3$  et  $35.3^{\circ}C$ ). La figure 5.11 résume

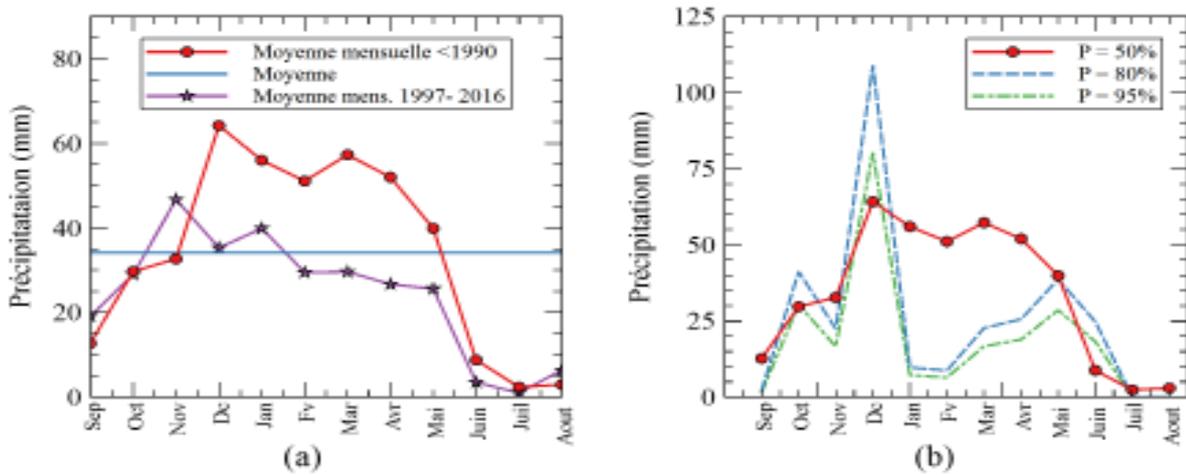


FIGURE 5.10 – (a) la répartition des précipitations moyenne mensuelles (Station de Maghnia), (b) distribution mensuelle des fréquences de précipitations

les moyennes mensuelles des températures minimales, moyennes et maximales pour la période (1997 – 2016).

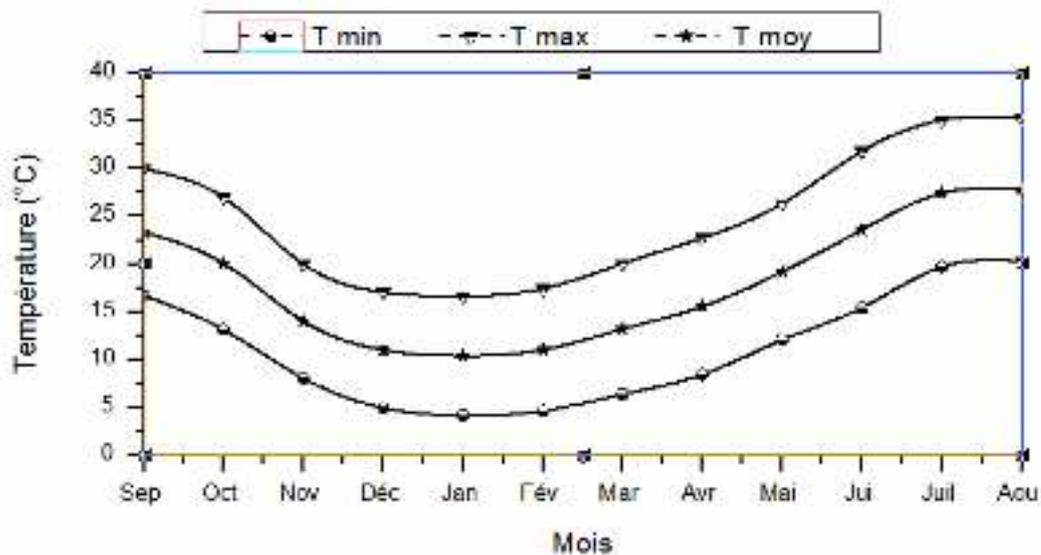


FIGURE 5.11 – Températures de la région de Maghnia (1997-2016)

## 5.4 Conclusion

Le cas d'étude de cette thèse est le barrage Hammam Boughrara situé au nord-ouest algérien (W. Tlemcen). Le choix est justifié par l'effet que cet ouvrage est construit pour un usage multi-objectifs, l'alimentation en eau potable de diverses agglomérations et l'irrigation des plaines de Maghnia (W. Tlemcen).

Le bassin versant alimentant le réservoir Hammam Bouhrara se situe dans une zone semi-aride. Le climat est très variable de l'hiver à l'été. Les précipitations, l'élément le plus important dans le processus de remplissage du réservoir, sont aussi caractérisées par une grande variabilité. Cette dernière rend la prévision des apports du réservoir une opération délicate est soumise aux fortes incertitudes.

Le fonctionnement du réservoir Hammam Bouhrara qui constitue le cas d'étude de ce travail fait l'objet du prochain chapitre.



# Chapitre 6

## Fonctionnement du barrage Hammam Boughrara

### 6.1 Introduction

Les barrages sont réalisés à travers le monde et depuis l'histoire, sans doute, à fin d'atteindre certains objectifs et accomplir certaines missions. La simulation et l'optimisation de la gestion d'un réservoir dans l'avenir passent obligatoirement par la modélisation des apports (les entrées) et les usages de l'eau (les sorties) du réservoir. Tout en gardant que cette modélisation ne peut être que conceptuelle et soumise à plusieurs incertitudes due à la méconnaissance exacte des différents processus physiques, sociales et économiques.

Ce chapitre a comme objectifs, (1) présenter la situation géographique et hydraulique (apports, pertes et qualité de l'eau) du barrage Hammama Boughrara (2) décrire les objectifs de l'usage de l'eau du réservoir, et (3) projeter les demandes en eau domestiques (AEP), industrielles et agricoles dans l'avenir (à différents horizons). L'horizon de 10 ans (du janvier 2017 au décembre 2026) est choisi dans cette étude. Il débute dès la fin de la série d'apports disponible (décembre 2016) du réservoir. Cette phase est essentielle pour construire la fonction objectif d'optimisation de fonctionnement du réservoir Hammam Boughrara.

### 6.2 Présentation du barrage Hammam Boughrara

Le barrage Hammam Boughrara est le plus important parmi ceux de la wilaya de Tlemcen. Il est situé à 50 Km environ au nord-ouest de la wilaya, 13km environ à l'Est de la Daira de Maghnia et à un kilomètre en amont de la localité Hammam Boughrara. Le barrage

est construit sur le point de confluence entre l'Oued Mouilah et l'Oued Tafna portant les coordonnées :  $X = 1^{\circ}39'61''$  Ouest,  $Y = 34^{\circ}52'23''$  Nord (UTM) et  $Z = 285m.NGA$  (mètre Nord Géodésique Algérien). La construction du barrage a été commencée en 1994, tandis que la mise en eau et l'exploitation n'ont été débutées qu'au 1998 et 2000 respectivement. La figure 6.1 démontre la situation géographique de l'ouvrage.



FIGURE 6.1 – La situation géographique du barrage Hammam Boughrara (W. Tlemcen)

### 6.2.1 La digue

Le barrage est réalisé par remblai en alluvions compactées avec noyau en argile, dont le rechargement des talus est réalisé par des graviers roulés (Figure 6.2). Les pentes de la digue en amont sont, 1/3 du pied à la risberme (cote 280.00 mNGA) et 1/2.5 de la risberme à la crête. En aval, elles sont 1/1.17 du pied à la risberme (cote 266.00 mNGA) et 1/2.25 de la risberme à la crête. Les largeurs des risbermes amont et aval sont 5m et 6m respectivement. A l'amont, le remblai s'appuie contre le batardeau intégré à la digue. La cote de retenue normale (RN) se situe à 305 mNGA. Par contre, les niveaux des crues centennales et millénales se trouvent respectivement à 308.5mNGA et 309.95mNGA. La crête est de largeur de 10m, sur laquelle une chaussée est aménagée et bordée par un parapet en béton aux côtés. Cette chaussée est réservée exclusivement au trafic lié à l'exploitation de l'ouvrage. Pour permettre le suivi de l'ouvrage, des bornes topographiques sont installées du côté aval de la crête (Bounoua 2014; Smail 2014; Adjim 2018).

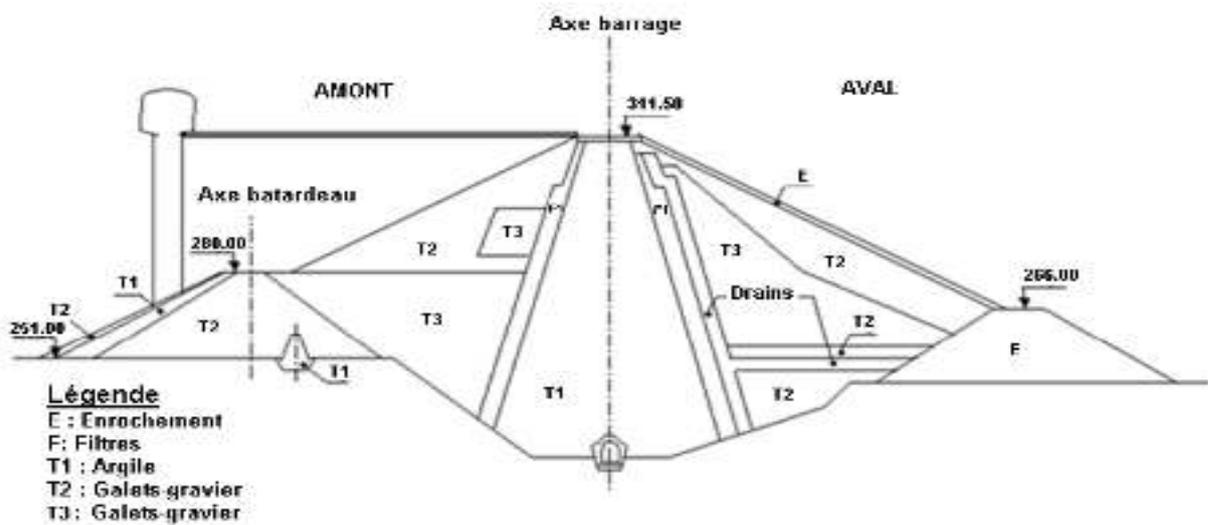


FIGURE 6.2 – Profil en travers du barrage Hammam Boughrara (Smail 2014)

### 6.2.2 Évacuateurs de crue

De plus la vidange de fond, le barrage Hammam Boughrara dispose deux évacuateurs de crues, le premier est principal, le deuxième est fusible.

#### 6.2.2.1 Évacuateur de crue principal

Un évacuateur de crue à seuil libre est prévu pour assurer la sécurité de l'ouvrage. C'est un déversoir à seuil libre latéral avec un coursier en pente (de longueur de 550m environ) qui se termine par un saut de ski situé en rive droite. Cet évacuateur est dimensionné pour évacuer la crue centennale estimée d'avoir un débit de  $3000 \text{ m}^3/\text{s}$ . De plus, le débit de pointe à évacuer via laminage de la retenue supporté par la vidange de fond est de  $2120 \text{ m}^3/\text{s}$ . En cas de crue milléniale (estimée à  $5880 \text{ m}^3/\text{s}$ ), l'évacuateur de crues principal est renforcé par un évacuateur de crue catastrophique complémentaire (TRACTEBEL 2003 ; Bounoua 2014 ; Smail 2014).

#### 6.2.2.2 Évacuateur de crue fusible

C'est l'évacuateur de crue catastrophique. Il prend la forme d'un déversoir à seuil large de longueur de 250m divisé en travées de 20m de longueur (comptées entre culées). L'évacuateur est constitué d'un radier général en béton armé avec des dalles de 1.5m d'épaisseur bétonnées sur place, sur lesquelles la partie fusible de l'évacuateur est construite. L'ouvrage est implanté en rive droite de la retenue, à environ 600 m de l'épaulement rive droite du barrage. La

crue millénaire supposée d'atteindre le débit de  $5880\text{m}^3/\text{s}$  est prévue d'être évacuée par l'évacuateur principal, la vidange de fond et l'évacuateur fusible. Ce dernier n'est pas confectionné pour fonctionner lorsque la crue centennale arrive (Bounoua 2014).

### **6.2.3 La tour de prise**

Une tour de prise classique de forme circulaire est prévue pour l'ouvrage. Elle est de 65m de hauteur (environ) et dotée de quatre niveaux de prise d'eau implantés aux cotes : 272.2 mNGA, 281.2 mNGA, 290.2 mNGA, 299.2 mNGA. Ces prises font 1.4m de diamètre et raccordées 2 par 2 à deux conduites qui descendent verticalement dans la tour. Ces conduites rejoignent l'aval dans la partie supérieure de la dérivation transformée. Pour garantir sa stabilité, elle a été intégrée au batardeau (Figure 6.2) (Bounoua 2014; Smail 2014).

De plus, l'ouvrage est équipé par une vidange de fond de capacité d'évacuation de  $300\text{m}^3/\text{s}$  et une galerie de dérivation servant à la vidange de fond dans sa partie inférieure et aux conduites de prise d'eau dans sa partie supérieure (TRACTEBEL 2003; Smail 2014). Le tableau 6.1 résume les caractéristiques principales du barrage Hammam Boughrara.

## **6.3 Pollution du réservoir Hammam Boughrara**

Le bassin versant du barrage Hammam Boughrara s'étale sur des vastes terres sur lesquelles multiples activités économiques (agriculture, industrie, commerce, ...) sont développées. Ces activités génèrent au fil du temps des volumes importants de pollutions hydriques et terrestres qui se déplacent vers la cuvette du réservoir. Le réseau hydrographique du bassin, constitué essentiellement par l'oued Mouilah et ses affluents (oued Isly, oued Bounaïm, oued Ouerdeffou et oued Abbès), est le principal transporteur de la pollution (Zenasni 2013; Hamlat et al. 2014). Depuis la mise en eau du barrage, le réservoir accumule les eaux polluées dérivées de l'agriculture, les eaux usées domestiques et industrielles issues de la ville de Maghnia en Algérie et la ville d'Oujda au Maroc. Malheureusement, pendant une longue durée, aucun traitement n'a été prévu en amont du barrage pour atténuer cette pollution (Bounoua 2014).

Plusieurs études ont été établies sur la qualité des eaux du barrage Hammam Boughrara afin de déterminer la nature et le degré de la pollution. Selon (Bouzaïd-Lagha and Djelita 2012), les eaux du barrage dévoilent des concentrations élevées en :  $Cl$ ,  $SO_4$ ,  $NH_4$ ,  $NO_2$  et  $DCO$ . Les auteurs ont conclu que les eaux sont polluées, séléniteuses et eutrophiées avec la possibilité de traitement biologique. (Hamlat et al. 2014) ont indiqué que les concentrations en :  $NO_2$ ,  $NH_4$ ,  $DBO_5$  et  $DCO$  ont dépassé les normes algériennes sur la période d'étude

TABLE 6.1 – Caractéristique du barrage Hammam Boughrara (TRACTEBEL 2003)

Caractéristique	Valeur
Type du barrage	Digue en terre hétérogène
Volume de remlais du barrage ( $m^3$ )	$3.491 \times 10^6$
Capacité totale ( $Mm^3$ )	$\cong 177$
Volume régularisé ( $Mm^3$ )	59
Volume mort ( $Mm^3$ )	$\cong 23.0$
Longueur en crête ( $m$ )	1100
Largeur en crête ( $m$ )	10
Hauteur de la digue ( $m$ )	61
Cote de retenue normale (R.N) ( $m$ NGA)	305
Cote plus hautes eaux (P.H.E) ( $m$ NGA)	309.95
Cote vidange de fond ( $m$ )	257
Cote de prise inférieure ( $m$ )	272
Apport moyen annuel ( $Mm^3/an$ )	54.15
Débit de la vidange de fond ( $m^3/s$ )	310
Apport solide moyen ( $T/an$ )	$5.011 \times 10^5$
Débit moyen d'infiltration $m^3/j$	100
Débit de crue millénale (1000 ans)( $m^3/s$ )	5880
Superficie de lac ( $Ha$ )	894
Superficie de bassin versant ( $Km^2$ )	4000
Années de construction	1994-1998
Année de mise en eau	1999

(2001-2010), et que la qualité de l'eau du réservoir tombe dans la catégorie pauvre, ainsi qu'elle a été menacée chaque année et atteint rarement les normes désirées.

TABLE 6.2 – Qualité de l'eau du réservoir Hammam Boughrara (2001-2010) (valeurs moyennes) (Hamlat et al. 2014)

Ca	Mg	Dureté	MES	$NO_3$	$NO_2$	$NH_4$	$PO_4$	Temp	pH	$BOD_5$	$DCO$
			(mg/L)					(°C)	-	(mg $O_2/L$ )	
71.04	54.89	406.31	1124.39	6.05	0.84	2.95	2.08	19.47	8.29	12.01	57.68

### **6.3.1 Pollution d'origine domestique**

La ville d'Oujda (le territoire marocain) déverse les eaux usées dans l'oued Bounaïm l'amont de l'oued Mouilah sans aucun traitement préalable. Longtemps, ces eaux trouvent la fin de chemin dans la cuvette du réservoir Hammam Boughrara.

Des campagnes d'analyse de la  $DBO_5$  et la  $DCO$  ont été faites entre 1996 et 2008 sur l'Oued Mouilah par le laboratoire de l'agence nationale des ressources hydriques (ANRH Oran). Les résultats d'analyse montrent que les valeurs de la  $DBO_5$  et la  $DCO$  sont augmentés à partir l'année 1999, en surpassant les normes fixées par l'Organisation Mondiale de la Santé (OMS) (Zenasni 2013).

Sur le coté algérien, les rejets de la région de Maghnia et Hammam Boughrara (plus de 126 000 habitants RGP-2008), ainsi toutes les agglomérations situées dans le bassin versant, se transportent par le réseau hydrographique et finissent dans le réservoir. Dans le but de réduire la charge polluante, une station d'épuration a été réalisée pour traiter les eaux usées de la ville de Maghnia (Bounoua 2014). De plus, une décharge publique ne répondant pas aux normes se situe à 2.5 Km environ de la ville de Maghnia, dans laquelle différents déchets ménagers, hospitaliers, industriels et agricoles sont jetés (Zenasni 2013). Sous l'effet de ruissellement, ces déchets trouvent la fin dans la cuvette du barrage Hammam Boughrara.

### **6.3.2 Pollution d'origine agricole et industrielle**

Les activités agricoles, attirées par la qualité du sol, s'étalent sur la quasi-totalité du bassin versant alimentant le réservoir Hammam Boughrara. Les engrais constituent le premier facteur de pollution. Les plus utilisés sont le sulfate de potassium (15-15-15) (6364 tonnes/an environ), phosphate super 46% (1144 tonnes/an environ), Sulfate de Potassium (511.5 tonnes/an environ) et Sulfate d'Ammonium (299 tonnes/an environ). Les pesticides contribuent fortement à la pollution des eaux souterraines et superficielles. Les plus utilisés sont le méthyle paratox "pesticide" (6850 kg/an environ) et le soufre sublimé "fongicides" (4801 kg/an environ). L'élevage avicole en particulier (environ  $20 \times 10^3$  tonnes/an), constitue une source majeure de pollution via la fientes de volaille qui touche les eaux superficielles et souterraines (Zenasni 2013).

Sur le coté algérien du bassin versant, quatre (04) unités industrielles sont actives à Maghnia : l'entreprise de bentonite et de la terre décolorante "ENOF", l'entreprise des corps gras "ENCG", l'entreprise des aliments et dérivés "Maïserie ERIAD" et le complexe de la vaiselle et céramique "CERTAF", lesquelles évacuent leurs rejets vers le barrage Hammam Boughrara. Cependant plus de 122 entreprises industrielles sont actives dans le coté marocain à Oujda

(Zenasni 2013) et participent à l’amplification de la pollution des eaux du barrage.

### 6.4 Présentation hydraulique du réservoir H. Boughrara

Dans cette section on s’intéresse à la description hydraulique du réservoir, c-à-d., en termes de bilan hydrique gouvernant le fonctionnement de ce-ci.

#### 6.4.1 Relation hauteur-surface-capacité du réservoir H.Boughrara

Le levé bathymétrique effectué au niveau de la cuvette du barrage a permis de construire les courbes hauteurs-surfaces-volumes (figure 6.3).

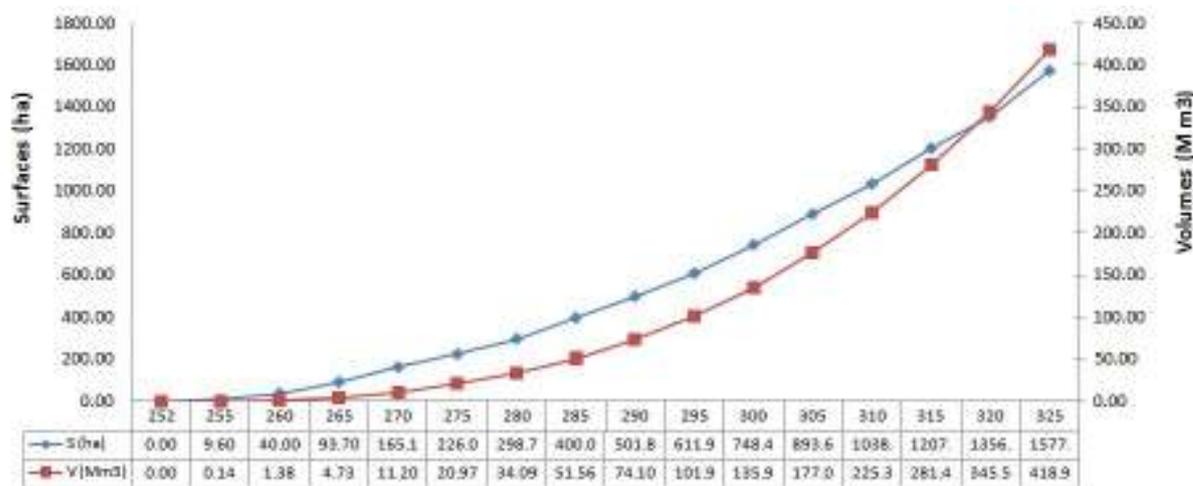


FIGURE 6.3 – Courbes hauteurs-surfaces-volumes du réservoir Hammam Boughrara (DRE (2012); ANBT (2012))

#### 6.4.2 Les apports du réservoir

Les apports "ou affluents" constituent le facteur le plus important dans le fonctionnement d’un barrage. Le réservoir Hammam Boughrara reçoit l’eau de deux oueds, Tafna et Mouilah. Ce dernier le principal adducteur du réservoir avec un taux de 98% d’environ. Les apports d’oued Tafna sont minimes, car ils sont retenus par le barrage Béni-Bahdel situé à 30Km environ en amont du barrage Hammam Boughrara (figure 6.4).

L’oued Mouilah prend naissance aux montées de la région d’Ebed (altitude 1250m environ) sur le territoire algérien. Il pénètre, ensuite, le territoire marocain en suivant un cours d’eau intermittent "Oued Isly". À l’aval de la ville d’Oujda (au Maroc), le cours d’eau devient

permanent en prenant le nom de "Oued Bounaîm". Ce dernier repénètre le territoire algérien dans la région de Maghnia en portant le nom d'oued Mouillah. Sur la rive gauche, il se renforce par l'oued Bouslita. Sur sa rive droite, il rattrape l'oued Ouerdeffou généré aux exutoires des oueds Lemhaguene et Aouina. Ensuite, l'oued Ouerdeffou rencontre l'oued Abbas, et continue jusqu'à déverser dans l'oued Mouillah (figure 6.4).

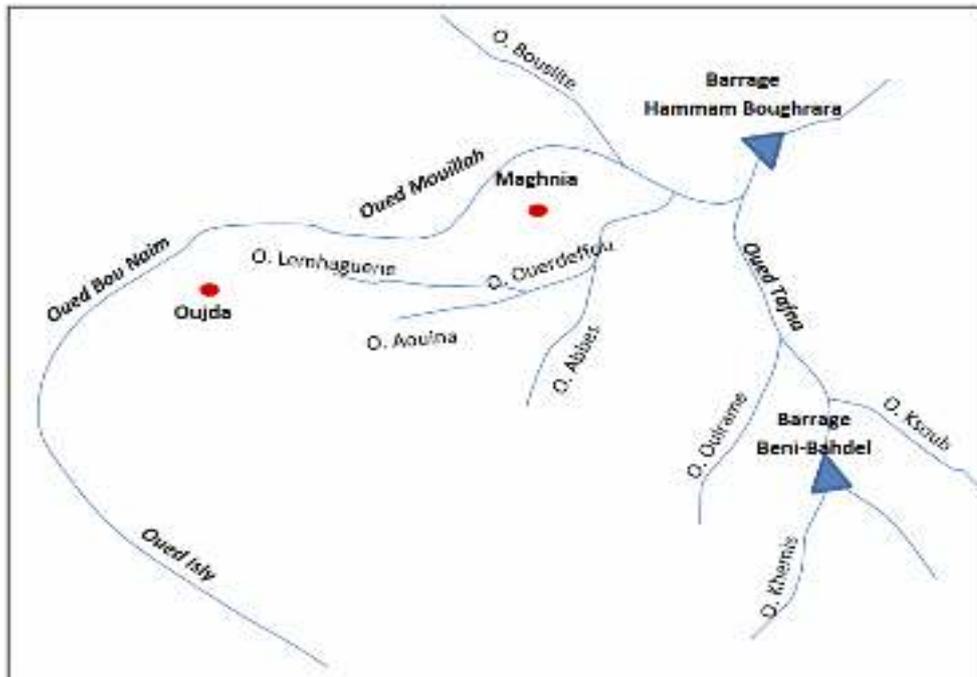


FIGURE 6.4 – Le réseau hydrographique principal alimentant le réservoir Hammam Boughrara (Bounoua 2014)

Les apports sont irréguliers et présentent une variation mensuelle et annuelle significatives (figures 6.6 et 6.5). L'apport annuel maximal a atteint  $174.68 Mm^3/an$ , apparu durant l'année hydrologique 1994/1995 (début des travaux de réalisation du barrage). Cependant, l'apport annuel minimal a figuré après 10 ans (année hydrologique 2004/2005) avec une valeur de  $15.39 Mm^3/an$ . Depuis la mise en service de l'ouvrage, les apports présentent une tendance de diminution sauf pour l'année hydrologique 2008/2009 où ils atteignent  $144.95 Mm^3/an$ . Le nombre d'années déficitaires "au-dessous de la moyenne" vaut 29/44, soit d'environ 66%. Elles sont fréquentes notamment après la mise en service du réservoir. L'écart-type et le coefficient de variation des apports annuels valent  $37.78 Mm^3/an$  et 0.71 respectivement (tableau 6.3). Ces derniers confirment les constatations tirées par les graphiques.

À l'échelle mensuelle, des fluctuations plus avérées apparaissent. Elles sont validées par les valeurs importantes de l'écart-type ( $8.08 Mm^3/mois$ ) et de coefficient de variation ( $CV \cong 1.82$ ) (tableau 6.3). Sur la série chronologique disponible, des pics résultant de fortes crues sont apparus en : Déc-1972 ( $45.34 Mm^3/mois$ ), Avril-1974 ( $54.88 Mm^3/mois$ ), Oct-1978 ( $51.67 Mm^3/mois$ ), Nov/1986 ( $81.08 Mm^3/mois$ ) et Mars-1995 ( $105.18 Mm^3/mois$ ), ce

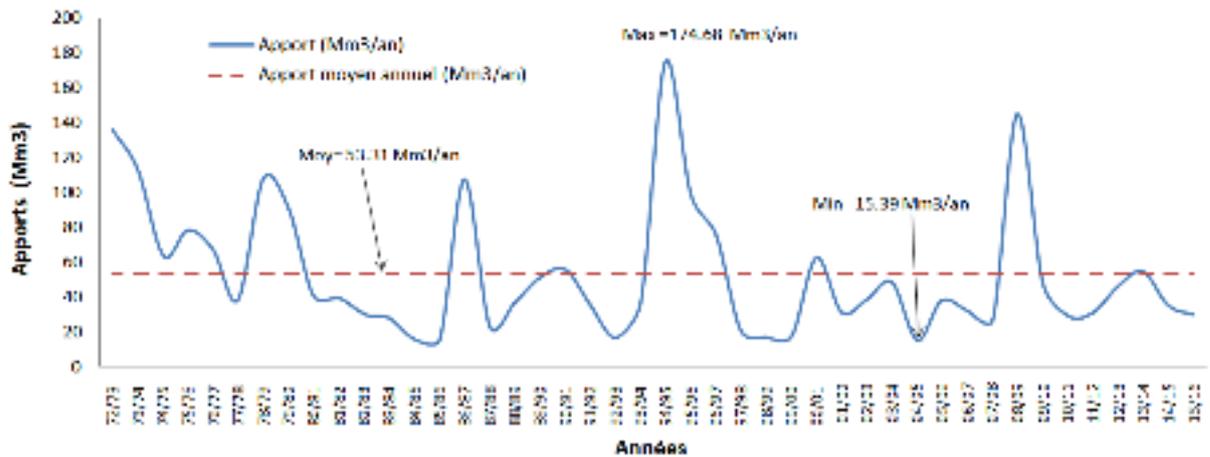


FIGURE 6.5 – Les apports annuels du réservoir Hammam Boughrara (1972- 2016) (DRE 2012; ANBT 2012)

dernier représente l’apport mensuel maximal. Néanmoins, depuis la mise en service du réservoir (en Sep-2000), quelques pics sont apparus, mais moins notoires. Le plus important est du mois d’Octobre 2008 ( $36.35Mm^3/mois$ ).

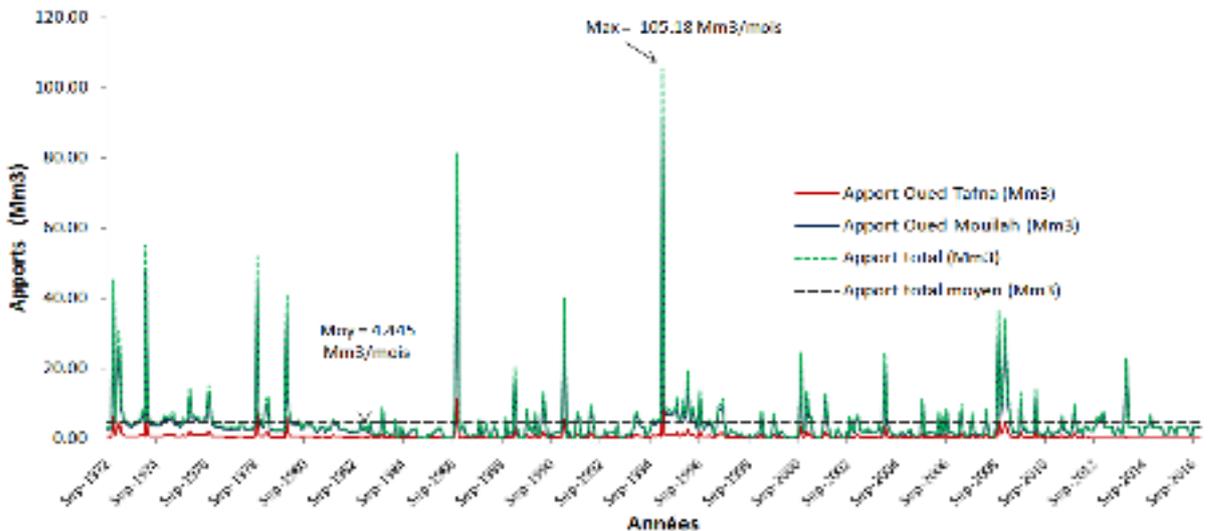


FIGURE 6.6 – Les apports mensuels du réservoir Hammam Boughrara (Septembre 1972 - Décembre 2016) (DRE 2012; ANBT 2012)

TABLE 6.3 – Caractéristiques de la série chronologique des apports du réservoir Hammam Boughrara à l'échelle annuelle et mensuelle

Apport total (Sep/1972-Déc/2016)	Min	Max	Moyenne	Médiane	Écart-type	CV <sup>a</sup>
	<i>(Mm<sup>3</sup>)</i>					-
Apports annuels (44 ans)	15.39	174.68	53.31	38.25	37.78	0.71
Apports mensuels (532 mois)	0	105.18	4.445	2.795	8.082	1.818

<sup>a</sup> Coefficient de variation (CV).

L'analyse des apports moyens mensuels interannuels permet de mettre en évidence le régime des cours d'eau et leur variation interannuelle et inter-saisonnière. L'apport moyen, maximal, minimal, écart-type et coefficient de variation de chaque mois sont donnés dans le tableau 6.4. Encore, la variation pertinente des apports mensuels se réaffirme, notamment aux mois de novembre, décembre, mars et avril. L'écart-type le plus important se localise au mois de mars avec une valeur de  $SD = 17.06Mm^3/mois$ , fait environ deux fois la moyenne ( $8.50Mm^3/mois$ ). En revanche, l'écart-type le plus faible figure au mois d'août ( $SD=2.23Mm^3/mois$ ). Les apports du mois de novembre se caractérisent par le coefficient de variation le plus important qui atteint  $CV = 2.22$ . De l'autre côté, les apports mensuels de faible variation interannuelle sont du mois de mai ( $CV = 0.74$ ).

La figure 6.7 montre la représentation graphique de la variation mensuelle interannuelle des apports du réservoir Hammam Boughrara. À partir de cette figure, on constate clairement que la saison des plus hautes eaux est l'automne (octobre, novembre et décembre). Également, l'arrière saison hivernale (mois de mars) peut présenter un apport mensuel considérable, néanmoins, l'écart-type et le coefficient de variation restent élevés dans ce cas. En revanche, la période des basses eaux est l'estivale (juin, juillet et août), dans laquelle l'apport mensuel atteint au moyen  $2.08Mm^3/mois$ .

TABLE 6.4 – Caractéristiques des apports moyens mensuels du barrage Hammam Boughrara (1972 – 2016)

Mois	Sep	Oct	Nov	Déc	Jan	Fev	Mars	Avr	Mai	Juin	Juil	Aout
Min <sup>a</sup>	0.24	0.17	0.29	0.61	0.23	0.29	0.49	0.13	0.14	0.00	0.00	0.12
Max <sup>a</sup>	13.69	51.67	81.08	79.00	33.86	24.43	105.18	54.88	13.11	9.16	9.97	11.02
Moy <sup>a</sup>	3.95	6.14	5.39	7.20	4.89	4.18	<b>8.50</b>	4.83	3.69	2.50	2.10	<b>2.08</b>
SD <sup>b</sup>	3.36	9.67	11.96	14.40	5.62	4.44	<b>17.06</b>	8.31	2.74	2.50	2.29	<b>2.23</b>
CV	0.85	1.58	<b>2.22</b>	2.00	1.15	1.06	2.01	1.72	<b>0.74</b>	1.00	1.09	1.07

<sup>a</sup> en  $Mm^3/mois$ .

<sup>b</sup> Ecart-type "Standard deviation" en  $Mm^3/mois$ .

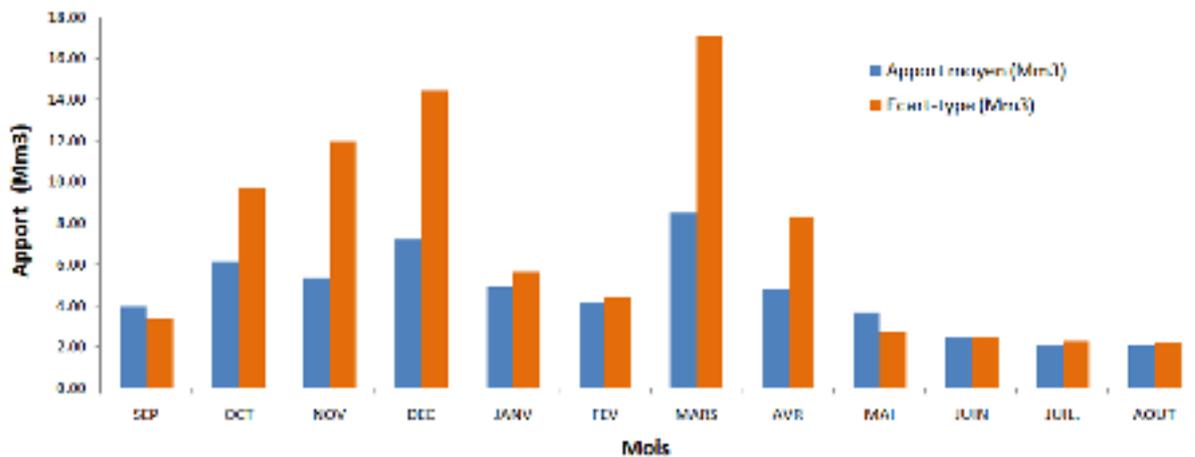


FIGURE 6.7 – Variation mensuelle interannuelle des apports du barrage Hammam Boughrara (Septembre 1972 - Décembre 2016)

### 6.4.3 Les pertes d'eau par évaporation

Dans les pays arides ou semi-arides, l'évaporation devient plus en plus conséquente. A cet effet, elle ne doit pas être écartée lorsque une planification de la gestion d'un réservoir est envisagée. La figure 9.2 montre la variation mensuelle des hauteurs d'eau évaporées à partir de lac du barrage Hammam Boughrara durant la période (septembre 2000 - août 2011). D'un mois donné, le volume de perte par évaporation peut être déterminé moyennant la multiplication de la hauteur évaporée par la surface de plan d'eau correspondante (équation 1.9). L'évaporation minimale a atteint  $12mm/mois$  observée en avril-2001, toutefois la maximale a atteint  $521mm/mois$  en mois d'août-2006. L'évaporation moyenne mensuelle de la période d'observation est d'environ  $119.42mm/mois$ .



Temouchent);

3. Contribution à l'irrigation du périmètre agricole de Maghnia (d'environ 7000ha) par un volume de  $9Mm^3/an$ . Notons que le périmètre est irrigué aussi par le barrage Béni-Bahdel.

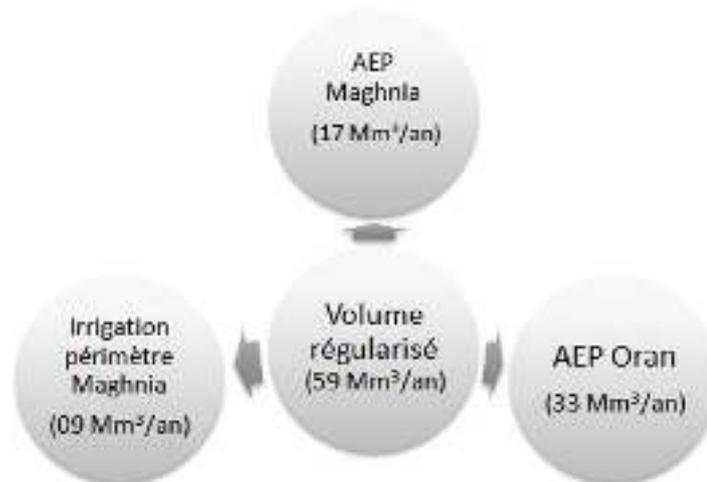


FIGURE 6.9 – Les objectifs initiaux d'allocation de l'eau du réservoir Hammam Boughrara

Avec la mise en service de la station de dessalement d'El-Mactaâ (W. Oran) en 2015, dont sa capacité de production atteint  $500'000 m^3/jour$ , destinée au renforcement de l'approvisionnement en eau potable de la région d'Oran, ainsi la réalisation de deux stations de dessalement de Honaine et de Souk Tleta (W. Tlemcen) (capacité de  $200'000 m^3/jour$  chacune), la direction des ressources en eau de la wilaya de Tlemcen a entamé une nouvelle allocation de l'eau du réservoir favorisant l'irrigation et la valorisation des terres agricoles. Ce changement d'allocation peut être consolidé par plusieurs raisons, parmi lesquelles :

- Le potentiel agricole de la région notamment le périmètre de Maghnia étendu sur grande superficie;
- La qualité eutrophiée de l'eau du barrage, qui est plus avantageuse pour l'irrigation;
- L'incapacité du réservoir à régulariser le volume de  $59 Mm^3$  dépassant le moyen annuel des apports (aux alentours de  $44.43 Mm^3$  "période 2001-2012").

Dans ce dernier contexte, (Adjim and Djedid 2018) ont étudié les séries chronologiques de précipitations de quatre stations météorologiques (Maghnia, Hammam Boughrara, Beni Bahdel et Sabra) situées à proximité du barrage. Les observations couvrent en totalité la période (1914-2011). Les auteurs ont constaté une diminution significative des précipitations à partir de l'année hydrologique 1975/1976. En effet, ils ont évoqué l'incapacité du réservoir à régulariser le volume de  $59Mm^3/an$  et ont proposé le volume de  $37Mm^3/an$  seulement, donc une diminution d'environ 37.3%.

D'après les services de l'office de périmètre irrigué de Maghnia (OPI), le périmètre été alimenté exclusivement à partir du barrage de Beni Bahdel. Avec la nouvelle affectation de la ressource en eau, un volume de  $38 \text{ Mm}^3/\text{an}$  du barrage de Beni Bahdel et  $20 \text{ Mm}^3/\text{an}$  du barrage de Hammam Boughrara (soit un volume total de  $58 \text{ Mm}^3/\text{an}$ ) sont dédiés au périmètre (Bensalem and Bouhenna 2017). Cette nouvelle allocation permet d'étendre le réseau d'irrigation pour couvrir la quasi-totalité du périmètre (d'environ  $7000 \text{ ha}$ ).

En guise de conclusion, la nouvelle affectation de l'eau envisagée prévoit  $17 \text{ Mm}^3/\text{an}$  au profit de l'approvisionnement en eau potable de la région de Maghnia et  $20 \text{ Mm}^3/\text{an}$  au profit de l'irrigation du périmètre agricole. La totalité du volume alloué s'égalise approximativement à la valeur du volume régularisé proposé par (Adjim and Djedid 2018).

### 6.5.2 Approvisionnement en eau potable

La région de Maghnia, à laquelle le réservoir Hammam Boughrara appartient, est alimentée par ce dernier depuis sa mise en service en septembre 2000. En revanche, la wilaya d'Oran n'a pas été allouée en eau car le projet du transfert n'est pas réalisé jusqu'à maintenant.

La figure 6.10 présente les statistiques des volumes annuels déjà desservis pendant la période 2000-2012.

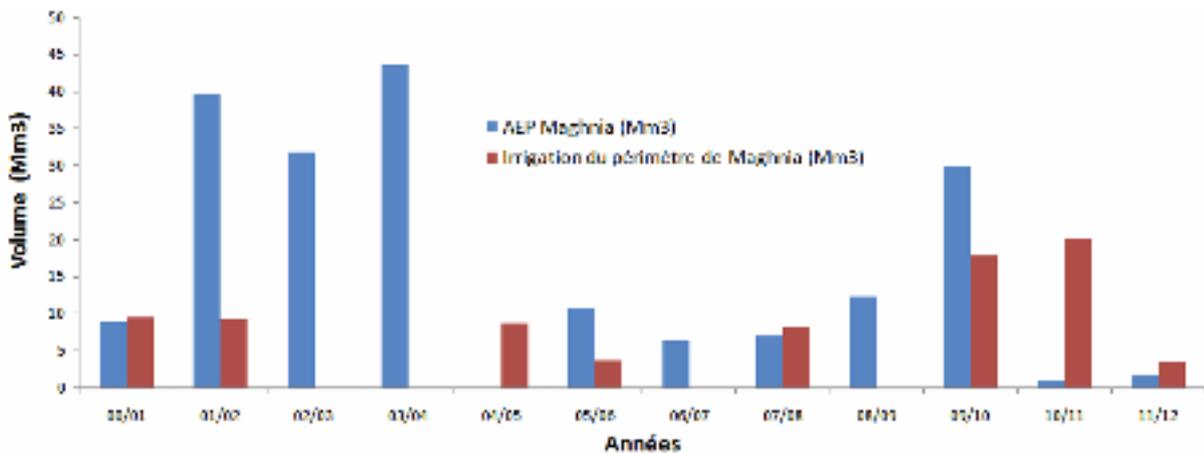


FIGURE 6.10 – Les volumes annuels desservis à la région de Maghnia et son périmètre d'irrigation à partir du réservoir Hammam Boughrara (2001-2012) (ANBT 2012 ; DRE 2012)

### 6.5.3 Approvisionnement en eau d'irrigation

Le périmètre de Maghnia (figure 6.11) représente un potentiel agricole dans la région. Le périmètre, créé en 1974, s'étend sur une superficie de  $7000 \text{ ha}$  environ. En 2015, l'État a

consacré une enveloppe de 04 Milliards de Dinars (DZD) pour la réhabilitation et l'extension de la superficie du périmètre et de son réseau d'irrigation. Dans ce cadre, la superficie a été prévue d'augmentée de 4500ha à 7000ha. Par ailleurs, plus de 172 Km de conduites ont été prévues d'être posées à la fin de l'opération. Le périmètre a été irrigué durant plusieurs années à partir du barrage Beni-Bahdel et quelques forages situés à ses alentours. Le réservoir Hammam Boughrara a été prévu de renforcer l'irrigation du périmètre par un volume moyen annuel de  $9Mm^3$  au départ, puis d'environ  $20Mm^3$  après l'extension. La figure 6.10 annonce les volumes déjà desservis au périmètre entre les années 2000 et 2012.

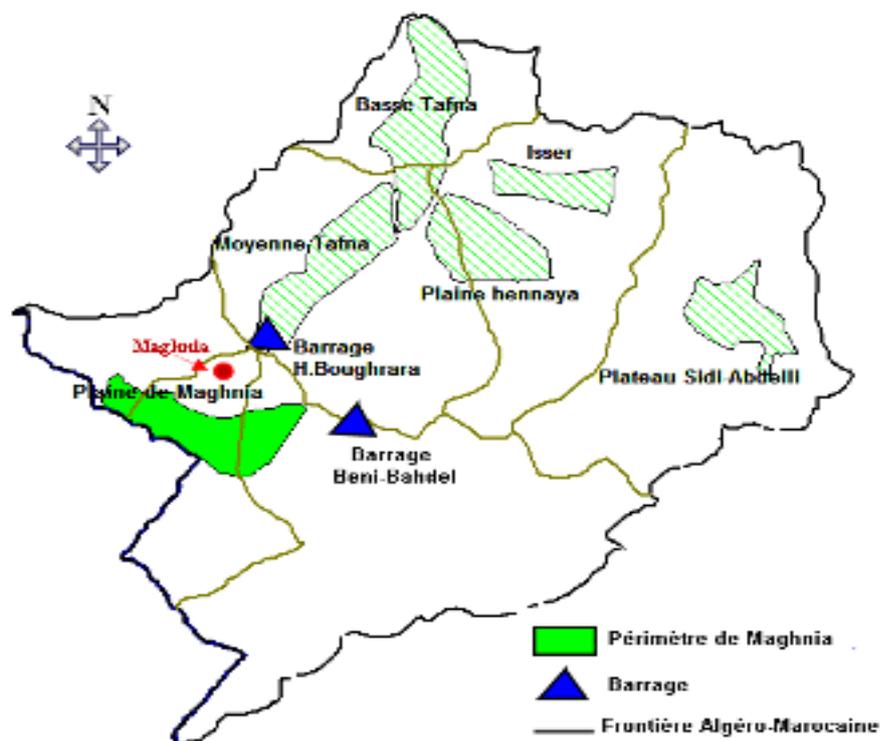


FIGURE 6.11 – Situation géographique du périmètre d'irrigation de Maghnia

## 6.6 Modélisation de l'usage de l'eau du réservoir

Le pas mensuel est choisi pour générer la consommation future. Il permet une généralisation grossière plus juste qu'une échelle plus fine (journalière ou hebdomadaire) qui nécessite une connaissance plus rigoureuse des facteurs intervenants dans l'opération de gestion d'un réservoir.

La projection des besoins de consommation en eau des usagers a comme finalité la construction de la fonction objectif de l'optimisation du réservoir.

### 6.6.1 Projection des besoins en eau potable

#### 6.6.1.1 Projection des besoins en eau potable domestique

Au début des années 2000, l'Algérie a vécu une période de sécheresse qui a touché le pays tout-entier. Elle a été l'étincelle de relancement des projets de mobilisation et transfert des eaux de surface. Dans ce cadre, pas mal de projets ont été lancés notamment pour répondre à la demande en eau potable des centres urbains. La région de Maghnia (Maghnia et Hammam Boughrara) n'échappe pas à la règle, elle est alimentée par les eaux de surfaces accumulées dans le barrage Hammam Boughrara au lieu les eaux souterraines. Dans cette section, les besoins en eau potable domestique de Maghnia sont projetés dans l'avenir.

Les besoins en eau potable d'une région dépendent de deux (02) principaux facteurs, son évolution démographique et son développement socio-économique. L'approximation de l'évolution démographique d'une agglomération peut être déterminée au moyen de la formule 6.1 ci-après :

$$P_N = P_0(1 + \tau)^N \quad (6.1)$$

Tels que,  $P_N$   $P_0$  représentent la population à l'horizon  $N$  et celle de l'année référence respectivement (en habitants).  $\tau$  est le taux d'accroissement de population de la région considérée.

D'après l'office national des statistiques (ONS), le taux d'accroissement de Maghnia est d'environ 3.0%. Les besoins domestiques en eau potable  $B_{Domestique}$  peuvent être estimés simplement par l'équation 6.2. La dotation journalière en eau potable est supposée d'environ 150 l/jour/habitant. La table 6.5 annonce l'évolution de la population de la région de Maghnia estimée par la formule 6.1 ainsi que l'estimation de ses besoins en eau potable à différents horizons.

$$B_{Domestique}(m^3/an) = Population(hab) \times Dotation(l/jour/hab) \times \frac{365}{1000} \quad (6.2)$$

TABLE 6.5 – Estimation de l'évolution de la population et ses besoins annuels en eau potable de la région de Maghnia

Années	2005	2010	2015	2017	2020	2026	2030
Population (habitant)	124 061	144 520	167 864	178 087	194 600	232 363	261 527
$B_{Domestique}$ ( $Mm^3/an$ )	6.792	7.912	9.191	9.750	10.654	12.722	14.319

La consommation mensuelle eau potable n'est pas constante d'un mois à l'autre. Généralement pour la région de Maghnia, elle suit la répartition présentée dans la figure 6.12.

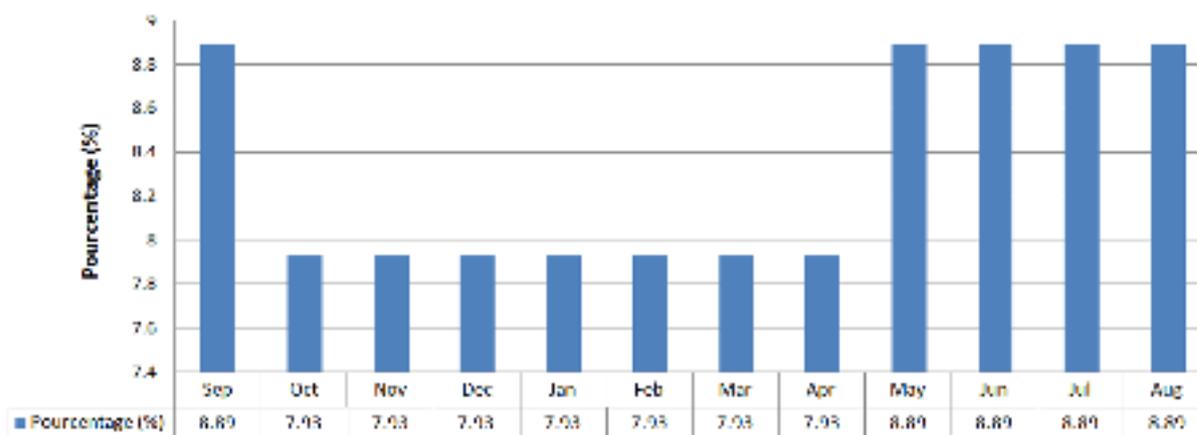


FIGURE 6.12 – Répartition mensuelle de la consommation de l'eau potable de Maghnia (année 2012) (DRE 2012)

### 6.6.1.2 Projection des besoins en eau industriels

Pour la plupart des industries, l'eau est un facteur de production. Elle peut être utilisée comme matière première, et comme incorporée aux produits finis, où même intervenir comme auxiliaire au cours des processus de fabrication. La consommation en eau des industries dépend de plusieurs paramètres parmi lesquels, le type d'industrie et processus de production, le développement technologique de l'industrie, la réduction des pertes dans le système de distribution et les procédures pour l'épargne de la consommation (Bounoua 2014).

Dans la ville de Maghnia, quelques industries agroalimentaires et de transformation sont actives. Le tableau 6.6 désigne ces dernières et leurs consommations annuelles en eau.

La projection de la consommation industrielle en eau est faite dans l'incertain. Dans cette étude, on opte pour un taux de développement annuel d'environ (2.4%) permettant de couvrir les besoins en eau des éventuels industries susceptibles d'apparaître notamment dans la filière de transformation agroalimentaire. Cela, en raison de la production agricole issue du périmètre irrigué de Maghnia. Le tableau (6.7) et la figure 6.13 résument l'estimation des besoins en eau domestiques et industriels de la région de Maghnia à différents horizons.

### 6.6.2 Projection des besoins en eau agricoles

L'approvisionnement en eau d'irrigation est un principal objectif du réservoir Hammam Boughrara. À cause de l'absence des données exhaustives sur la production agricole, le rendement du périmètre irrigué, le prix de vente des produits, la variation et l'évolution de la production dans le temps, ... etc., l'optimisation pour la maximisation des revenus économiques "monétaires" ne peut pas être établie dans ce cas. D'ailleurs, selon (Lebdi et al. 1997),

TABLE 6.6 – Consommation industrielle en eau potable de la région de Maghnia (année 2006) (Bounoua 2014)

Unité industrielle	Branche industrielle	Besoins en eau $Mm^3/an$
ERIAD MAIS	Farine maïzena	0.5475
ENCG	Corps gras	0.365
ECVO (CERTAF)	Céramique	0.292
ENOF	Bentonite	0.1825
Total		1.387

TABLE 6.7 – Projection des besoins en eau domestiques et industriels de la région de Maghnia à différents horizons

Industrie	Besoins industriels en eau ( $Mm^3/an$ )						
	2005	2010	2015	2017	2020	2026	2030
ERIAD MAIS	0.54	0.60	0.68	0.71	0.76	0.88	0.97
ENCG	0.36	0.40	0.45	0.47	0.51	0.59	0.64
ECVO (CERTAF)	0.29	0.32	0.36	0.38	0.41	0.47	0.52
ENOF	0.18	0.20	0.23	0.24	0.25	0.29	0.32
Total ( $Mm^3/an$ )	1.37	1.53	1.72	1.80	1.93	2.23	2.45
Besoins domestiques en eau ( $Mm^3/an$ )							
AEP -Maghnia- ( $Mm^3/an$ )	6.79	7.91	9.19	9.75	10.65	12.72	14.32
Total ( $Mm^3/an$ )	8.16	9.44	10.91	11.55	12.59	14.95	16.77

"sauf pour la production hydroélectrique, il est bien rare que les différents usages d'un barrage puissent être agrégés et évalués en termes d'unités monétaires".

D'après les services de la DSA (Direction des Services Agricoles) de la wilaya, quatre (04) types de cultures sont cultivées dans le périmètre, à savoir : céréales, maraichères, olivier et les arbres fruitiers "arboriculture". Les pourcentages (%) de l'assolement actuel, donnés au tableau 6.8, sont calculés à la base d'une surface de 4705 ha (Bensalem and Bouhenna 2017).

#### 6.6.2.1 Estimation des besoins en eau des cultures

Les besoins en eau des cultures dépendent de plusieurs facteurs, les types "espèces" des plantes d'elles-mêmes, le stade végétatif, la réserve de l'eau dans le sol, et notamment les conditions climatiques de la région dans laquelle les cultures sont implantées. Le climat condi-

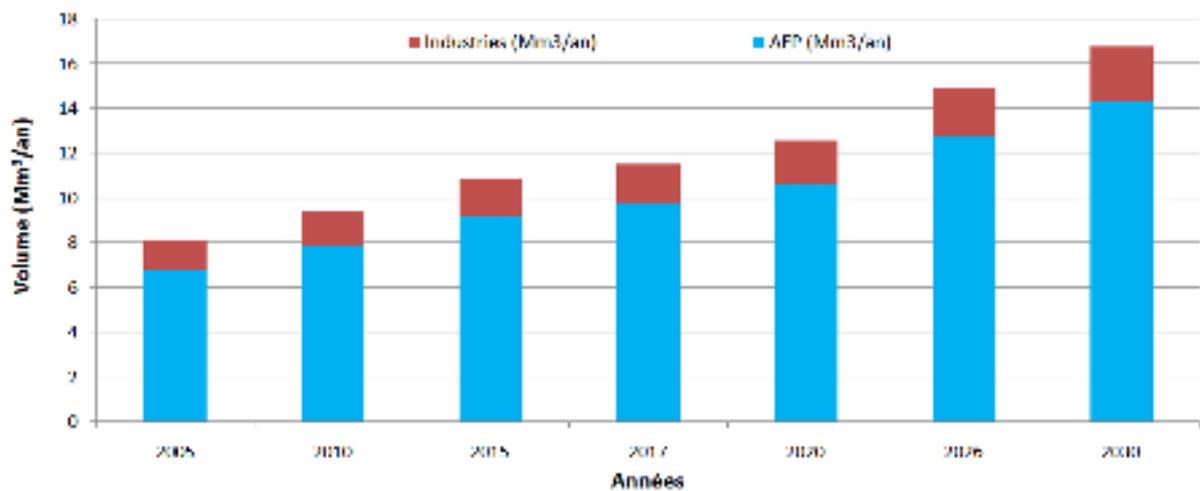


FIGURE 6.13 – Évolution des besoins domestiques et industriels en eau de la région de Maghnia (2005-2030)

TABLE 6.8 – Assolement actuel et proposé du périmètre irrigué de Maghnia

Cultures	Assolement (%)		Surfaces (ha)	
	Existant	Proposé	Existant	Proposé
Céréales	74.4	60	3500	4200
Maraichère	8.5	20	400	1400
Olivier	15.9	15	750	1050
Arbres fruitiers	1.2	5	55	350
<b>Total</b>	<b>100%</b>	<b>100%</b>	<b>4705 ha</b>	<b>7000 ha</b>

tionne l'activité des stomates disposées sur la surface des feuilles, et par conséquent l'évapotranspiration. Cette dernière représente la quasi-totalité (plus de 98%) de l'eau consommée par les plantes. Le reste (moins de 2%) est retenu à l'intérieur pour fabriquer la matière organique des cultures (tiges, feuilles, fruits, ... etc).

Pour déterminer les besoins en eau des cultures du périmètre irrigué de Maghnia on se base sur le travail de (Bensalem and Bouhenna 2017), qui ont réalisé une étude de faisabilité de l'extension du périmètre. Cependant, une autre variante est proposée concernant l'assolement (tableau 6.8) et le rendement du réseau. Par conséquent, une autre estimation des besoins en eau d'irrigation.

L'évaluation des besoins en eau des cultures s'effectue au moyen de l'équation 6.3. On commence par l'estimation de l'évapotranspiration potentielle "ou de référence"  $ET_0$  ou  $ET_p$  (figure 6.14). Cette dernière se réfère à la plante du gazon, et exprime l'influence du climat sur l'évapotranspiration d'une plante donnée autre le gazon. Plusieurs formules permettant le calcul

de  $ET_0$  sont établies, telles que Blaney-Criddle (1945), Penman (1948), Turc (1960) et Penman-Monteith (1965). Cette dernière, implémentée dans le logiciel "Cropwat" est celle utilisée dans ce cas. Le coefficient culturel  $K_c$  renferme les caractéristiques permettant de distinguer une culture à celle de référence "gazon". Ces caractéristiques englobent la hauteur de la culture, la résistance de la surface sol-végétation au mouvement de l'air, l'albédo et l'évaporation depuis le sol. La table 5 (en annexes) annonce les valeurs de coefficients  $K_c$  pour chaque type de culture.

$$B_{net} = ETM - P_u - (X \times RFU) \quad (6.3)$$

Tels que :

- $B_{net}$  : le besoin net en eau de la culture en question (mm) ;
- $ETM$  : l'évapotranspiration maximale de la culture (mm),  $ETM = ETR$  lorsque le coefficient de sécheresse  $K_s = 1$  ;
- $P_u$  : la précipitation utile "ou efficace" (mm), calculée en retranchant les pertes par ruissellement et par percolation profonde de la précipitation de fréquence de 80%,
- $RFU$  : la réserve facilement utilisable par la culture (mm) ;
- $X \times RFU$  : variation de la réserve d'eau du sol (mm), différence entre les quantités d'eau disponible dans la zone racinaire au début et à la fin d'une saison végétative.

Les précipitations mensuelles utiles, les données et les résultats de calcul des valeurs mensuelles de  $ET_0$  sont données au tableau 6 (Annexes).

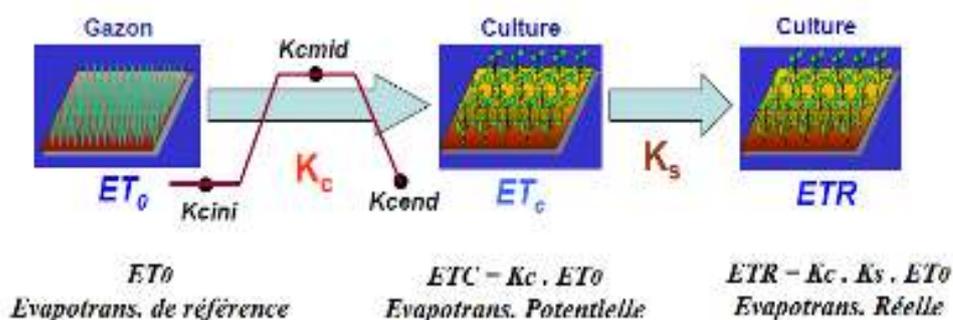


FIGURE 6.14 – Évaluation de l'évapotranspiration réelle des cultures

L'estimation grossière des besoins nets mensuels en eau d'irrigation de l'ensemble des cultures est aperçue sur la figure 6.15 ci-après.

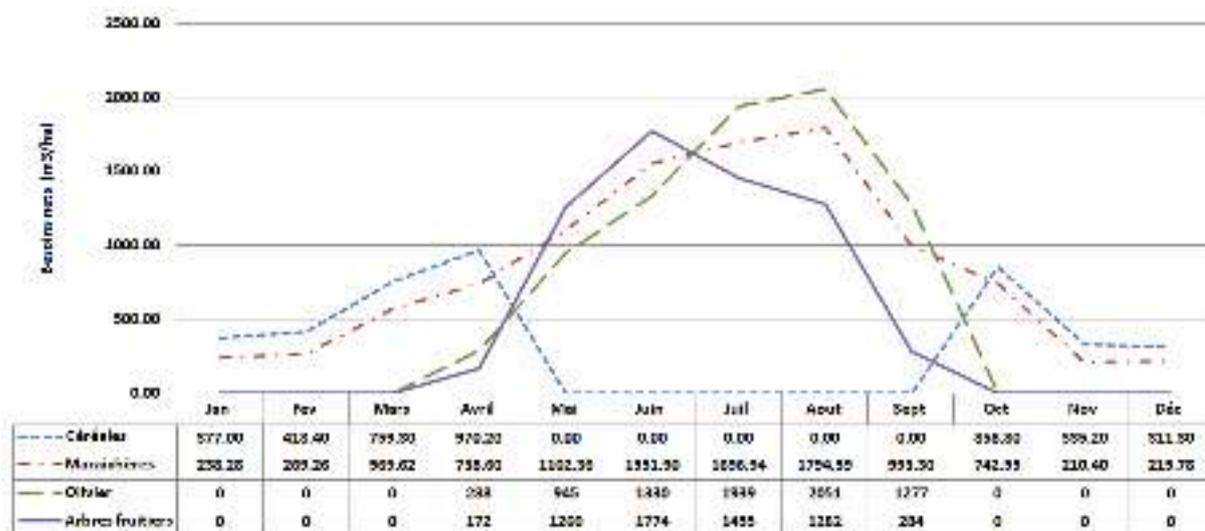


FIGURE 6.15 – Estimation des besoins nets mensuels en eau des cultures de périmètre irrigué de Maghnia ( $m^3/ha$ )

### 6.6.2.2 Estimation des besoins en eau d'irrigation

La modélisation des besoins en eau d'irrigation ne peut être que perfectible, car elle ignore plusieurs facteurs importants qui modulent la demande en eau dans le temps et dans l'espace (exemple, modification des types de culture d'une année à l'autre ou la possibilité de mobiliser des ressources en eau locales ou proches (cours d'eau voisins, puits, bêche de stockage des eaux de pluies, ... etc)). Cette modélisation a comme objectif l'estimation plausible des besoins en eau pour un assolement inter-annuel moyen (François 2013).

Par définition, "le besoin en eau d'irrigation est la quantité d'eau que l'on doit apporter à la culture pour être sûr qu'elle reçoit la totalité de son besoin en eau ou une fraction déterminée de celui-ci. Si l'irrigation est la seule ressource en eau, le besoin en eau d'irrigation sera au moins égale le besoin en eau de la culture et il est souvent plus important en raison de pertes à la parcelle (besoin de lessivage, percolation profonde, inégalité de répartition, etc.)" (Vermeiren and Jobling 1983). À partir de là, les besoins nets et bruts en eau se définissent. Le besoin net ( $B_{net}$ , équation 6.3) représente le volume (ou la hauteur d'eau exprimée en "mm" généralement) théoriquement nécessaire pour obtenir une production normale sans compter les différentes pertes d'eau. Toutefois, le besoin en eau brut ( $B_{brut}$ ) représente le volume (ou la hauteur d'eau) pratiquement nécessaire en tenant comptes les éventuelles pertes (Vermeiren and Jobling 1983).

À partir des définitions précédentes le rendement "ou l'efficience" d'un système d'irrigation s'impose. Pour ce dernier, le rendement global est le rapport entre les quantités d'eau effectivement utilisées et celles dérivées en tête de l'aménagement (CLEMENT et al. 1978).

L'efficience globale d'un système d'irrigation peut être donnée par l'équation suivante (équation 6.4) :

$$R_G = R_R \times R_U \quad (6.4)$$

Tels que :

- $R_G$  : le rendement global ou "l'efficience" (%);
- $R_R$  : le rendement de réseau (le rapport de la qualité de d'eau distribuée en tête de chaque parcelle à la qualité prélevée en tête du réseau) (%);
- $R_U$  : le rendement de l'uniformité d'irrigation (le rapport entre les besoins des plantes et la quantité d'eau distribuée en tête des parcelles) (%).

Considérant que le transport et la distribution sont prévus par des conduites chargées (sous pression), donc  $R_R \cong 95\%$ . Par ailleurs, en tenant compte les différentes techniques d'irrigation pratiquées dans la région (irrigation par surface, par aspersion et l'irrigation localisée), ainsi celles convenables à chaque type de culture, on attribue un rendement de l'uniformité d'irrigation  $R_{U_i}$  à chaque culture. Ensuite, par pondération on estime le rendement  $R_U$  en question (équation 6.5).

$$R_U = \frac{R_{U_i} \times A_i}{A_T} \quad (6.5)$$

Tels que :

- $R_{U_i}$  : rendement de l'uniformité d'irrigation de la culture  $i$ ;
- $A_i$  : surface réservée à la culture  $i$ ;
- $A_T$  : surface globale du périmètre.

La table 6.9 montre les résultats de calcul des différents rendements d'irrigation. Notons que les efficacités des techniques d'irrigation sont tirés depuis (CLEMENT et al. 1978).

Les besoins nets mensuels en eau d'irrigation sont calculés via l'équation 6.3. Ensuite, à la base du rendement global d'irrigation " $R_G$ " (tableau 6.9) les besoins bruts mensuels et volumes mensuels en eau d'irrigation sont déterminés en fonction de surface exploité du périmètre (tableau 6.10).

La table 7 (Annexes) montre les résultats d'évaluation des besoins et volumes. Cette évaluation nous permet d'arriver au volume d'environ  $56.08 \times 10^6 m^3/an = 56.08 Mm^3/an$  pour la surface de 7000ha. Les besoins mensuels maximaux sont aux mois d'avril ( $7.39 Mm^3/mois$ ) et d'août ( $6.91 Mm^3/mois$ ), tandis que les minimaux sont aux mois de novembre et décembre ( $2.30 Mm^3/mois$ ) et ( $2.18 Mm^3/mois$ ) respectivement.

TABLE 6.9 – Rendements du réseau, d'uniformité et global d'irrigation du périmètre de Maghnia

Cultures	Techniques d'irrigation	$R_{U_i}$ (%)	$A_i$ (ha)	$R_{U_i} \times A_i$
Céréales	Aspersion, pivot ...	0.8	4200	3360
Maraichère	Aspersion, bassins, sillons, planches ..	0.6	1400	840
Olivier	Irrigation localisée, par bassins ...	0.9	1050	945
Arbres fruitiers	Irrigation localisée, par bassins ...	0.9	350	315
Rendement de l'uniformité d'irrigation $R_U$				0.78 (%)
Rendement du réseau d'irrigation $R_R$				0.95 (%)
Rendement global d'irrigation $R_G$				0.74 (%)

TABLE 6.10 – Variation des volumes de besoins bruts en eau en fonction de la surface cultivée du périmètre irrigué de Maghnia

Surface (ha)	Mois												Total annuel
	Jan	Fev	Mars	Avril	Mai	Juin	Juil	Aout	Sept	Oct	Nov	Déc	
3500	1.30	1.44	2.69	3.70	2.01	2.83	3.33	3.46	1.91	3.14	1.15	1.09	<b>28.04</b>
4000	1.48	1.65	3.08	4.22	2.30	3.24	3.80	3.95	2.19	3.59	1.31	1.24	<b>32.05</b>
5000	1.85	2.06	3.85	5.28	2.87	4.04	4.75	4.94	2.73	4.49	1.64	1.55	<b>40.06</b>
6000	2.22	2.47	4.62	6.34	3.45	4.85	5.70	5.92	3.28	5.38	1.97	1.86	<b>48.07</b>
7000	2.59	2.88	5.39	7.39	4.02	5.66	6.65	6.91	3.83	6.28	2.30	2.18	<b>56.08</b>

La figure 6.17 dévoile l'estimation des différents besoins en eau à l'horizon de l'année 2026. Cette dernière est choisie pour avoir une projection sur 10 ans (120 mois), comptée à partir la fin de la série chronologique disponible des apports qui s'étend sur la période du septembre 1972 au décembre 2016. Il est trouvé que les demandes mensuelles en eau minimale, maximale et moyenne touchent respectivement  $2.01Mm^3/mois$ ,  $8.59Mm^3/mois$  et  $5.14Mm^3/mois$ .

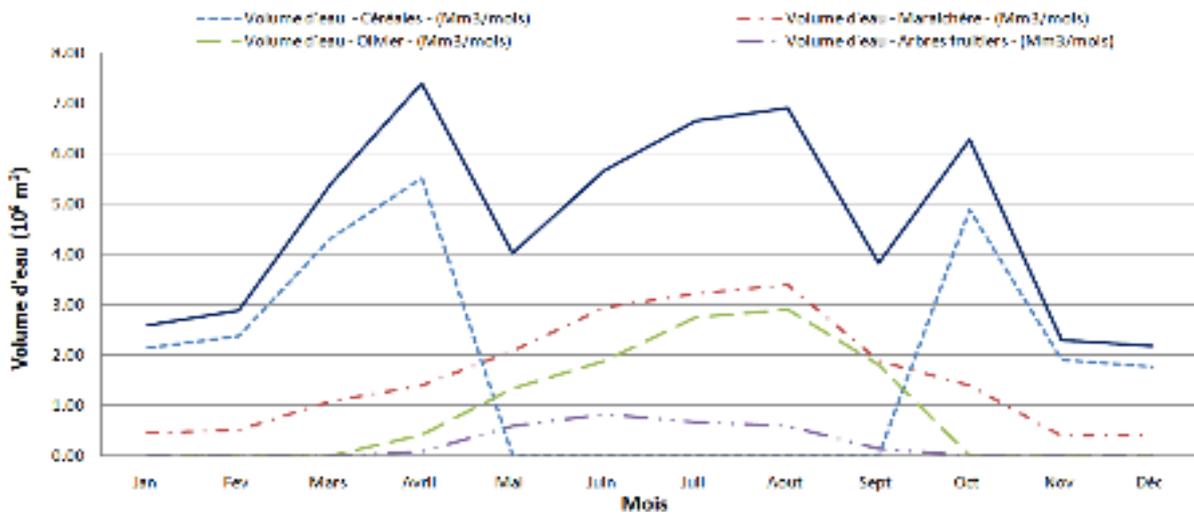


FIGURE 6.16 – Volumes des besoins bruts mensuels de différentes cultures du périmètre irrigué de Maghnia (surface totale 7000 ha)

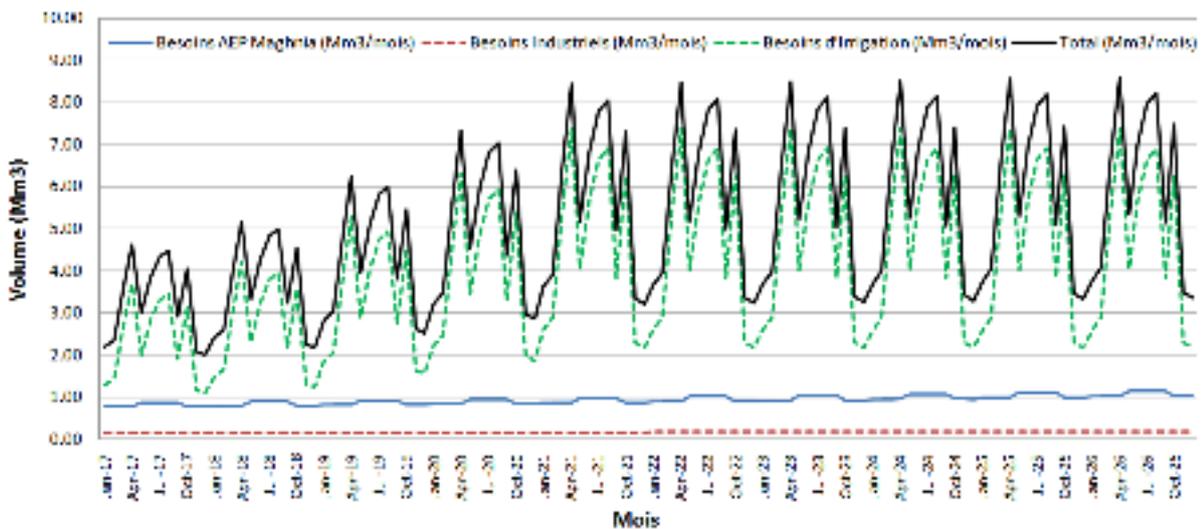


FIGURE 6.17 – Volumes des besoins en eau mensuels domestiques (AEP de la région de Maghnia), industriels et d’irrigation du périmètre agricoles de Maghnia (Jan-2017 – Déc-2026)

## **6.7 Conclusion**

Le barrage Hammam Boughrara, dont la capacité totale atteint  $177Mm^3$ , représente sans doute un potentiel hydraulique dans la région. Le barrage est construit en terres, dans le but de répondre aux besoins en eau domestiques, industriels et agricoles de la région. Au départ, la politique de gestion du réservoir a visé d'affecter  $17 Mm^3/an$  pour l'approvisionnement en eau potable de la région de Maghnia (Maghnia et Hammam Boughrara),  $33Mm^3/an$  pour renforcer l'AEP d'Oran, et  $09 Mm^3/an$  pour soutenir l'irrigation du périmètre agricole de Maghnia.

Suite à la réalisation de nombreuses stations de dessalement sur le littoral des wilayas d'Oran et Tlemcen, les autorités et les services compétents ont pensé à la réaffectation de la ressource en eau au profit de l'irrigation. Dans ce sens, la part d'allocation en eau du périmètre à partir du réservoir Hammam Boughrara est augmentée de  $09Mm^3/an$  pour atteindre  $20Mm^3/an$ . Cela a permis d'étaler la surface irriguée de  $4000ha$  à  $7000ha$  environ.

L'analyse de la série chronologique des apports du réservoir (septembre 1972 - décembre 2016) a montré des fluctuations imposantes. À l'échelle annuelle, la moyenne est de  $53.31Mm^3/an$ , l'écart-type et le coefficient de variation sont conséquents ( $37.78Mm^3$  et  $0.71$  respectivement). A l'échelle mensuelle, la même constatation peut se relever. L'apport moyen mensuel est d'environ  $4.45Mm^3$ , l'écart-type et coefficient de variation s'élèvent à  $8.08Mm^3/mois$  et  $1.82$  respectivement.

La projection des besoins en eau domestiques et industriels de la région de Maghnia à différents horizons, et dans les limites de la politique globale de la gestion, est faite dans le but de concrétiser la fonction objectif de l'optimisation. Cette étape ne peut être que grossière, approximative et incertaine car elle est sujette à plusieurs incertitudes liées aux intervenants du système tout entier. Dans ce contexte et à titre d'exemples, le taux d'accroissement de la population et de développement industriel, ainsi le degré d'exploitation du périmètre agricole et l'assolement ne peuvent être qu'imprécis.

Tout bien considéré, à l'horizon 2026, les demandes en eau domestiques et industrielles sont estimées atteindre d'environ  $12.72Mm^3/an$  et  $2.23Mm^3/an$  respectivement, et continue à la hausse au file des années. De l'autre côté, la mise en service intégral du réseau d'irrigation permet l'exploitation entière du périmètre agricole, et donc la consommation d'un volume estimé de  $56.08Mm^3/an$  proche au volume d'allocation par les deux barrages  $58Mm^3/an$  ( $20Mm^3/an$  par le réservoir Hammam Boughrara et  $38Mm^3/an$  par Béni-Bahdel). Ces volumes sont répartis mensuellement en fonction du régime de consommation de la population et les besoins agricoles mensuels.



# Chapitre 7

## Prévision des apports mensuels du barrage Hammam Boughrara par réseaux de neurones artificiels

### 7.1 Introduction

La prévision des apports liquides des réservoirs est, sans doute, une étape primordiale dans la gestion opérationnelle de ces derniers. Plusieurs approches ont été utilisées pour concrétiser cette phase telles que le modèle Thomas-Fiering (Harms and Campbell 1967 ; Srikanthan 1978), chaînes de Markov (Yapo 1992 ; Marshall et al. 2004), les modèles ARMA et ARIMA (Valipour et al. 2013 ; Sharma et al. 2015), les réseaux de neurones artificiels (Thirumalaiah and Deo 2000 ; Kişi 2004 ; Kişi 2007 ; Akhtar et al. 2009 ; Valipour et al. 2013 ; Darras 2015), machines à vecteurs de support "Support Vector Machine, SVM" (Lin et al. 2009 ; Kalteh 2013), ... etc.

Contrairement aux modèles mathématiques qui revendiquent la connaissance précise de toutes les variables contributives d'un processus donné, les réseaux de neurones artificiels (RNAs) peuvent estimer le comportement d'un processus même avec des informations incomplètes. Encore, les RNAs sont des modèles génériques de type "boîte noire". Cela signifie qu'ils peuvent traiter une large variété de problèmes sans nécessité d'adaptation ou de réimplémentation.

Prédire la réponse de cours d'eau d'un bassin versant est tellement complexe et dure, qu'il n'existe pas un modèle de prévision fiable à 100%. La complexité se multiplie surtout avec la connaissance partielle des phénomènes déroulant à l'intérieur du bassin. Devant ces

contraintes, et vu leurs aptitudes, les réseaux de neurones artificiels peuvent constituer un alternatif judicieux de prévision dans ce cas. L'objectif de cette partie est d'utiliser les RNAs pour prévoir les apports du réservoir Hammam Boughrara, l'étape cruciale avant de procéder à l'optimisation de la gestion du réservoir dans un avenir incertain.

## **7.2 Prévion des apports mensuels**

Selon (Oliveira and Maia 2018), parmi les générateurs stochastiques des séquences de flux des cours d'eau les plus utilisés on trouve : les modèles autorégressifs "AR" (ARMA, ARIMA, ... etc), modèles de bruit de tir "shot noise", les ondelettes "wavelet", les schémas de sangle de démarrage de bloc "Block bootstrap scheme", les modèles de K-plus proches voisins "K-Nearest Neighbors (KNN)", et les réseaux de neurones artificiels (ANNs). Ces derniers sont utilisés dans cette étude tant que des modèles de régression (comme ARMA, ARIMA, KNN ...). Notons que les modèles ARMA sont exclus d'être présentés dans cette étude à cause de mauvais résultats produits dans le cas des apports mensuels du réservoir Hammam Boughrara.

Dans le but de chercher le meilleur réseau de neurones artificiels capable de prévoir les apports du réservoir Hammam Boughrara avec une marge d'erreur acceptable, une étude comparative est proposée dans ce qui suit.

Les différentes méthodes d'apprentissage des RNAs exposées précédemment sont comparées. Les méthodes classiques d'une part (la rétropropagation du gradient d'erreur "BP" et Levenberg-Marquardt "LM"), et d'autre part, les algorithmes évolutionnaires (l'algorithme génétique à codage réel "RGA" et l'algorithme hybride HPSOGWO). Ces méthodes sont appliquées sur des réseaux d'architectures différentes de point de vue nombre d'entrées, nombre de couches cachées et nombre de neurones dans chaque couche. Pour éviter la perte de généralisation des RNAs, le nombre de couches cachées doit être limitée et le plus faible possible.

La comparaison entre les différentes architectures des RNAs et les différents algorithmes d'apprentissage est basée sur l'erreur moyenne absolue " $MAE$ ", la racine de l'erreur moyenne quadratique " $RMSE$ ", le coefficient de corrélation " $R$ ", le coefficient de détermination " $R^2$ ", le critère de Nash-Sutcliffe " $Nash$ ", et l'indice de Willmot " $d$ ".

### **7.2.1 La série chronologique des apports**

La série chronologique observée de l'ensemble des débits d'oued Mouilah et d'oued Tafna du septembre 1972 au décembre 2016 (soient 532 mois ou 44 ans environ) est utilisée. Elle comprend les flux des deux (02) oueds avant la construction du barrage Hammam Bough-

rara, transformés par la suite en apports du barrage dès le commencement de remplissage de son réservoir. Soixante-dix pourcent environ ( $\cong 70\%$ ) de la série est utilisé pour accomplir la phase d'entraînement. Le reste (soit  $\cong 30\%$  environ) est réservé à la phase de validation "ou de test". Dans certains cas, le test et la validation sont séparés, c-à-d., la série des données se compose en trois parties (entraînement, test et validation). Dans cette étude, on opte pour la décomposition en deux parties seulement, en unifiant le test et la validation dans la même série de données d'apports mensuels.

La figure (7.1) démontre la série chronologique de l'ensemble de débits d'oued Mouilah et d'oued Tafna. Les caractéristiques statistiques de la série entière, d'entraînement et de validation sont présentées dans le tableau 7.1.

Les débits peuvent être standardisés dans l'intervalle  $[0,1]$  au moyen de l'équation 7.1. Sous l'effet que le débit minimal mensuel observé est nul ( $Q_{min} = 0Mm^3$ ), la standardisation est effectuée simplement par la division des débits de la série mensuelle sur le débit maximal observé durant la période totale.

$$Q_t = \frac{Q_{obs_t} - Q_{min}}{Q_{max} - Q_{min}} \quad (7.1)$$

Tels que :

- $Q_t$  : débit mensuel standardisé dans l'intervalle  $[0, 1]$  de mois  $t$ ;
- $Q_{obs_t}$  : le débit mensuel observé au mois  $t$  (en  $Mm^3$ );
- $Q_{min}$  et  $Q_{max}$  : sont les débits minimal et maximal respectivement de la série chronologique entière (en  $Mm^3$ ).

Les réseaux de neurones artificiels répondant mieux seront utilisés pour prévoir les apports dans la phase d'optimisation de l'exploitation du réservoir.

TABLE 7.1 – Caractéristiques statistiques de la série temporelle des apports du barrage Hammam Boughrara

Paramètre statistique	Série entière	Série d'entraînement	Série de validation
		(69.92% $\cong$ 70%)	(30.08% $\cong$ 30%)
Nombre d'observations	532	372	160
Min ( $Hm^3/mois$ )	0	0	0.558
Max ( $Hm^3/mois$ )	105.18	105.18	36.348
Moyenne ( $Hm^3/mois$ )	4.445	4.746	3.747
Médiane ( $Hm^3/mois$ )	2.795	2.952	2.150
Écart-type ( $Hm^3/mois$ )	8.082	9.021	5.250
Coefficient de variation (CV)	1.818	1.900	1.401

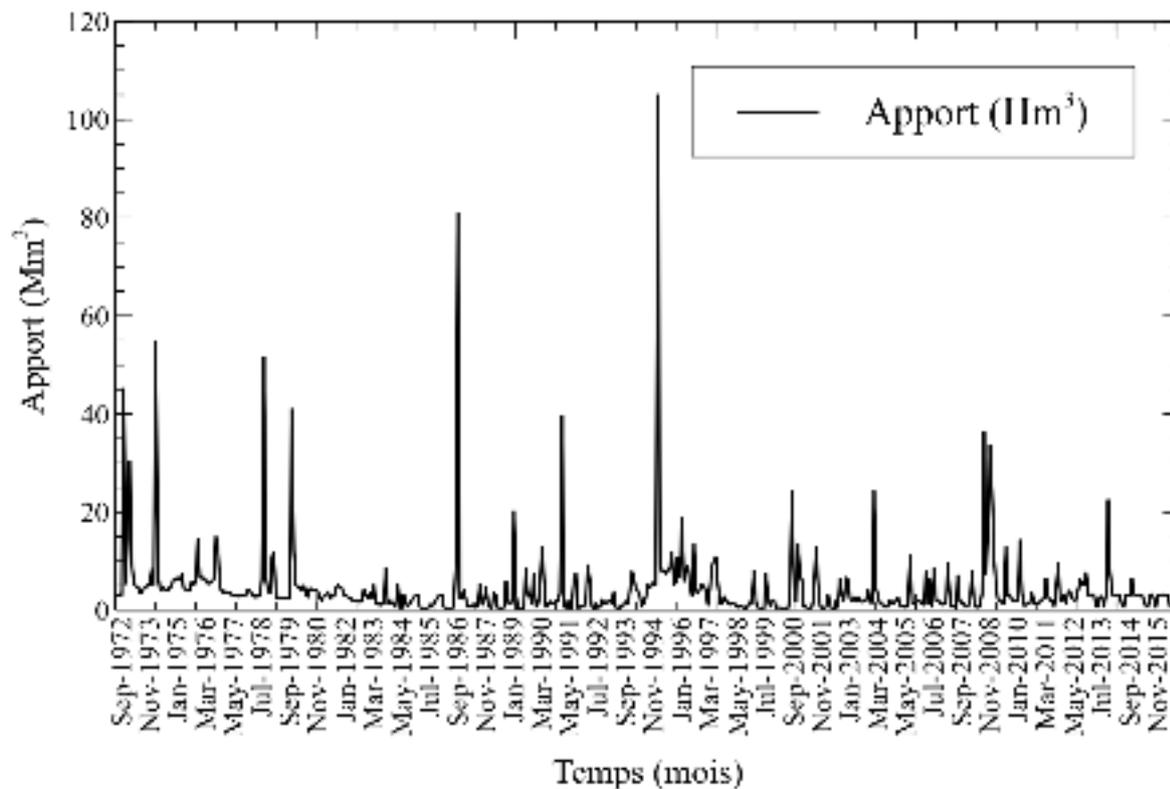


FIGURE 7.1 – Tracé de la série chronologique des apports mensuels du réservoir Hammam Boughrara (septembre 1972 au décembre 2016)

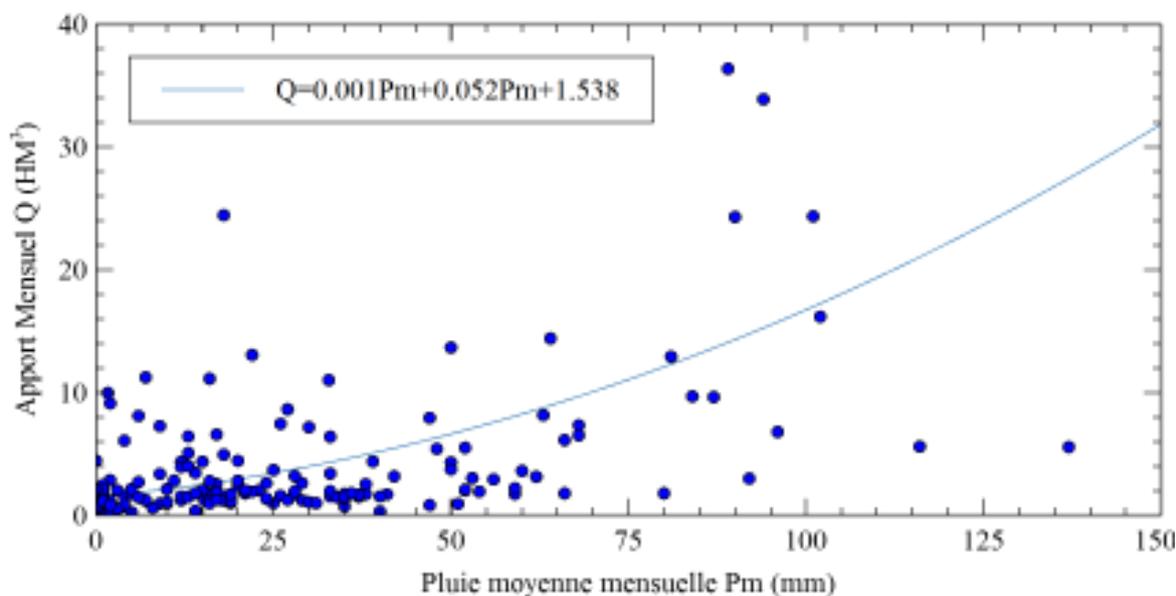


FIGURE 7.2 – Corrélation entre les pluies moyennes mensuelles enregistrées à la station de Maghnia et la série chronologique des apports (janvier 1997 au décembre 2016)

### **7.2.2 Critères d'évaluation des résultats**

Les résultats d'entraînement et de validation issus des réseaux de neurones artificiels sont évalués au moyen de divers critères. Une analyse visuelle de confusion et de superposition des résultats calculés avec les valeurs mesurées permet, dans un premier temps, une évaluation primaire des performances des modèles. Cependant, le recours aux critères objectifs d'évaluation est nécessaire. Ces critères sont des fonctions mathématiques des erreurs permettant de mesurer le déphasage entre les valeurs mesurées et celles estimées.

Dans cette étude, les critères usuels fréquemment utilisés dans les cas similaires sont pris en considération, tels que le coefficient de détermination  $R^2$ , le critère de Nash "ou Nash-Sutcliffe" (Nash and Sutcliffe 1970), et l'indice de Willmott ( $d$ ) ou "Index of agreement  $d$ " (Willmott 1981. Selon (Krause et al. 2005), chacun de ces critères a des avantages et des inconvénients spécifiques qui doivent être pris en compte lors de l'étalonnage ou de l'évaluation des modèles.

#### **7.2.2.1 L'erreur moyenne absolue "Mean Absolute Error" (MAE)**

Elle est donnée par la formule suivante :

$$MAE = \frac{1}{N_s} \sum_{i=1}^{N_s} |x_i - y_i| \quad (7.2)$$

Tels que :

- $x_i$  : valeur observée d'indice  $i$  ;
- $y_i$  : valeur calculée par le modèle de prévision d'indice  $i$  ;
- $N_s$  : la taille de la série chronologique.

#### **7.2.2.2 La racine de l'erreur moyenne quadratique "Root Mean Square Error" (RMSE)**

Elle est donnée par la formule suivante :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_s} (x_i - y_i)^2}{N_s}} \quad (7.3)$$

#### **7.2.2.3 Le coefficient de corrélation ( $R$ )**

La plage du  $R$  se situe entre 0 et 1, ce qui décrit la quantité de dispersion entre les valeurs observées et celles calculées (ou prévues dans ce cas). Une valeur de zéro "0" signifie

aucune corrélation, alors qu'une valeur de un "1" signifie que la dispersion de la prédiction est égale à celle de l'observation.

$$R = \frac{\sum_{i=1}^{N_s} (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N_s} (x_i - \bar{X})^2 \sum_{i=1}^{N_s} (y_i - \bar{Y})^2}} \quad (7.4)$$

Tels que :

- $\bar{X}$  : la moyenne des valeurs observées ;
- $\bar{Y}$  : la moyenne des valeurs calculées (prévues par le modèle).

#### **7.2.2.4 Le coefficient de détermination ( $R^2$ )**

C'est le carré du coefficient de corrélation ( $R$ ). Comme ce dernier, le coefficient  $R^2$  varie entre 0 et 1. Une valeur de 0 du  $R^2$  signifie aucune corrélation, tandis qu'une valeur de 1 marque une corrélation parfaite. Selon (Krause et al. 2005), le coefficient de détermination et le critère de Nash-Sutcliffe sont les plus fréquemment utilisés dans les travaux de prévision des débits des cours d'eau et des apports des réservoirs, et sont très sensibles aux débits de pointe, au détriment d'une meilleure performance dans les conditions de faible débit.

$$R^2 = \left( \frac{\sum_{i=1}^{N_s} (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N_s} (x_i - \bar{X})^2 \sum_{i=1}^{N_s} (y_i - \bar{Y})^2}} \right)^2 \quad (7.5)$$

#### **7.2.2.5 Le critère de Nash-Sutcliffe**

Il est proposé par (Nash and Sutcliffe 1970) et communément appelé "critère de Nash". Ce critère est largement utilisé en études de modélisation hydrologique. Il exprime la proportion de la variance des sorties expliquée par le modèle hydrologique. Ce critère varie entre moins l'infini " $-\infty$ " et un "1". De même que le coefficient  $R^2$ , les résultats seront d'autant meilleurs que le critère de Nash se rapproche de un "1". Tandis qu'une valeur du critère inférieure à zéro, (c-à-d.,  $Nash \in ] - \infty, 0]$ ) indique que la moyenne des valeurs observées de la série est un meilleur prédicteur que le modèle. Le critère de Nash seul ne peut donc suffire pour déterminer la qualité d'un modèle de prévision (Darras 2015). L'équation (7.6) démontre la formule de calcul de critère de Nash.

$$Nash = 1 - \frac{\sum_{i=1}^{N_s} (x_i - y_i)^2}{\sum_{i=1}^{N_s} (x_i - \bar{X})^2} \quad (7.6)$$

### 7.2.2.6 L'indice de Willmot $d$ "Index of agreement"

Il est introduit par (Willmott 1981). De même que  $R^2$  et le critère de Nash, l'indice  $d$  est très sensible aux débits de pointe. Cet indice est donné par la formule suivante :

$$d = 1 - \frac{\sum_{i=1}^{N_s} (x_i - y_i)^2}{\sum_{i=1}^{N_s} (|x_i - \bar{X}| + |y_i - \bar{X}|)^2} \quad (7.7)$$

### 7.2.3 Organisation des données d'entraînement des RNAs

Au cours de modélisation mathématique d'un processus physique, deux ensembles d'informations sont essentiels pour garantir des résultats corrects et précis. Tout d'abord, il y a le nombre de paramètres "variables" impliqués dans le processus; deuxièmement, il y a la connaissance précise des interrelations entre eux, c-à-d., peu importe la complexité et la non-linéarité entre les différents paramètres. Lorsque les réseaux neuronaux sont utilisés pour construire une fonction qui estime le comportement du processus, une connaissance complète de ces deux facteurs (tous les paramètres impliqués et leurs interrelations) n'est pas une nécessité absolue (Kişi 2004).

En se basant sur le tracé de la fonction d'autocorrélation simple (ACF) de la série chronologique des apports mensuels (figure 7.3), la dépendance entre les débits n'est pas significative. Au-delà de décalage du 3 mois, la corrélation est moins de 10% sauf pour le décalage de 14 mois. À cet effet, plusieurs combinaisons entre les débits sont essayées dans le but de trouver une meilleure association entre eux.

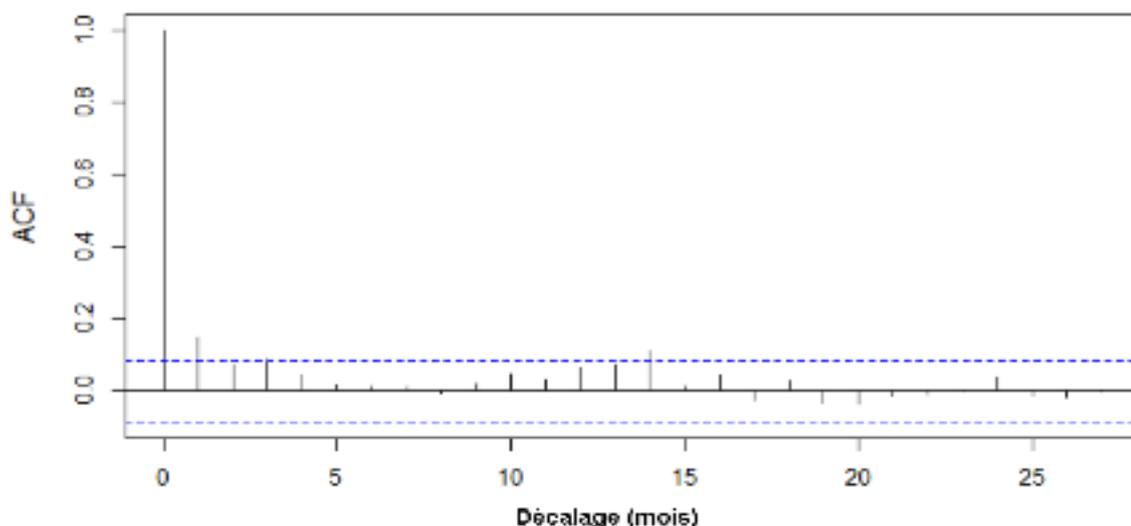


FIGURE 7.3 – Tracé de la fonction d'autocorrélation partielle simple de la série entière de débits (septembre 1972 au décembre 2016)

Les entrées d'entraînement sont organisées selon les modèles de régression apparus dans le tableau 7.2. Pour le modèle ANN-Algo-1 par exemple, le RNA utilise la méthode d'apprentissage notée "Algo" qui peut être : "BP, LM, RGA ou HPSOGWO" selon le cas, et prévoit l'apport mensuel  $Q_{c_t}$  du mois  $t$  en fonction de l'apport de mois précédent  $Q_{t-1}$  et l'indice du mois  $t$ . Ce dernier se calcule par la standardisation de rang du mois (variant de 1 à 12) dans l'intervalle  $[0, 1]$ . De même, pour le modèle ANN-Algo-4, il estime l'apport mensuel  $Q_{c_t}$  (du mois  $t$ ) en fonction des apports mensuels des quatre (04) mois qui le précèdent ( $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$ ,  $Q_{t-4}$ ) et l'indice du mois  $t$ . Le temps est considéré dans ces modèles de régression afin d'inclure l'information sur la saisonnalité des apports.

TABLE 7.2 – Modèles de régression et organisation de la série chronologique des apports en entrées et sorties

Modèle	ANNs	Entrées (inputs)	Sortie (output)
1	ANN-Algo <sup>a1b</sup>	$Q_{t-1}$ , $t_s^c$	$Q_{c_t}^d$
2	ANN-Algo-2	$Q_{t-1}$ , $Q_{t-2}$ , $t_s$	$Q_{c_t}$
3	ANN-Algo-3	$Q_{t-1}$ , $Q_{t-2}$ , $Q_{t-3}$ , $t_s$	$Q_{c_t}$
4	ANN-Algo-4	$Q_{t-1}$ , $Q_{t-2}$ , $Q_{t-3}$ , $Q_{t-4}$ , $t_s$	$Q_{c_t}$
5	ANN-Algo-5	$Q_{t-1}$ , $Q_{t-2}$ , $Q_{t-3}$ , $Q_{t-4}$ , $Q_{t-5}$ , $t_s$	$Q_{c_t}$
6	ANN-Algo-6	$Q_{t-1}$ , $Q_{t-2}$ , $Q_{t-3}$ , $Q_{t-4}$ , $Q_{t-5}$ , $Q_{t-6}$ , $t_s$	$Q_{c_t}$

<sup>a</sup> Algorithme d'apprentissage (BP, LM, RGA ou HPSOGWO).

<sup>b</sup> Nombre de débits d'apports précédents, utilisés comme entrées de réseau de neurones artificiels.

<sup>c</sup> Indique l'indice de mois considéré par la prévision (varie entre 1 à 12 et standardisé dans l'intervalle  $[0, 1]$ ).

<sup>d</sup> Débit de sortie (calculé) au mois considéré  $t$ .

Pour chaque méthode d'apprentissage, les RNAs testés sont classés en deux (02) catégories de point de vue nombre de couches cachées :

1. RNAs d'une (01) seule couche cachée, c-à-d., une (01) couche d'entrée, une (01) couche cachée, et une (01) de sortie ;
2. RNAs de deux (02) couches cachées, c-à-d., une (01) couche d'entrée, deux (02) cachées, et une (01) de sortie.

Ce choix est basé sur plusieurs études de prévision des débits des cours d'eau telles que (Thirumalaiah and Deo 2000 ; Kişi 2004 ; Kişi 2007 ; Akhtar et al. 2009), dans lesquelles les réseaux de neurones d'une seule couche cachée ont prouvé des bons résultats.

Les structures des réseaux de neurones artificiels testés sont présentées dans le tableau 7.3. Par exemple, le réseau "ANN-LM-3 (6)" est un réseau de neurones entraîné par l'algorithme de Levenberg-Marquardt "LM". Il exige trois (03) apports mensuels à l'entrées ( $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$ ) et l'indice du mois  $t$  pour donner une seule sortie calculée ( $Q_{c_t}$ ). En outre, il est doté d'une seule couche cachée composée de six (06) neurones (ou nœuds). De même pour le réseau "ANN-RGA-3(6, 3)", mais cette fois, le réseau est entraîné par l'algorithme génétique à codage réel "RGA", et composé de deux (02) couches cachées. La première est de six (06) neurones (nœuds), tandis que la deuxième est de trois (03) neurones.

Plusieurs structures de réseaux de neurones artificiels ont été évaluées à fin de chercher un meilleur RNA capable de prévoir les apports du barrage Hammam Boughrara avec une bonne précision. Pour chaque algorithme d'apprentissage, 54 RNAs d'une seule couche cachée sont testés. Ces RNAs sont formés en variant d'un côté le nombre d'entrées (d'un seul apport mensuel ( $Q_{t-1}$ ) à six (06) apports mensuels " $Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}, Q_{t-6}$ "), et d'autre côté le nombre de neurones de la couche cachée. Conformément au tableau 7.3, les neurones cachés peuvent varier de 1 à 15. De même pour les RNAs à deux (02) couches cachées, 726 RNAs ont été évalués pour chaque algorithme d'apprentissage. Dans ce cas aussi, le nombre d'entrées varie de 1 à 6, et le nombre de neurones des couches cachées varie de 1 à 15. La sortie est toujours une seule, représentée par l'apport mensuel calculé " $Q_{c_t}$ " de mois  $t$ .

Le tableau 7.4 montre les méthodes d'apprentissage et les fonctions d'activation utilisées ainsi leurs paramètres. Selon (Thirumalaiah and Deo 1998 ; Kişi 2004) la fonction d'activation sigmoïde peut donner des bonnes approximations. Encore, selon (Kişi 2007), les nombres d'itérations  $K_{max} = 2 \times 10^4$  et  $K_{max} = 50$  sont suffisants pour donner des meilleurs résultats d'apprentissage par les algorithmes de rétropropagation "BP" et Levenberg-Marquardt "LM" respectivement.

TABLE 7.3 – Structures des RNAs utilisés pour la prévision des apports

Réseaux de Neurones	Nombre de couches cachées	Nœuds de la couche 1 " $HL_1$ "	Nœuds de la couche 2 " $HL_2$ "
RNA-Algo <sup>a</sup> - I <sup>b</sup> ( $HL_1^c$ )	1	1, 2, ... 15	//
RNA-Algo-I ( $HL_1, HL_2^d$ )	2	1, 2, ... 15	1, 2, ... 15

<sup>a</sup> L'algorithme d'apprentissage utilisé, peut être : BP, LM, RGA ou HPSOGWO.

<sup>b</sup> Nombre de débits d'apports mensuels utilisés comme entrées de réseau de neurones artificiels (1, 2, ..., 6).

<sup>c</sup> Le nombre de nœuds (neurones) dans la première (ou la seule) couche cachée.

<sup>d</sup> Le nombre de nœuds dans la deuxième couche cachée.

TABLE 7.4 – Paramètres des algorithmes d'apprentissage et fonctions d'activation

<i>Algorithme BP</i>						
Nombre d'itérations		Fonction d'activation			$\alpha_{activ}$	
20000		Sigmoïde			2	
<i>Algorithme LM</i>						
Nombre d'itérations		Fonction d'activation			$\alpha_{activ}$	
50		Sigmoïde			2	
<i>RGA</i>						
Nbr d'itérations <sup>a</sup>	Population	Sélection	Croisement	Mutation	Fn. d'activ <sup>b</sup>	$\alpha_{activ}$
1000	70	Élitisme	0.75	0.25	Sigmoïde	2
<i>HPSOGWO</i>						
Nbr d'itérations <sup>a</sup>	Population	$C_1$	$C_2$	$C_3$	Fn. d'activation <sup>b</sup>	$\alpha_{activ}$
1000	40	0.5	0.5	0.5	Bipolaire Sigmoïde	2

<sup>a</sup> Nombre d'itérations.

<sup>b</sup> Fonction d'activation.

#### 7.2.4 Résultats d'apprentissage et de validation des RNAs

Les tableaux 8 et 9 (Annexes) montrent les résultats d'évaluation des meilleurs réseaux de neurones artificiels pour chaque algorithme d'apprentissage et modèle de régression. Les tableaux montrent respectivement les résultats des RNAs à une seule et à deux couches cachées.

##### 7.2.4.1 Réseaux de neurones mono-couche cachée

À partir le tableau 8 (Annexes), il est claire que les valeurs des critères d'évaluations trouvées sont loin d'être satisfaisantes pour l'ensemble des réseaux de neurones en considérant les deux (02) phases d'apprentissage et de validation.

Tout d'abord, pour les meilleurs réseaux de neurones entraînés par la méthode de rétropropagation de l'erreur "BP", pas de changement significatif des résultats en variant le modèle de régression des apports d'un côté, et le nombre de neurones de la seule couche cachée de

l'autre côté. Les RNAs mieux répondant dans ce cas, sont dotés d'un nombre faible de neurones cachés (02 dans la majorité des cas). En premier lieu, les critères  $d$ ,  $Nash$ ,  $R$  et  $R^2$  de la phase d'apprentissage n'ont pas dépassé 0.35, 0.112, 33.8% et 0.114 respectivement. Ce qui indique le sous apprentissage des réseaux de neurones artificiels dans ce cas. Le sous apprentissage a induit une sous estimation des apports. En revanche, les valeurs des différents critères de la phase de validation sont meilleures par rapport à la phase d'apprentissage. Cela, peut être expliqué par l'ampleur des apports mensuels de la série d'entraînement qui sont plus importants que de la série de validation (figure 7.1). Concernant les racines des erreurs moyennes quadratiques  $RMSE$  de la phase d'apprentissage et de validation, elles sont légèrement inférieures aux écart-types des séries chronologiques des apports d'entraînement ( $9.021Mm^3$ ) et de validation ( $5.25Mm^3$ ) respectivement.

Les réseaux de neurones artificiels entraînés par l'algorithme Levenberg-Marquardt ont présenté des valeurs semblables à ceux entraînés par l'algorithme BP, sauf les deux (02) réseaux de neurones "ANN-LM-4(5)" et "ANN-LM-5(8)". Ce dernier est composé de huit (08) neurones cachés et exige cinq (05) apports ( $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$ ,  $Q_{t-4}$ ,  $Q_{t-5}$ ) pour prévoir l'apport  $Q_t$ . Ce réseau de neurones a prouvé une amélioration significative des critères d'évaluation de la phase d'apprentissage, où les indices  $d$ ,  $Nash$ ,  $R$  et  $R^2$  ont atteint 0.95, 0.79, 0.92 et 0.84 respectivement, ainsi,  $MAE$  et  $RMSE$  sont les plus faibles par des valeurs  $\cong 2.52Mm^3/mois$  et  $4.02Mm^3/mois$ . Cette amélioration dans la phase d'apprentissage a engendré une dégradation des mêmes critères dans la phase de validation. Les valeurs des indices d'évaluation sont les plus défavorables parmi celles des autres réseaux de neurones artificiels. Ces résultats peuvent indiquer un sur-apprentissage de ce réseau de neurones. La série des apports mensuels d'entraînement est celle qui présente les débits les plus forts. En conséquence, une amélioration dans cette phase des critères  $Nash$ , et  $R^2$ , qui sont plus sensibles aux pics, a engendré la dégradation des critères par rapport aux apports de la phase de validation. Car, cette dernière est caractérisée par des apports plus faibles.

Concernant les réseaux de neurones entraînés par les métaheuristiques RGA et HPSOGWO, les résultats du premier (RGA) se ressemblent aux RNAs entraînés par les méthodes BP et LM, tandis que ceux du dernier (HPSOGWO), les résultats sont mauvais par rapport aux autres algorithmes sauf les coefficients de corrélation  $R$ , détermination  $R^2$  et  $RMSE$  dans la phase de validation. En outre, pas de changement significatif des résultats en changeant le modèle de régression dans ce cas. Donc, l'algorithme converge vers les mêmes poids synaptiques.

D'autres remarques peuvent être soutirées à partir des résultats :

- Les valeurs des indices d'évaluation des résultats de validation sont meilleures par rapport aux ceux d'entraînement, sauf quelques RNAs entraînés par la méthode de Levenberg-Marquardt.
- Les valeurs des  $RMSE$  d'entraînement et de validation sont de l'ordre de la

déviations standard "écart-type" des séries d'entraînement ( $SD$  [série d'entraînement]= $9.021 Mm^3$ ) et de validation ( $SD$  [série de validation] =  $5.25 Mm^3$ ) respectivement.

- Dans la majorité des cas, pas de variation significative des résultats même avec le changement des structures des réseaux de neurones artificiels.

D'une manière générale, les résultats montrent que les réseaux de neurones testés n'ont pas produit des débits estimés suffisamment proches aux débits observés. Cette divergence est peut-être due à la forte variation des débits d'un mois à l'autre et d'une année à l'autre dans les zones semi-arides telles que le cas étudié. Cette variation intensive est confirmée par la déviation standard ( $SD$ ) et le coefficient de variation ( $CV$ ) (tableau 7.1) qui sont de l'ordre de  $8.08 Mm^3$  et  $1.82$  respectivement pour la série des débits entière,  $9.02 Mm^3$  et  $1.9$  pour la série d'entraînement et  $5.25 Mm^3$  et  $1.4$  pour la série de validation. De plus, le tracé de la fonction d'autocorrélation simple (figure 7.3) confirme ces indicateurs, dont il relève une corrélation très faible entre les apports affectant la qualité de régression.

#### **7.2.4.2 Réseaux de neurones à deux couches cachées**

Le tableau 9 (Annexes) montre les valeurs des critères d'évaluation des réseaux de neurones à deux (02) couches cachées. L'entraînement dans ce cas est effectué par les mêmes algorithmes, paramètres et données qu'auparavant.

Commençant par les RNAs entraînés par la méthode BP. Les valeurs des différents critères d'évaluation utilisés dans cette étude sont légèrement améliorés par rapport à ceux dotés d'une seule couche cachée. Mais, cette amélioration reste insignifiante que ce soit pour la phase d'entraînement ou du test.

En ce qui concerne les RNAs entraînés par la méthode LM, les modèles de régression consommant trois (03) apports mensuels ou plus, ont prouvé des résultats remarquables notamment le RNA-LM-3(6, 15). Ce dernier exige trois (03) apports mensuels dans la couche d'entrées (plus l'indice de mois cible de prévision standardisé dans l'intervalle  $[0, 1]$ ) et doté de six (06) neurones artificiels dans la première couche cachée et quinze (15) neurones dans la deuxième couche cachée. Dans la phase d'apprentissage, ce RNA a démontré  $d \cong 0.9$ ,  $Nash \cong 0.67$ ,  $R \cong 82\%$  et  $R^2 \cong 0.67$ . Par ailleurs, l'erreur moyenne absolue  $MAE \cong 2.68 Mm^3/mois$  et la racine de l'erreur moyenne quadratique  $RMSE \cong 5.19 Mm^3/mois$ , sont les plus faibles dans cette catégorie des RNAs. Parallèlement, dans la phase de test, les résultats sont meilleurs en considérant l'ensemble des critères à la fois. D'une manière générale, c'est le réseau de neurones présentant les meilleurs résultats en considérant les deux phases d'entraînement et de validation.

Concernant les RNAs entraînés par les métaheuristiques (RGA et HPSOGWO), une amélioration est observée pour les réseaux de neurones entraînés par l'algorithme RGA par rapport à ceux dotés d'une seule couche cachée et entraînés par le même algorithme (section précédente) d'une part, et ceux entraînés par les algorithmes BP et HPSOGWO d'autre part. À titre d'exemple, le réseau de neurones ANN-RGA-6(9, 9) a affirmé des résultats meilleurs à ceux entraînés par BP et HPSOGWO et quelques réseaux entraînés par LM dans la phase d'apprentissage. Toutefois, il indique des résultats qui se rapprochent à ceux des meilleurs entraînés par les autres algorithmes.

En dernier lieu, la confrontation des résultats des RNAs entraînés par le métaheuristique HPSOGWO avec les résultats des autres RNAs indique une légère amélioration pour certains modèles et une dégradation pour d'autres. D'une manière générale, l'ensemble des RNAs ont souffert d'un sous-apprentissage.

Finalement, à travers de l'analyse des structures des réseaux de neurones artificiels testés dans cette étude, on remarque clairement que les mieux adaptés disposent dans la première couche cachée un nombre de neurones inférieur ou égale à ce de la deuxième couche. De plus, le modèle de régression le mieux répondant est celui basé sur trois (03) apports mensuels ( $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$ ) pour prévoir le quatrième ( $Q_t$ ) qui les succède.

La figure 7.4 montre la comparaison graphique entre les apports mensuels observés (utilisés pour accomplir la phase d'apprentissage) d'une part, et d'autre part les apports mensuels calculés par les réseaux de neurones artificiels dont le coefficient de corrélation dépasse 50% dans la phase d'apprentissage. Les autres réseaux de neurones ont été ignorés car ils ont souffert du phénomène de sous-apprentissage.

À partir de cette figure, on observe des fluctuations des apports calculés par les deux RNAs autour des apports observés inférieurs à  $10Mm^3/mois$  environ. En précisant que l'ampleur des fluctuations est plus importante pour le "ANN-RGA-6 (9, 9)". Concernant les apports dans l'intervalle  $10Mm^3/mois$  à  $20Mm^3/mois$ , ils sont fréquemment ratés par les deux (02) réseaux de neurones notamment le "ANN-LM-3 (6, 15)". En revanche, pour les apports très forts, ce dernier réseau de neurones a comporté mieux que le "ANN-RGA-6 (9, 9)", où il a pu détecté la moitié de ces apports.

La figure 7.5 démontre la comparaison entre les apports observés et calculés dans la phase de validation. Les deux réseaux de neurones artificiels "ANN-LM-3 (6, 15)" et "ANN-RGA-6 (9, 9)" ont légèrement surestimé les débits inférieurs à  $10Mm^3/mois$ , mais ont pu détecter la saisonnalité des apports dans la plupart des cas. Pour les apports forts, ils sont ratés dans certains périodes, avec une surestimation par le réseau de neurones "ANN-LM-3 (6, 15)" et une sous-estimation par l'autre "ANN-RGA-6 (9, 9)".

### **7.2.5 Prédiction par réseaux de neurones artificiels**

La figure 7.6 dévoile le tracé graphique des résultats de prédiction des apports par les meilleurs réseaux artificiels (ANN-LM-3(6, 15) et ANN-RGA-6 (9, 9)) présentés dans les sections précédentes. Une période de 120 mois (soit 10 ans) est couverte. Les apports sont injectés d'une façon séquentielle pour produire d'autres prévus. Ces derniers (apports prévus) sont confrontés avec les moyennes mensuelles interannuelles calculées à la base de la série chronologique entière (septembre - 1972 au décembre - 2016).

Tout d'abord, une divergence est remarquée entre les apports prévus par les deux réseaux de neurones artificiels (ANN-LM-3 (6, 15) et ANN-RGA-6(9, 9)). Les apports prévus par ce dernier sont largement faibles par rapport à ceux prévus par le premier, mais ils sont proches aux apports moyens mensuels interannuels. En revanche, les apports prévus par les deux modèles ont présenté des signaux répétitifs caractérisés par une saisonnalité.

Pour le réseau de neurones artificiels entraîné par l'algorithme de Levenberg-Marquardt "ANN-LM-3 (6, 15)", les apports augmentent à la fin de la saison hivernale pour atteindre environ  $25Mm^3/mois$  sur presque la totalité de la période de prédiction. Ce qui représente environ trois fois ( $\times 3$ ) l'apport moyen maximal interannuel enregistré. Les apports diminuent, ensuite, rapidement durant la saison printanière simultanément avec les moyens mensuels en se rapprochant ces derniers. Puis, ils tendent à la hausse progressivement durant la saison estivale et automnale. Cette situation, peut être due à la surestimation des apports remarquée dans la phase d'apprentissage et de validation du réseau de neurones en question.

En ce qui concerne les apports mensuels prévus par le réseau de neurones artificiels "ANN-RGA-6 (9, 9)". Ils se rapprochent grossièrement aux apports moyens interannuels. Cependant, ils diminuent pendant la saison hivernale et augmente progressivement pour avoir le pic au mois de juin, puis tendent à la diminution progressive par la suite en suivant une distribution cyclique le long de la période de prédiction. Ces apports prévus présentent une allure contradictoire avec celle des apports moyens mensuels interannuels.

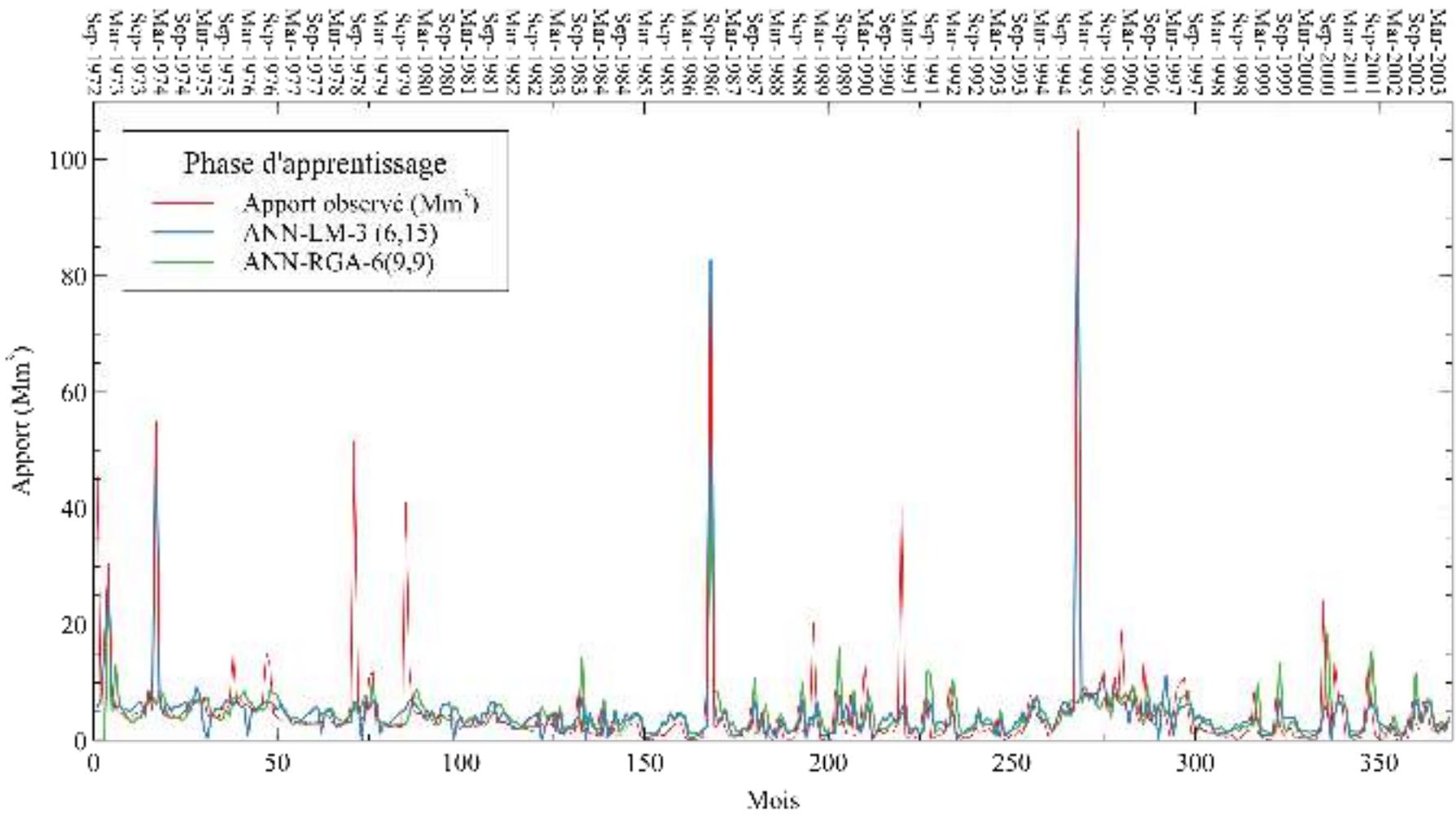


FIGURE 7.4 – Comparaison des apports observés et ceux calculés par les meilleurs RNAs (Phase d'apprentissage)

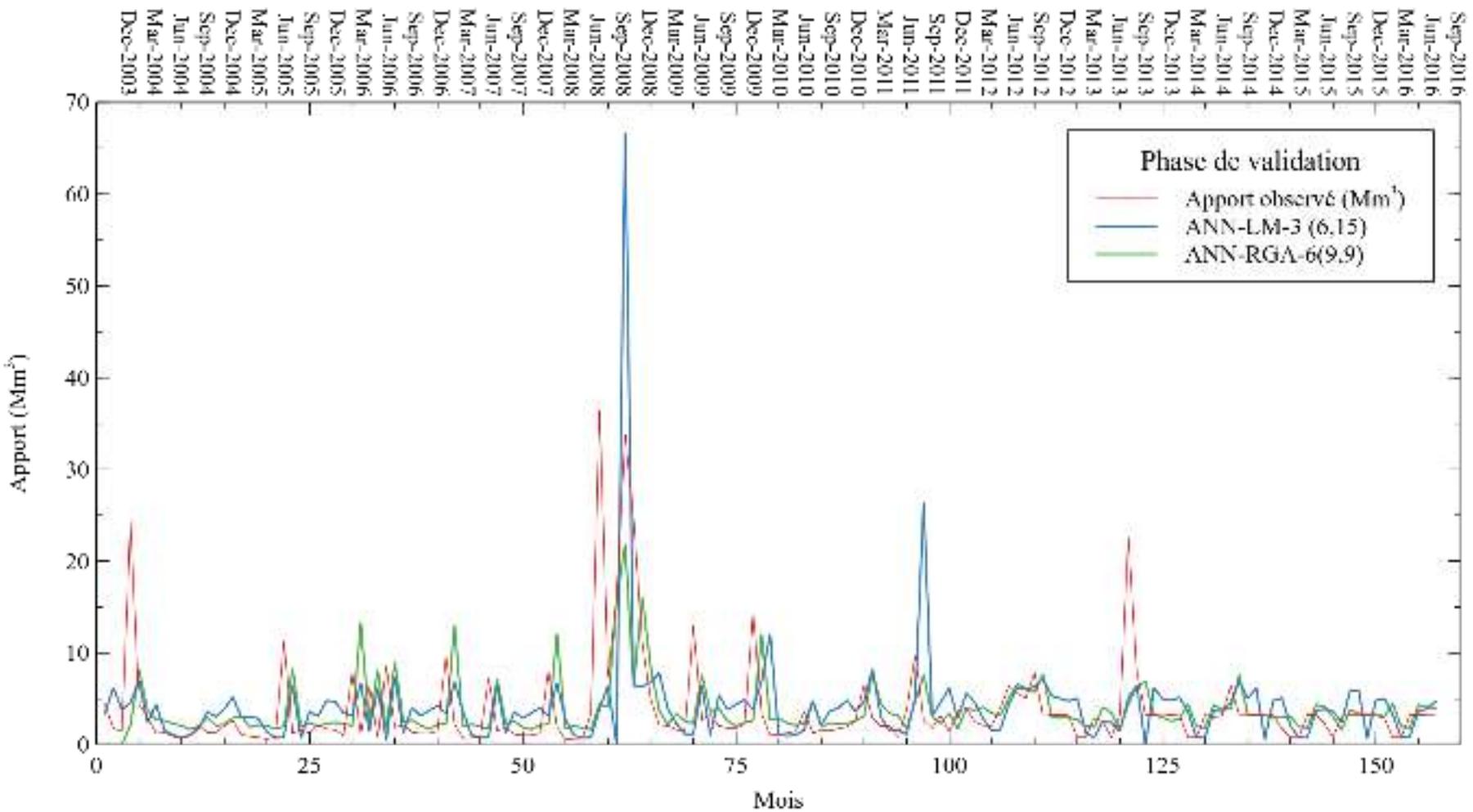


FIGURE 7.5 – Comparaison des apports observés et ceux calculés par les meilleurs RNAs (Phase de validation)

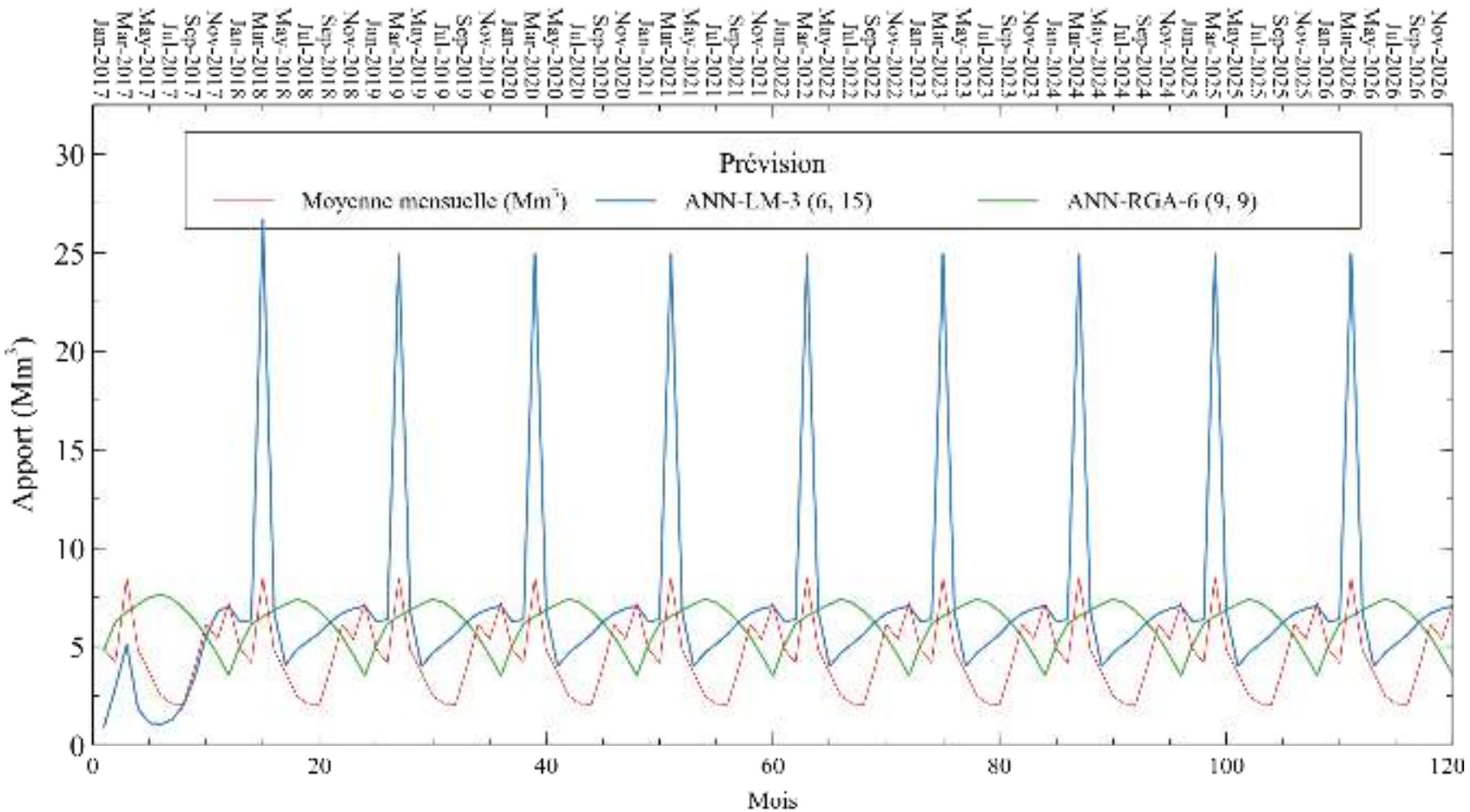


FIGURE 7.6 – Prévision des apports par les meilleurs réseaux de neurones artificiels (les apports moyens mensuels interannuels sont calculés à la base de la série chronologique entière (septembre - 1972 au décembre - 2016))

### **7.3 Conclusion**

Ce chapitre a été consacré à l'utilisation des réseaux de neurones artificiels "RNAs" (présentés dans le chapitre 3) pour prévoir les apports mensuels du barrage Hammam Boughrara (W. Tlemcen) via six modèles de régression.

La série chronologique des apports mensuels de la période du septembre 1972 au décembre 2016 a été utilisée dans cette étude. Une portion de 70% de la série a été réservée à l'apprentissage des réseaux de neurones, tandis que le reste a été conservé à leurs validations.

Différentes structures de réseaux de neurones de point de vue nombre d'apports à l'entrée (selon les modèles de régression), nombre de couches cachées et nombre de neurones de chaque couche cachée sont testées. Par ailleurs, quatre (04) algorithmes d'apprentissage sont comparées. Les deux (02) premières sont dites classiques (BP et LM), alors que les dernières sont des métaheuristiques à base de population (RGA et HPSOGWO). La comparaison des résultats est basée sur un ensemble de critères comprenant l'indice de Willmot  $d$ , le critère  $Nash$ , le coefficient de corrélation  $R$ , de détermination  $R^2$ , l'erreur moyenne absolue  $MAE$  et la racine de l'erreur moyenne quadratique  $RMSE$ .

À travers les résultats trouvés, les principales conclusions qui peuvent être tirées se résument en :

- Les réseaux de neurones artificiels dotés de deux (02) couches cachées ont démontré une qualité de solution relativement meilleure par rapport à ceux d'une seule couche cachée dans certain cas. Néanmoins, dans la plupart des cas, l'amélioration n'est pas significative.
- Les meilleurs résultats sont trouvés par les réseaux de neurones entraînés par les algorithmes LM et RGA. Le LM a démontré des résultats meilleurs suivi par le RGA dans l'entraînement en considérant l'ensemble des réseaux testés et les critères d'évaluation utilisés.
- Les RNAs entraînés par le LM et le RGA ont pu détecter les pointes des débits mieux que les autres.
- La majorité des RNAs ont subi un sous-apprentissage. Ce phénomène s'aperçoit avec les critères d'évaluation faibles de la phase d'apprentissage.
- L'amélioration des résultats dans la phase d'apprentissage ne permet pas d'améliorer ceux de la phase de validation. Au contraire, dans la plupart des cas, elle entraîne leurs dégradations. Cette constatation s'explique par le phénomène de sur-apprentissage, où les réseaux de neurones artificiels perdent la caractéristique de généralisation.

La faible corrélation entre les débits observés et ceux calculés par les RNAs est peut-être due à leurs fortes variations mensuelles. Ce dernier phénomène est fréquemment observé dans les zones arides ou semi-arides telle que la zone étudiée. Encore, le tracé de la fonction d'autocorrélation simple (ACF) de la série chronologique mensuelle a relevé une corrélation très faible entre les éléments de celle-ci.

Les prévisions des apports du réservoir Hammam Boughrara par les réseaux de neurones artificiels présentant les meilleures performances nous a permis de constater une divergence des résultats. La surestimation des apports dans les phases d'apprentissage et de validation est propagée pour avoir impact sur les apports prévus. La même chose est valable le cas de sous-estimation des apports aussi.

Les apports prévus ont présenté un signal répétitif de mêmes amplitudes d'une année à l'autre sur la plupart de la période ciblée par la prévision. Ceci est peut-être induit par :

- Le facteur temps ( $t$ ) qui a été introduit comme entrée aux réseaux de neurones artificiels, et qui a peut-être forcé les sorties à prendre cette forme. Notons que ce facteur a été inclus dans le but d'intégrer la saisonnalité des apports et d'éviter la production des valeurs ne correspondant pas à son historique dans la région (p.ex., débits estivaux très forts et hivernaux très faibles) ;
- L'approche de prévision des apports utilisée, qui est basée sur la régression de la série chronologique mensuelle. Sachant qu'elle a été utilisée avec succès à différents pas du temps dans (Thirumalaiah and Deo 2000 (pas journalier) ; Kişi 2004 (pas mensuel) ; Kişi 2007 (pas journalier)) ;
- L'erreur due à la régression, c-a-d., la considération de l'apport prévu (calculé) d'une période dans la prévision de l'apport de la période qui la succède. Donc, l'erreur commise dans la prévision de l'apport d'un mois donné se répercute sur les mois qui le succèdent.

D'une manière générale, ces résultats posent la question sur l'efficacité de l'approche de régression utilisée dans la prévision des apports dans les régions semi-arides telles que le nord Algérien. De plus, ils nous encouragent à mieux l'explorer via les réseaux de neurones évolutionnaires, la tâche qui fait l'objet du chapitre suivant.

Finalement, en considérant l'ensemble de résultats d'apprentissage et de validation du réseau de neurones artificiels "ANN-LM-3 (6, 15)" déjà mentionnés. Les apports prévus par ce dernier seront utilisés comme un scénario d'une période humide dans la phase d'optimisation. Ceci permettra d'évaluer la performance des différents algorithmes d'optimisation présentés dans cette étude, ainsi la capacité du réservoir Hammam Boughrara à répondre aux différents objectifs de sa gestion.



# Chapitre 8

## Prévision des apports mensuels du barrage Hammam Boughrara par réseaux de neurones artificiels évolutionnaires

### 8.1 Introduction

Comme évoqué précédemment, l'objectif principal du recours aux réseaux de neurones artificiels est la prévision des apports mensuels du réservoir Hammam Boughrara, l'opération cruciale pour passer à l'optimisation de son exploitation dans un avenir incertain. Dans le chapitre précédent (chapitre 7), nous avons testé des RNAs paramétrés manuellement. Ce paramétrage ne garantit pas l'efficacité des réseaux de neurones testés.

Dans ce chapitre, l'approche de recherche des RNAs capables d'imiter le comportement d'un bassin versant en termes de débit d'écoulement à l'exutoire par des métaheuristiques, proposée dans le chapitre 4, est évaluée.

Les objectifs de ce chapitre peuvent être résumés en :

1. Évaluation de l'approche de recherche des réseaux de neurones artificiels évolutionnaires "EANNs" présentée dans le chapitre 4. Les algorithmes d'optimisation testés et comparés sont l'algorithme de recherche gravitationnelle "GSA" et l'algorithme hybride HPSOGWO. Ces derniers sont couplés séparément avec la rétropropagation du gradient d'erreur "BP" et Levenberg-Marquardt "LM", utilisés pour l'apprentissage des réseaux de neurones artificiels.
2. La recherche des réseaux de neurones artificiels meilleurs capables de prévoir les apports du réservoir. Dans ce cas, l'évaluation est basée sur des critères statistiques

différents (l'erreur moyenne absolue " $MAE$ ", la racine de l'erreur moyenne quadratique " $RMSE$ ", le coefficient de corrélation " $R$ ", le coefficient de détermination " $R^2$ ", le critère de Nash-Sutcliffe, et l'indice de Willmot " $d$ ").

3. Prévision des apports futurs du réservoir Hammam Boughrara par les meilleurs réseaux de neurones artificiels issus des RNAs proches-optimaux. Les apports générés seront utilisés pour accomplir la phase d'optimisation de la gestion du réservoir à travers des scénarios d'apports ultérieurement.

## 8.2 Recherche des réseaux de neurones évolutionnaires

### 8.2.1 Organisation des données d'entraînement

La série chronologique des débits d'oued Mouilah et d'oued Tafna du septembre 1972 au décembre 2016 (figure 7.1, chapitre 7) est réutilisée. De même, 70% de la série est réservé à l'apprentissage, tandis que le reste (30%) est conservé pour la phase de test et validation des modèles. L'organisation des données utilisées et la nomination des modèles évalués sont présentées dans le tableau 8.1. L'appellation "EANN, Evolutionary Artificial Neural Network" proposée par (Yao 1993) est optée dans cette étude pour distinguer les réseaux de neurones artificiels issus d'une opération d'optimisation via un algorithme évolutionnaire de ceux paramétrés par une tentation manuelle (procédure essai-erreur).

TABLE 8.1 – Modèles de régression et organisation de la série chronologique des apports en entrées et sorties

Modèle	EANNs	Entrées (inputs)	Sortie (output)
1	EANN-Algo1 <sup>a</sup> Algo2 <sup>b</sup> 1	$Q_{t-1}, t^c$	$Q_{c_t}$
2	EANN-Algo1-Algo2-2	$Q_{t-1}, Q_{t-2}, t$	$Q_{c_t}$
3	EANN-Algo1-Algo2-3	$Q_{t-1}, Q_{t-2}, Q_{t-3}, t$	$Q_{c_t}$
4	EANN-Algo1-Algo2-4	$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, t$	$Q_{c_t}$
5	EANN-Algo1-Algo2-5	$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}, t$	$Q_{c_t}$
6	EANN-Algo1-Algo2-6	$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}, Q_{t-6}, t$	$Q_{c_t}$

<sup>a</sup> Algorithme d'optimisation (GSA ou HPSOGWO).

<sup>b</sup> Algorithme d'apprentissage (BP ou LM).

<sup>c</sup> Indique l'indice de mois considéré par la prévision, varie entre 1 à 12 et standardisé dans l'intervalle [0,1].

### 8.2.2 Paramétrage des algorithmes d'optimisation

Comme indiqué auparavant, la procédure de recherche consiste à utiliser un algorithme d'optimisation métaheuristique (noté "Algo1" dans le tableau 8.1) pour paramétrer l'algorithme d'apprentissage "Algo2" en principe. Dans cette étude, l'algorithme GSA et HP-SOGWO sont utilisés pour accomplir la phase d'optimisation. Leurs paramétrages sont présentés dans le tableau 8.2. Concernant les algorithmes d'apprentissage, nous avons opté pour les méthodes classiques (la rétropropagation d'erreur "BP" et Levenberg-Marquardt "LM") seulement. Les métaheuristicques ne sont pas utilisés comme algorithmes d'apprentissage dans ce cas, car à travers plusieurs tests effectués, on a trouvé qu'ils exigent un temps d'exécution excessif dépassant les 46 heures pour chaque modèle sans amélioration de résultats.

TABLE 8.2 – Paramètres des algorithmes d'optimisation

<i>GSA</i>				
Taille de population	Nombre d'itérations	$G_0$	$\alpha_g$	
30	100	100	20	
<i>HPSOGWO</i>				
Taille de population	Nombre d'itérations	$C_1$	$C_2$	$C_3$
30	100	0.5	0.5	0.5

### 8.3 Résultats de recherche des EANNs

Le tableau 10 (Annexes) dévoile les résultats de recherche des EANNs proches-optimaux par l'algorithme GSA. Les algorithmes d'apprentissage employés, séparément, sont la rétropropagation du gradient de l'erreur "BP" et Levenberg-Marquardt "LM". L'algorithme GSA à été lancé pour faire 100 itérations. Dans chaque une, 30 réseaux de neurones ont été créés et manipulés pour aboutir à un réseau de neurones artificiels espéré d'être proche-optimal.

À partir de ces résultats, il est claire que la fonction d'activation la mieux adaptée est la sigmoïde, avec un coefficient  $\alpha_{activ}$  variant dans la plage [2, 3.4] le cas de l'algorithme d'apprentissage BP et dans l'intervalle [0.2, 1.85] le cas du LM. De même, le taux d'apprentissage  $\mu_{app}$  variant dans les intervalles [0.47, 0.79] et [0.52, 0.73] pour le BP et LM respectivement.

L'erreur d'apprentissage  $E_{app}$  et le nombre d'itérations  $K_{max-app}$  sont deux (02) paramètres permettant d'arrêter le processus d'apprentissage lorsque la condition sur l'un des deux est atteinte.

À partir du même tableau, il est clair que le paramétrage de l'erreur d'apprentissage par le GSA varie entre 0.38 à 0.74 pour la méthode BP et 0.03 à 0.43 pour la méthode LM.

Concernant le nombre d'itérations, il varie entre 13832 à 19300 le cas de l'algorithme d'apprentissage BP. Ce nombre est élevé pour la majorité des modèles EANNs, sachant que l'algorithme d'optimisation GSA fait le choix de la valeur de ce paramètre dans l'intervalle [10'000, 20'000]. Cependant, pour l'algorithme d'apprentissage LM ce nombre n'a pas dépassé 61 itérations, sachant que pour cet algorithme l'intervalle de choix est fixé à [40, 70].

Les structures des réseaux de neurones artificiels proches optimaux, en termes de nombre de couches cachées et nombre de neurones dans chaque couche, sont fixées par l'algorithme d'optimisation GSA. À partir de ces résultats (tableau 10, Annexes), il est trouvé que la structure en deux couches cachées ( $N_c = 2$ ) est la meilleure dans la totalité des modèles entraînés par la méthode BP. Le nombre de neurones dans la 1<sup>ère</sup> couche varie entre 5 à 9, tandis que dans la 2<sup>ème</sup> couche cachée, il varie de 4 à 12 selon le modèle de régression.

De même, pour les réseaux de neurones entraînés par l'algorithme LM, la structure composée de deux ( $N_c = 2$ ) couches cachées est l'optimale dans la majorité des modèles sauf pour le modèle "EANN-GSA-LM-3" et "EANN-GSA-LM-4". Le premier est d'une seule couche cachée ( $N_c = 1$ ), composée de cinq (05) neurones artificiels. Par contre, le deuxième est de trois ( $N_c = 3$ ) couches cachées dotées de 6, 5 et 2 neurones respectivement. D'une manière générale, le nombre de neurones cachés varie entre 2 à 12 selon le modèle de régression.

Le tableau 11 (Annexes) annonce les résultats de recherche des réseaux de neurones artificiels proches optimaux par l'algorithme HPSOGWO. Identiquement au cas précédent, les méthodes d'apprentissage utilisées sont la rétropropagation de l'erreur et Levenberg-Marquardt. L'algorithme HPSOGWO a exécuté 100 itérations lors le processus d'optimisation, dans lesquelles une population composée de 30 réseaux de neurones a été modifiée et améliorée pour aboutir à un réseau de neurones artificiels considéré proche optimal.

À partir de ces résultats, la fonction d'activation sigmoïde est trouvée meilleure pour la totalité des modèles. Encore, pour les EANNs entraînés par le BP, la valeur de coefficient  $\alpha_{activ}$  varie entre 1.26 à 2.09 environ. Néanmoins, pour ceux entraînés par le LM, il varie de 0.46 à 2.42 environ.

Le nombre d'itérations effectuées par la méthode d'apprentissage BP pour chaque modèle varie de 18'353 à 19'830. Il est important vu l'intervalle de paramétrage considéré au départ ([10'000, 20'000]). De même pour les réseaux de neurones entraînés par l'algorithme LM, le nombre d'itérations varie de 60 à 69. Il est faible par rapport à la méthode de rétropropagation, mais reste relativement important par rapport aux EANNs optimisés par l'algorithme GSA et entraînés par le même algorithme "LM".

Concernant la structure des réseaux de neurones, il est trouvé que celle formée par deux (02) couches cachées est la meilleure pour la totalité des modèles. Le nombre de neurones de la 1<sup>ère</sup> couche cachée varie de 4 à 7 le cas des EANNs entraînés par le BP, et de 4 à 9 pour ceux entraînés par le LM. De même pour la deuxième (2<sup>ème</sup>) couche cachée, mais cette fois-ci, le nombre de neurones varie de 6 à 13 pour la totalité des modèles évalués. La remarque qu'on peut soulever ici, est que le nombre de neurones artificiels de la 2<sup>ème</sup> couche est plus important par rapport à la 1<sup>ère</sup> dans presque la totalité des réseaux de neurones trouvés.

De même, l'erreur moyenne absolue  $MAE$  et la racine de l'erreur moyenne quadratique  $RMSE$  sont réduites par rapport à celles des réseaux de neurones artificiels paramétrés manuellement (tableaux 8 et 9). L'erreur moyenne absolue a atteint  $2.572 Mm^3/mois$  le cas de réseau de neurones "EANN-GSA-PB-4(9,10)" et  $1.688 Mm^3/mois$  le cas de "EANN-GSA-LM-6(7,10)". D'autre part, le coefficient  $Nash$  est amélioré aussi. Il varie entre 0.39 à 0.6 pour les réseaux de neurones entraînés par la méthode BP et 0.63 à 0.88 pour ceux entraînés par la méthode LM. Pour cette dernière méthode, les valeurs du critère  $Nash$  sont les plus significatives.

Concernant les valeurs de coefficient de corrélation et de détermination  $R$  et  $R^2$ , ils sont améliorés aussi en considérant la phase d'apprentissage uniquement. Le coefficient de corrélation  $R$  varie entre 0.48 à 0.81 environ le cas des réseaux entraînés par la méthode BP et de 0.8 à 0.94 environ le cas de ceux entraînés par la méthode LM.

Les critères  $Nash$ ,  $R$  et  $R^2$  sont les plus sensibles aux déphasages entre les pics (extrêmes) des débits observés et calculés. Les valeurs relativement bonnes déjà trouvées lors de la phase d'apprentissage indiquent une correspondance meilleure entre les débits observés et ceux calculés dans cette phase.

Tenant compte la phase de validation, il est clair que l'ensemble des résultats des différents critères sont dégradés, notamment l'indice de Willmot  $d$ , le critère de  $Nash$ , le coefficient de corrélation  $R$ , de détermination  $R^2$ . L'indice de Willmot  $d$  n'a pas dépassé la valeur 0.5 pour la totalité des réseaux de neurones. Le critère  $Nash$  est négatif. Il varie de -1.07 à -0.434 le cas des réseaux de neurones entraînés par la méthode BP, et de -10.4 à -0.143 pour le cas des réseaux entraînés par la méthode LM. Dans ce cas, les moyennes mensuelles sont peut-être préférées par rapport aux sorties de ces EANNs.

De même pour les coefficients  $R$  et  $R^2$ . Les résultats sont médiocres pour la majorité des réseaux de neurones sauf le "EANN-GSA-BP-6(7,11)" (composé de 02 couches cachées. La 1<sup>ère</sup> est de 7 neurones. La 2<sup>ème</sup> est de 11 neurones, et exige à l'entrée les débits :  $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$ ,  $Q_{t-4}$ ,  $Q_{t-5}$  et  $Q_{t-6}$ , ainsi l'indice de mois considéré pour la prévision  $t$  standardisé dans l'intervalle  $[0, 1]$ ), où le coefficient de corrélation  $R$  a atteint 22.8%, qui reste

très modéré. Par ailleurs, les valeurs atteintes des critères ( $R$  et  $R^2$ ) sont faibles ( $R \leq 28.5\%$ ) le cas des RNAs entraînés par la méthode de Levenberg-Marquardt.

Non seulement les critères  $d$ ,  $Nash$ ,  $R$  et  $R^2$ , l'erreur moyenne absolue  $MAE$  et la racine de l'erreur moyenne quadratique  $RMSE$  sont dégradées aussi. Les valeurs de cette dernière ( $RMSE$ ) varient de  $6.4 \text{ Mm}^3/\text{mois}$  à  $7.58 \text{ Mm}^3/\text{mois}$  le cas des réseaux de neurones artificiels entraînés par la méthode d'apprentissage BP, et de  $5.62 \text{ Mm}^3/\text{mois}$  à  $17.82 \text{ Mm}^3/\text{mois}$  le cas de ceux entraînés par la méthode LM.

Tout bien considéré, les observations qui peuvent être relevées à travers ces résultats, sont :

- La méthode d'entraînement des réseaux de neurones Levenberg-Marquardt est plus efficace que la rétropropagation du gradient d'erreur ;
- L'amélioration des résultats de la phase d'apprentissage a induit la dégradation de ceux de la phase de validation. Cela, indique un sur-apprentissage des réseaux de neurones artificiels dans ce cas.

Sur le même volet, le tableau 13 (Annexes) avise l'évaluation des résultats d'apprentissage et de validation des réseaux de neurones artificiels produits par l'algorithme HPSOGWO (tant qu'algorithme d'optimisation) et les algorithmes BP et LM (algorithmes d'apprentissage).

Tout d'abord, les réseaux de neurones artificiels issus de cette opération d'optimisation ont démontrés des résultats très significatifs lors la phase d'apprentissage. L'indice de Willmot  $d$  varie de 0.69 à 0.94 environ le cas des EANNs entraînés par l'algorithme BP, et de 0.94 à 0.97 le cas de ceux entraînés par le LM. Encore, le critère  $Nash$  varie dans l'intervalle  $[0.38, 0.77]$  et  $[0.79, 0.91]$  pour les réseaux de neurones entraînés par les algorithmes BP et LM respectivement. En plus, les valeurs de coefficients de corrélation  $R$  (et par conséquent celles du coefficient de détermination  $R^2$ ) sont très significatives dans la phase d'entraînement. Dans ce sens,  $R$  varie de 0.62 à 0.89 environ le cas des EANNs entraînés par le BP, et de 0.89 à 0.951 le cas de ceux entraînés par le LM.

Concernant les valeurs de l'erreur moyenne absolue  $MAE$  et de la racine de l'erreur moyenne quadratique  $RMSE$ , elles sont améliorées aussi. Les valeurs  $MAE$  sont de  $2.46 \text{ Mm}^3/\text{mois}$  à  $3.18 \text{ Mm}^3/\text{mois}$  concernant les EANNs entraînés par l'algorithme BP, et de  $1.61$  à  $2.33 \text{ Mm}^3/\text{mois}$  le cas de ceux entraînés par le LM. De l'autre côté, les valeurs  $RMSE$  sont de  $4.25 \text{ Mm}^3/\text{mois}$  à  $7.09 \text{ Mm}^3/\text{mois}$  le cas des réseaux entraînés par le BP, et de  $2.74 \text{ Mm}^3/\text{mois}$  à  $4.085 \text{ Mm}^3/\text{mois}$  les cas de ceux entraînés par le LM. Ces résultats sont meilleurs par rapport à ceux des RNAs paramétrés manuellement (chapitre 7).

Sur le côté validation, les résultats des différents critères sont dégradés pour la totalité

des modèles, quelle que soit la méthode d'apprentissage utilisée. En premier lieu, les valeurs de l'indice de Willmot n'ont pas dépassé 0.5, et pour la majorité des cas, elles sont inférieures à 0.3. En deuxième lieu, les valeurs du critère *Nash* varient dans la plage -10.40 à -0.24. Elles sont plus basses le cas des EANNs entraînés par la méthode LM. En troisième lieu, les valeurs de coefficient de corrélation *R* sont très basses (ne dépassant pas 28.5%), notamment pour les EANNs entraînés par la méthode BP. En dernier lieu, les valeurs *MAE* et *RMSE* sont dégradées par rapport à la phase d'entraînement et pour la majorité des modèles. La dégradation des résultats est plus pertinente le cas des EANNs entraînés par la méthode LM.

### 8.3.1 Comparaison graphique des résultats de la phase d'apprentissage

Les figures 7 et 8 (Annexes) montrent la comparaison entre les apports observés et ceux calculés en phase d'apprentissage par les différents réseaux de neurones proches optimaux.

La comparaison entre les apports observés et ceux calculés par le réseau de neurones artificiels "EANN-GSA-BP-6 (7, 11)" a montré une sous estimation des débits d'apports mensuels sur la majorité de la période d'entraînement (370 mois). La correspondance entre les apports est modeste pour les débits faibles à moyens ( $\leq 5Mm^3/mois$ ) et faible pour le reste, à l'exception des apports très forts ( $> 60Mm^3/mois$ ), où une très bonne correspondance est observée pour ces derniers.

Concernant le réseau de neurones artificiels "EANN-GSA-LM-2 (7, 8)", des fluctuations des apports calculés autour des apports observés sont remarquées. Ces fluctuations couvrent généralement, les apports les plus fréquents ( $\leq 9Mm^3/mois$ ), ce qui explique les bonnes valeurs des critères d'évaluation notamment *d*, *MAE*, *R* et *R*<sup>2</sup>. Par contre, ce réseau n'a pas pu détecter la majorité des apports variant de  $9Mm^3/mois$  à  $45Mm^3/mois$ , néanmoins, les pics des apports ( $> 45Mm^3/mois$ ) sont moyennement détectés. D'une manière générale, la phase d'apprentissage de ce réseau de neurones (EANN-GSA-LM-2 (7,8)) est mieux qu'au précédent (EANN-GSA-BP-6 (7, 11)).

L'apprentissage du réseau de neurones artificiels "EANN-HPG-BP-6 (6, 9)" a prouvé des valeurs meilleures des critères d'évaluation dans la phase d'apprentissage (tableau 13, Annexes). La comparaison graphique entre les apports calculés dans cette phase et ceux observés montre des fluctuations pareilles au réseau de neurones précédent (EANN-GSA-LM-2 (7, 8)) dans le cas des débits ( $\leq 9Mm^3/mois$ ). Cependant, les amplitudes des fluctuations sont réduites par rapport à ce dernier (EANN-GSA-LM-2 (7, 8)). Les apports variant entre  $9Mm^3/mois$  et  $45Mm^3/mois$  sont ratés dans la plus part des cas, toutefois, la correspondance dans cette fourchette reste meilleure qu'avec les réseaux de neurones optimisés par l'algorithme GSA et entraînés par les deux algorithmes BP et LM. Concernant les apports extrêmes ( $> 45Mm^3/mois$ ), la

correspondance est très significative. Ceci, explique la qualité des valeurs des critères d'évaluation atteintes, notamment  $Nash$  et  $R^2$  sensibles aux pics.

En dernier, le réseau de neurones artificiels "EANN-HPSOGWO-LM-4 (8, 13)" a témoigné les meilleurs résultats dans la phase d'apprentissage (tableau 13, Annexes). La comparaison graphique montre une similitude très significative entre les apports mensuels observés et calculés  $\leq 20Mm^3/mois$ , et relativement parfaite entre les apports mensuels extrêmes ( $> 40Mm^3/mois$ ). En revanche, la correspondance entre les apports de la fourchette  $20Mm^3/mois$  à  $40Mm^3/mois$  n'est pas forte, mais reste signifiante.

### **8.3.2 Comparaison graphique des résultats de la phase de validation**

La figure 9 (Annexes) montre la comparaison entre les apports observés et ceux calculés en phase de validation par les meilleurs modèles de réseaux de neurones trouvés auparavant.

Commençant par les réseaux de neurones recherchés par l'algorithme GSA. On remarque clairement que les apports calculés par le réseau de neurones artificiels "EANN-GSA-BP-6 (7, 11)" sont sous-estimés sur la plupart de la période du test. Toutefois, les apports calculés par le réseau de neurones "EANN-GSA-LM-2 (7,8)" ont présentés des fluctuations autour des apports observés, prennent des ampleurs de 1 à 4  $Mm^3$  environ pour les apports faibles et moyens. Ce résultat n'est pas applicable sur les apports forts, dont ces derniers sont ratés dans la plupart des cas.

Concernant les réseaux de neurones artificiels recherchés par l'algorithme HPSOGWO, le réseau de neurones "EANN-HPSOGWO-BP-6 (6, 9)" entraîné par l'algorithme "BP", a prouvé des apports calculés légèrement inférieurs par rapport aux apports observés. Mais, de point de vue volume d'apport sur la totalité de la période du test, le volume total estimé par ce modèle est proche au volume total observé. En revanche, le réseau de neurones artificiels "EANN-HPSOGWO-LM-4 (8, 13)" a généré des apports proches mais surestimés par rapport aux débits observés. Cependant, les apports calculés par ce modèle ont présenté des pics plus importants dépassant les pics enregistrés.

### **8.3.3 Performance des algorithmes d'optimisation**

La figure 8.1 annonce l'évolution de la valeur de la fonction objectif adoptée ( $Fobj_2$ , équation 4.1) en fonction de nombre d'itérations. Les résultats correspondent aux meilleurs agents de recherche relatifs aux algorithmes d'optimisation GSA et HPSOGWO, où ces derniers sont couplés séparément avec ceux d'apprentissage (BP et LM).

La figure annonce clairement qu'à partir de la cinquième (5ème) itération ( $k = 5$ ), le couplage des algorithmes "HPSOGWO-LM" a prouvé la meilleure performance. Cette dernière se répercute dans la faible valeur de la fonction objectif, et par conséquent la faible erreur d'apprentissage. Ce modèle a été secondé par le couplage d'algorithmes "GSA-LM" jusqu'à l'itération soixante ( $k = 60$ ). Puis, le couplage d'algorithmes "HPSOGWO-BP" a surpassé le précédent jusqu'à la fin du processus d'optimisation.

La figure 10 (Annexes) montre l'évolution de la valeur de la fonction objectif ( $F_{obj_2}$ , équation 4.1) de l'ensemble des modèles testés. À partir de ces résultats, il est clair que le couplage des métaheuristiques avec l'algorithmes d'apprentissage LM a donné des erreurs plus faibles en comparaison avec l'algorithme BP dans la majorité des cas.

Le tableau 8.3 décrit le temps moyen de recherche des réseaux de neurones proches optimaux par les deux (02) métaheuristiques. Les résultats montrent évidemment que le couplage d'un métaheuristique avec l'algorithme d'apprentissage LM est plus rapide (10 fois plus) et efficace qu'avec l'algorithme BP. Cela due simplement au nombre faible d'itérations exigées par le LM pour aboutir à un résultat satisfaisant. De l'autre côté, l'algorithme GSA a pris au moyen trois heures ( $03^h$ ) plus que le HPSOGWO lorsque a été couplé avec l'algorithme d'apprentissage BP. En revanche, il a été plus rapide (avec d'environ 23 minutes) quand a été couplé avec l'algorithme LM.

Comme constatations finales, les réseaux de neurones entraînés par l'algorithme LM ont présenté les plus faibles erreurs d'apprentissage dès les premières itérations. En terme d'optimisation, l'algorithme hybride HPSOGWO a indiqué une meilleure performance en comparaison avec le GSA dans la problématique en question.

TABLE 8.3 – Le temps moyen de recherche des EANNs proches-optimaux

Modèle	Algo. d'optimisation	Algo. d'apprentissage	Temps moyen
EANN-GSA-BP	GSA	Rétropropagation "BP"	$13^h 7^m 12^s$
EANN-GSA-LM	GSA	Levenberg-Marquardt "LM"	$31^m 13.8^s$
EANN-HPG-BP	HPSOGWO	Rétropropagation "BP"	$9^h 57^m 36^s$
EANN-HPG-LM	HPSOGWO	Levenberg-Marquardt "LM"	$54^m 45.6^s$

## 8.4 Résultats de prévision des apports

Cette section est consacrée aux résultats de prévision des apports mensuels du barrage Hammam Boughrara par les réseaux de neurones artificiels évolutionnaires aboutissant aux meilleures valeurs en termes d'indices d'évaluation. La prévision est réalisée sur une durée

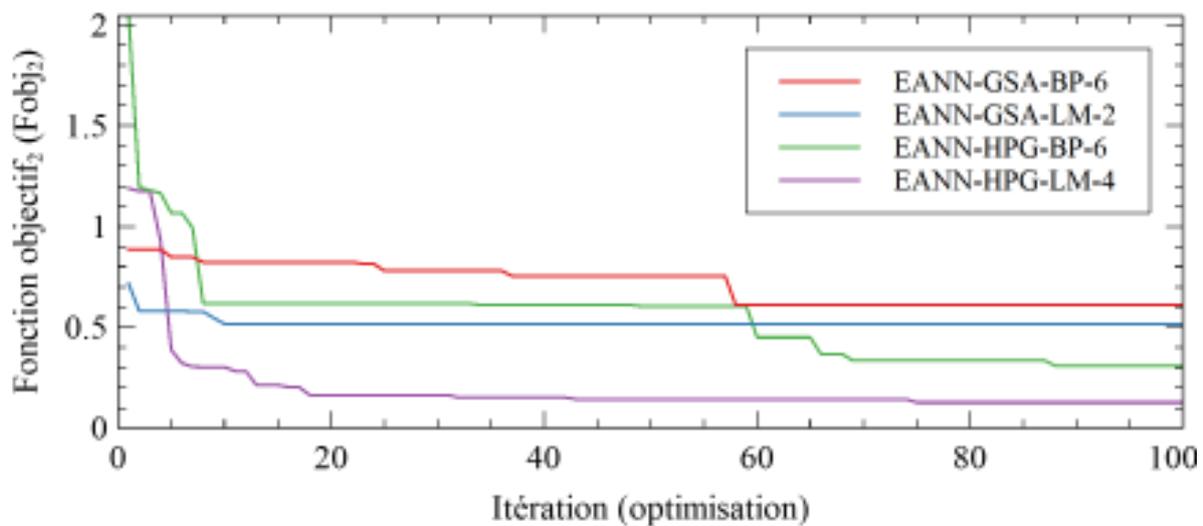


FIGURE 8.1 – Évolution de la valeur de la fonction objectif (équation (4.1)) en fonction de nombre d'itérations (les meilleurs modèles)

de 120 mois (janvier 2017 au décembre 2026). Le processus de prévision est par régression. Il commence par l'injection des apports observés dans le modèle (réseau de neurones artificiels) selon le nombre d'entrées, puis, le débit à prévoir du mois cible sera calculé. Par la suite, ce dernier sera réinjecté dans le modèle pour générer l'apport de mois qui le succède. Ce processus est répété séquentiellement jusqu'à avoir les apports de la période de prévision. La figure 8.2 présente le tracé graphique des apports prévus, comparés aux apports moyens mensuels interannuels comptés du septembre 1972 au décembre 2016.

D'abord, le réseau de neurones artificiels "EANN-GSA-BP-6 (7, 11)" a généré un signal saisonnier répétitif (cyclique) des apports, dont les valeurs sont nulles du mois de janvier au mois d'aout, et très faibles ( $< 4Mm^3/mois$ ) du mois de septembre au décembre. Ces derniers sont inférieurs aux moyennes mensuelles interannuelles. En outre, le réseau de neurones artificiels "EANN-GSA-LM-2 (7, 8)" a produit des apports mensuels cycliques à partir du 16<sup>ième</sup> mois (avril 2018), mais sont proches aux moyennes mensuelles et diffèrent aux apports produits par le premier réseau de neurones. Ces apports sont caractérisés par deux (02) pics, hivernal ( $\cong 10.2Mm^3$  au mois de janvier) et printanier ( $\cong 10.6Mm^3$  au mois d'avril), et des apports estivaux supérieurs à la moyenne mensuelle de cette période.

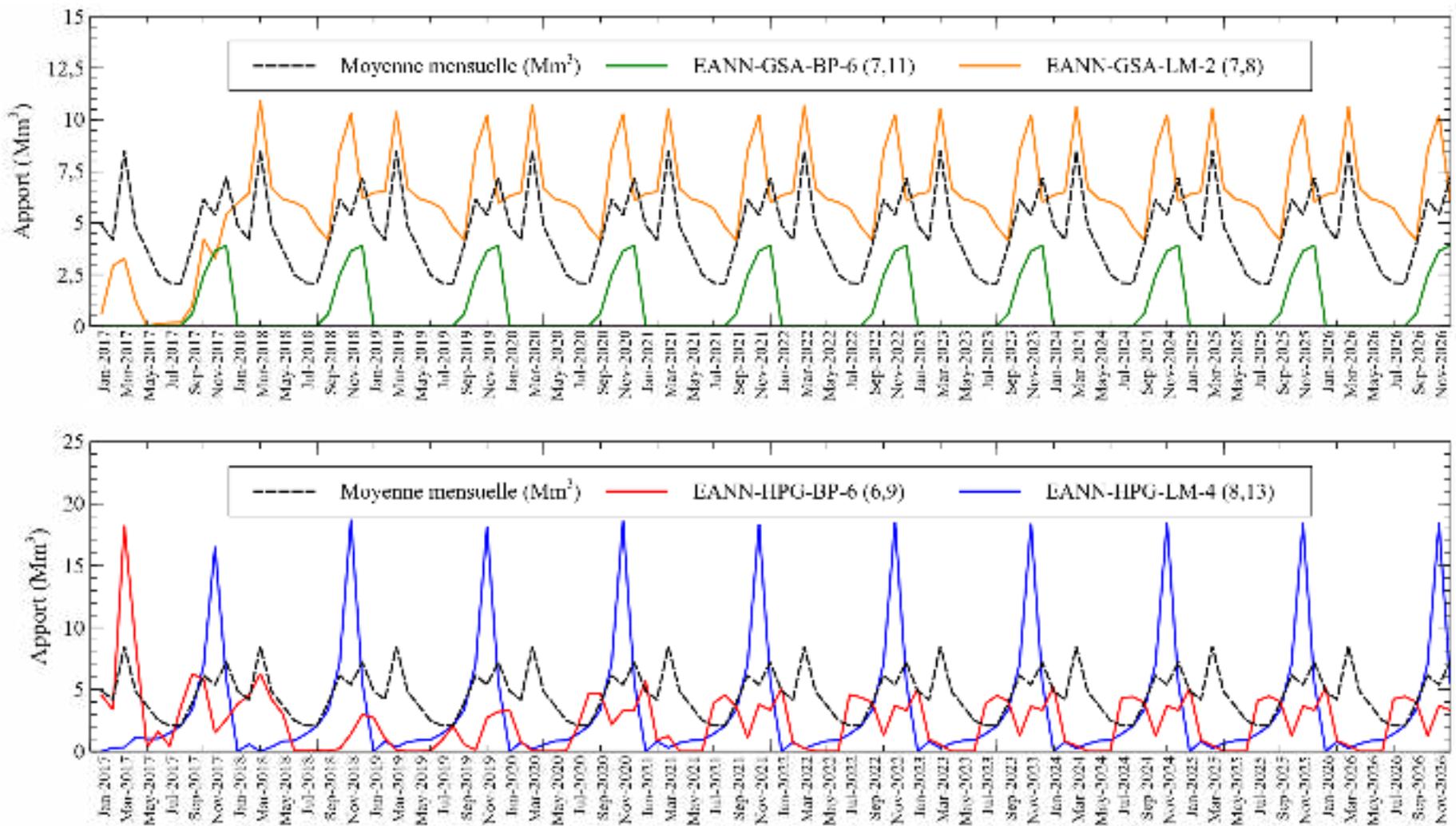


FIGURE 8.2 – Résultats de prévision des apports du réservoir Hammam Boughrara par les réseaux de neurones évolutionnaires

De même, le réseau de neurones artificiels "EANN-HPSOGWO-BP-6 (6, 9)" a dégagé un signal différent par rapport aux autres. Hormis, un pic de  $18.23Mm^3$  est généré pour le mois de mars 2017 (3<sup>ème</sup> mois de prévision), suivi par des apports mensuels faibles par rapport aux moyennes mensuelles jusqu'au mois 43 (juillet 2020) de la période de prévision. Par la suite, un signal répétitif est produit jusqu'à la fin de la période de prévision (120 mois). Il est caractérisé par des pics figurant au mois d'aout et de janvier par environ  $4.4Mm^3/mois$  et  $5.10Mm^3/mois$  respectivement.

En dernier lieu, le réseau de neurones artificiels "EANN-HPSOGWO-LM-4 (8, 13)" a rapporté des apports différents par rapports à ceux fournis par les EANNs présentés auparavant. Cependant, les apports résultants présentent une forte saisonnalité. Ils augmentent rapidement à partir de mois septembre ( $\cong 3.4Mm^3/mois$ ) pour atteindre environ  $\cong 18Mm^3$  au mois de novembre. Puis, ils diminuent rapidement pour rapprocher des apports presque nuls le reste de l'année. Ce régime d'apports se répète le long de la période de prévision, avec un changement léger d'une année à l'autre.

La divergence entre les résultats de prévision des apports du réservoir Hammam Boughrara aboutis par les différents réseaux de neurones artificiels présentés dans ce chapitre ainsi le chapitre précédent (7) constitue une source d'incertitude. À défaut d'une connaissance préalable sûre dans la matière, les apports mensuels prévus peuvent constituer des scénarios d'apports à la base desquels des politiques de gestion du réservoir peuvent être élaborées. Cela, permet de mesurer la performance de fonctionnement du réservoir selon le scénario considéré et de prendre les mesures nécessaires le cas échéant.

## **8.5 Conclusion**

Ce chapitre est dédié principalement à la prévision des apports mensuels du réservoir Hammam Boughrara par les réseaux de neurones artificiels évolutionnaires (EANNs) et l'évaluation de l'approche de recherche de ces derniers par des métaheuristiques.

En premier lieu, les deux (02) métaheuristiques (l'algorithme de recherche gravitationnelle "GSA" et l'algorithme hybride HPSOGWO) sont utilisés pour la recherche des EANNs. Les résultats ont montré que les réseaux de neurones artificiels dotés de deux (02) couches cachés (c-à-d., une couche d'entrée, deux cachées et une de sortie) et dont la fonction d'activation est la sigmoïde, sont les mieux adaptés dans ce cas. Ces EANNs sont constitués de deux (02), quatre (04) ou six (06) neurones à l'entrée et par conséquent exigent le même nombre des apports mensuels. En effet, les modèles de régression 2, 4 et 6 ont montré des résultats meilleurs par rapport aux autres.

En deuxième lieu, par rapport à la méthode de paramétrage manuelle des RNAs (chapitre 7), les EANNs ont prouvé une amélioration significative des indices d'évaluation ( $d$ ,  $Nash$ ,  $R$ ,  $R^2$ ,  $MAE$  et  $RMSE$ ) de la phase d'entraînement au détriment de ceux de la phase de validation "ou de test". À titre d'exemple, le coefficient de corrélation  $R$  de la phase d'apprentissage a dépassé 62% pour tous les modèles et allant jusqu'à 95.14%. Hormis, ce coefficient n'a pas dépassé 37.7% dans la phase de validation pour tous les modèles évalués. Cela signifie un sur-apprentissage des réseaux de neurones artificiels dans la plupart des cas.

Concernant les algorithmes d'optimisation, la comparaison des deux (02) métaheuristiques a montré des résultats relativement proches, avec une légère supériorité au profit de l'algorithme hybride HPSOGWO par rapport au GSA, notamment en termes de convergence rapide. Cela, permet d'avoir des réseaux de neurones de qualité relativement bonne avec un processus d'optimisation assez court.

Parlant des méthodes d'apprentissage des réseaux de neurones artificiels. La méthode de Levenberg-Marquardt "LM" a démontré une performance meilleure avec un nombre faible d'itérations par rapport à la méthode de rétropropagation du gradient d'erreur "BP". En effet, le couplage de la méthode LM avec les métaheuristiques consomme un temps d'exécution faible et permet en conséquence de lancer plusieurs tentatives de recherche des EANNs dans un temps raisonnable. Néanmoins, à travers les différents tests effectués, on a remarqué l'instabilité occasionnelle de cette méthode, c-à-d., son exécution multiples avec les mêmes entrées et les mêmes paramètres ne garantit pas les mêmes résultats.

Les réseaux de neurones artificiels EANN-HPSOGWO-BP-6(6, 9) et EANN-HPSOGWO-LM-4(8, 13) ont montré une performance pertinente par rapport aux autres. Le 1<sup>er</sup> est composé d'une couche d'entrée de six (06) neurones, deux couches cachées de six (06) et neuf (09) neurones respectivement, et d'une couche de sortie. Le 2<sup>ème</sup> réseau est constitué de quatre (04) couches, une d'entrées, deux (02) cachées de huit (8) et treize (13) neurones, et une couche de sortie. L'algorithme d'optimisation des paramètres est le HPSOGWO, tandis que les algorithmes d'apprentissage sont le BP et LM respectivement. Vu les résultats atteints d'apprentissage, d'entraînement et de prévision des apports mensuels du réservoir Hammam Boughrara par ces deux réseaux de neurones artificiels, ses apports prévus seront utilisés comme des scénarios d'apports dans la phase d'optimisation.

Enfin, l'incertitude concernant la prévision des apports est sans doute importante notamment dans régions arides ou semi-aride telle que l'Algérie. Cette problématique d'incertitude n'est pas toutefois le sujet de cette thèse. Cependant, la perfection de l'optimisation de l'exploitation des réservoirs d'eau nécessite la considération de l'amélioration des modèles de prévisions par la communauté scientifique dans la discipline en question.



# Chapitre 9

## Optimisation de la gestion du réservoir Hammam Boughrara : Résultats et discussions

### 9.1 Introduction

L'optimisation de la gestion d'un réservoir dans l'avenir ne peut être qu'incertaine. Les sources d'incertitudes sont multiples, notamment celles de la prévision des apports et des besoins futurs des usagers. Sous l'effet de l'absence d'une connaissance sûre et tranchée des entrants d'un modèle d'optimisation (apports, besoins, taux d'évaporations, revenus économiques de lâchers, ... etc), la modélisation simplificatrice de ces derniers constitue le seul moyen de perfectionnement du modèle d'optimisation. La modélisation génère des sorties selon l'approche de calcul utilisée, y compris la méthode de calcul choisie, le degré de simplification adopté, les facteurs pris en compte, ... etc. Cette opération conduit dans la plupart des cas à la génération des scénarios sur les apports, la consommation de la ressource en eau, ... etc.

Rappelons ici que les règles de gestion de la ressource en eau délivrée par barrages réservoirs subdivisent en deux catégories, i) règle de répartition de la ressource entre usagers et ii) règle de sa distribution temporelle. Dans ce chapitre, nous allons concentrer sur cette dernière (la règle de distribution temporelle) face aux scénarios d'apports dérivés de l'étape de prévision discutée dans les chapitres précédents.

Dans ce contexte, les objectifs du chapitre se résument principalement dans les deux points suivants :

1. Évaluation de la performance des différents algorithmes d'optimisation métaheu-

ristiques exposés auparavant (RGA, GSA, HPSOGWO et IGWO) dans l'optimisation de la gestion (règles de répartition temporelles) de l'eau du réservoir Hammam Boughrara. Dans ce cadre, les quatre métaheuristiques sont paramétrés et lancés plusieurs fois. La qualité de solutions, la convergence et la stabilité des métaheuristiques sont jugées.

2. Évaluation de l'efficacité des règles de répartition temporelle de la ressource en eau optimisées par les quatre algorithmes à accomplir les objectifs tracés du réservoir Hammam Boughrara face aux trois scénarios d'apports mensuels différents, en respect des différentes contraintes relatives à son exploitation. L'estimation de la performance est référée à plusieurs critères tels que les volumes déversés, les fiabilités volumétrique et temporelle, la résilience, la vulnérabilité du système, ... , etc.

## 9.2 Fonction objectif et entrées du modèle d'optimisation

L'optimisation de la gestion du réservoir Hammam Boughrara discutée dans ce qui suit, vise à maximiser la satisfaction de la demande en eau ( $D_t$ ) en aval du réservoir en respect des différentes contraintes liées au système de gestion (bassin versant - réservoir - usagers). En effet, la décision prise par les algorithmes d'optimisation se prononce sur les variables de décision (ou d'intérêt) représentées par les lâchers mensuels ( $R_t$ ,  $t = 1, 2, \dots, 120$ ). La fonction objectif sur la base de laquelle l'optimisation est effectuée est décrite par l'équation 1.17 (chapitre 1), rétablie ci-après. Le système de tarification et de facturation appliqué actuellement sur la ressource en eau délivrée par les barrages peut donner un sens de revenus monétaires à cette fonction au profit du gestionnaire. L'optimisation couvre une durée de 120 mois allant du 01 janvier 2017 au 31 décembre 2026. En effet, 120 variables sont à déterminer par chaque algorithme d'optimisation et pour chaque scénario d'apports considéré.

$$F_{obj} = \text{Minimiser} \sum_{t=1}^{T=120} \left[ \left( \frac{R_t - D_t}{D_{max}} \right)^2 + P_{1,t} + P_{2,t} + P_{3,t} \right]$$

Tels que :

- $R_t$  : lâcher du réservoir "sortie" (en  $Mm^3$ ) au mois  $t$ . Il représente la variable de décision ;
- $D_t$  : la demande en eau en aval du réservoir au mois  $t$  (en  $Mm^3$ ) ;
- $D_{max}$  : la demande en eau maximale en aval au cours de la période d'exploitation totale  $T$  considérée ;
- $P_{1,t}, P_{2,t}, P_{3,t}$  : pénalités appliquées dans les algorithmes durant le processus d'op-

timisation afin de considérer les contraintes sur le stockage minimal et maximal, et sur le lâcher maximal respectivement ;

- $T$  : la durée totale d'optimisation considérée ( $T = 120\text{mois}$ ).

La figure 9.1 annonce les trois scénarios d'apports du réservoir estimés par deux réseaux de neurones artificiels évolutionnaires (scénarios 1 et 2) et par un réseau de neurones artificiels conventionnel (scénario 3). Les trois scénarios couvrent la période du janvier 2017 au décembre 2026, soit 120 mois. La figure indique aussi la courbe modélisant l'évolution de la demande mensuelle en eau (domestique + agricole + industrielle) à assurer par la retenue durant la même période.

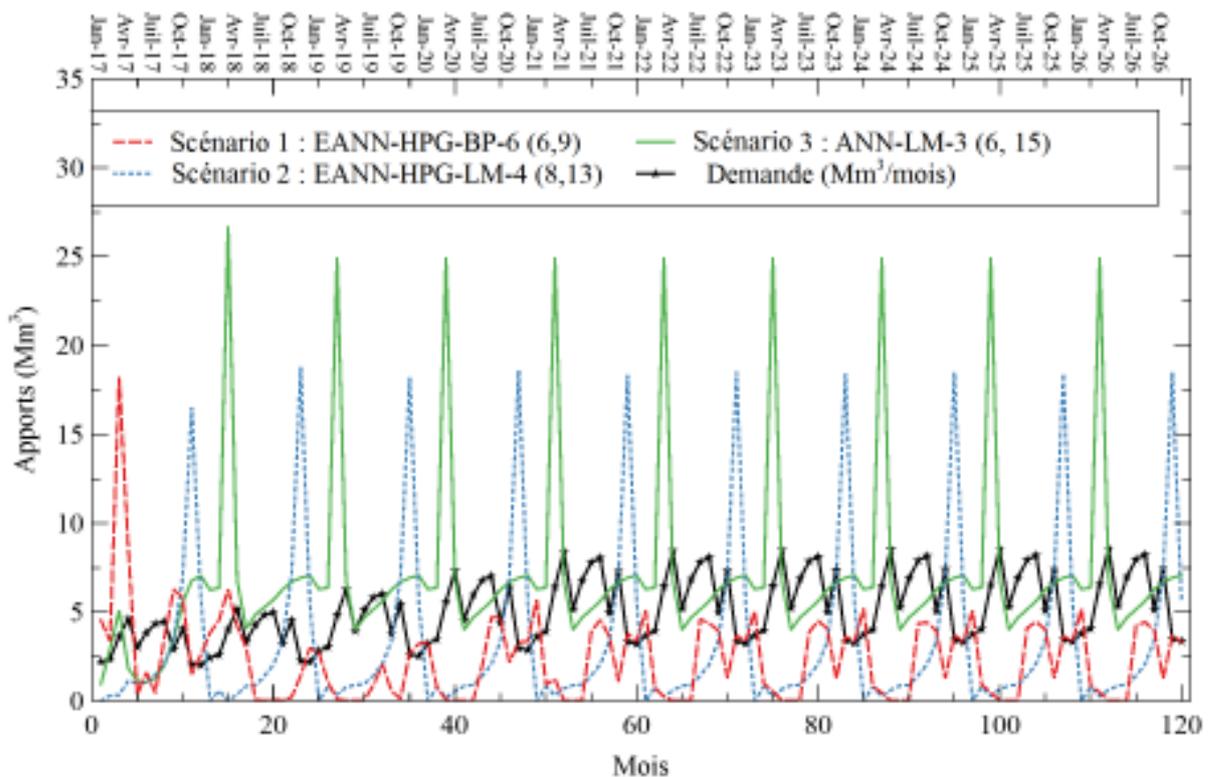


FIGURE 9.1 – Scénarios d'apports 1 et 2 prévus par les réseaux de neurones artificiels évolutionnaires (EANNs) et scénario 3 prévu par un réseau de neurones artificiels conventionnel (ANN), ainsi la courbe modélisant la demande en eau mensuelle en aval du réservoir Hammam Boughrara

Les volumes mensuels de pertes d'eau par évaporation et infiltration sont considérés dans le modèle d'optimisation. À un mois donné, la perte par évaporation est calculée en fonction de la surface moyenne du plan d'eau de la retenue et du taux d'évaporation moyen de la région du mois en question (équation 1.9). Les taux moyens d'évaporation pris en compte dans cette étude sont indiqués dans la figure 9.2 ci-après. Cependant, les volumes de pertes d'eau par infiltration sont pris par une manière plus simplifiée, dont une valeur constante égale à  $0.033\text{Mm}^3/\text{mois}$  est adoptée.

Les caractéristiques principales du réservoir sont prises en compte dans cette phase d'optimisation, notamment la capacité maximale ( $S_{max} \cong 177Mm^3$ ) et minimale (volume mort =  $S_{min} \cong 23.0Mm^3$ ), ainsi le volume d'eau stocké au début de la période objet d'optimisation  $S_0 = 153Mm^3$ . Ce dernier volume correspond au stockage du réservoir enregistré à la fin du mois décembre 2016 (ANBT 2017).

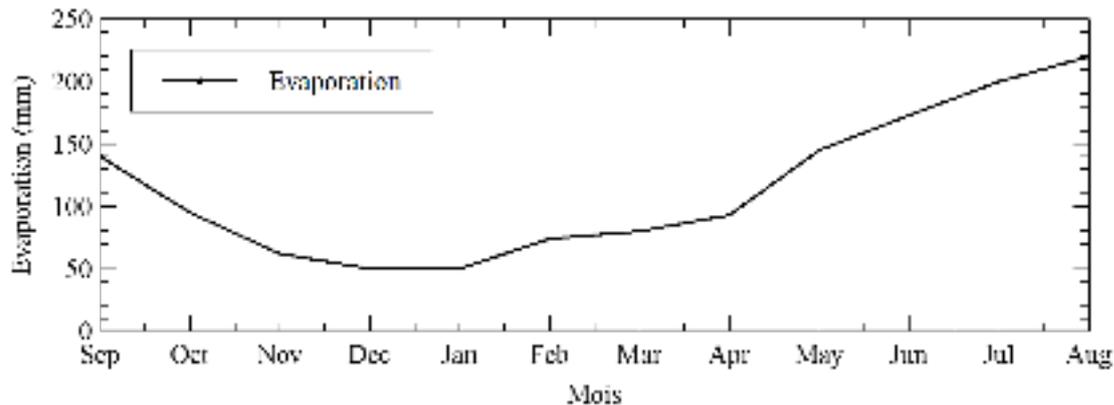


FIGURE 9.2 – Les taux d'évaporation mensuels utilisés dans le processus d'optimisation

### 9.3 Algorithmes d'optimisation et paramétrage

Comme évoqué précédemment, les algorithmes RGA, GSA, HPSOGWO et IGWO (chapitre 2) sont utilisés séparément comme moteur de calcul dans le modèle d'optimisation. Les sorties issues de l'application de ces algorithmes sont comparées à fin de déterminer l'algorithme le plus adéquat à ce genre de problèmes.

Le paramétrage de cette famille de méthodes d'optimisation joue un rôle important dans leurs convergences. Dans notre papier (Dahmani and Yebdri 2020, Annexes), nous avons confirmé ce rôle à travers une étude de sensibilité des algorithmes RGA, GSA et HPSOGWO. Dans ce qui suit, les valeurs des paramètres conduisant aux meilleurs résultats présentés dans (Dahmani and Yebdri 2020) sont gardées à l'exception la taille de population de recherche et le nombre d'itérations. Ces derniers doivent être suffisamment agrandis pour tenir le nombre important de variables à déterminer (120 lâchers mensuels) et pour donner aux algorithmes suffisamment de tentations dans l'espérance d'atteindre la solution optimale globale. Le tableau 9.1 dévoile le paramétrage opté à ces algorithmes. Concernant le métaheuristique IGWO, la valeur adoptée du paramètre intrinsèque  $\mu_{IGWO}$  est celle proposée par l'auteur de l'algorithme (Long et al. 2017).

La taille de la population de recherche ainsi le nombre d'itérations sont gardés identiques pour les quatre algorithmes. Dans cette étude, chaque algorithme effectue  $(150 \times 10^4 =$

$1.5 \times 10^6$ ) opérations d'évaluation des solutions pour chaque exécution (essai). Ceci permet de réaliser le même nombre d'évaluations des solutions au cours du processus d'optimisation et de conduire à une comparaison plus judicieuse entre les algorithmes.

Les trois scénarios d'optimisation sont lancés au moins cinq (05) fois pour chaque algorithme. Les meilleurs résultats sont présentés et discutés dans ce qui suit. La synthèse des résultats relatifs à l'ensemble des essais fait l'objet de la fin du chapitre. L'examen de multiples essais permet de juger la stabilité et la vitesse d'exécution moyenne de chaque algorithme.

TABLE 9.1 – Paramétrage des algorithmes d'optimisation RGA, GSA, HPSOGWO et IGWO

<i>RGA</i>				
Taille de population	Nbr. d'itérations	Méthode de croisement	Fréquence de mutation	
150	10'000	1 point aléatoire	0.2	
<i>GSA</i>				
Taille de population	Nombre d'itérations	$G_0$	$\alpha_g$	
150	10'000	20	10	
<i>HPSOGWO</i>				
Taille de population	Nombre d'itérations	$C_1$	$C_2$	$C_3$
150	10'000	0.6	0.6	0.6
<i>IGWO</i>				
Taille de population	Nombre d'itérations		$\mu_{IGWO}$	
150	10'000		1.1	

## 9.4 Critères de performance

Plusieurs critères (ou indicateurs) de performance sont proposés dans la littérature pour évaluer la gestion de tels systèmes de ressource en eau (Hashimoto et al. 1982 ; Moy et al. 1986 ; Mathlouthi and Lebdi 2008, ... etc.). Les plus utilisés sont dits "critères quantitatifs" (François 2013). Ils sont appliqués sur les sorties du modèle d'optimisation, lesquelles sont obtenues séparément via différents algorithmes d'optimisation. Le recours à un seul indicateur ne reflète plus la performance globale d'un système ou d'une politique de gestion d'un système, donc recourir à plusieurs indicateurs s'avère nécessaire.

### 9.4.1 La valeur finale de la fonction objectif

Le premier indicateur de performance considéré dans l'évaluation des résultats est la meilleure valeur de la fonction objectif ( $F_{obj}$ ) atteinte à la fin du processus d'optimisation (à itération  $10^4$ ) relative au meilleur agent de recherche. Cette valeur est nommée "best fitness" dans le jargon de l'optimisation métaheuristiques. Elle concerne la meilleure solution obtenue, c-à-d., la meilleure distribution temporelle (mensuelle) des lâchers, représentée par le vecteur ( $\vec{R} = \{R_1, R_2, \dots, R_{T=120}\}$ ) dans ce cas. Ce critère exprime la qualité des lâchers (la satisfaction de la demande) d'une part et la convergence des algorithmes d'autre part.

### 9.4.2 Volumes déversés

L'indicateur des volumes déversés ( $V_{Dev}$ ) se définit comme le volume d'eau total déversé référé à une période de simulation donnée (équation 9.1) (François 2013). Malgré que ce volume peut être utilisé par les consommateurs (notamment les agriculteurs sur les berges des cours d'eau), il n'est pas considéré fructueux pour le gestionnaire du réservoir. En effet, la limitation de ce volume est un objectif indirect de la gestion, car il influe négativement sur le stockage et les lâchers fournis aux consommateurs.

$$V_{Dev} = \sum_{t=1}^T O_t \quad (9.1)$$

Tel que :

- $O_t$  : volume déversé (overflow en anglais) durant la période  $t$ .

### 9.4.3 Fiabilité volumétrique

La fiabilité volumétrique ( $a_v$ ) reflète la relation entre les lâchers d'eau et la demande en aval de la retenue pendant la période d'exploitation prise en compte  $T$  (Hashimoto et al. 1982). Elle est donnée par l'équation suivante :

$$a_v = \frac{\sum_{t=1}^T R_t}{\sum_{t=1}^T D_t} \quad (9.2)$$

### 9.4.4 Fiabilité temporelle

La fiabilité temporelle  $a_t$  se base sur le temps. Elle exprime en pourcentage (%) le nombre de périodes dans lesquelles le système a réussi d'atteindre les objectifs tracés (Hashi-

moto et al. 1982). L'équation suivante (Eq. 9.3) exprime mathématiquement la fiabilité temporelle :

$$a_t = \frac{T - f}{T} \quad (9.3)$$

Tels que :

- $T$  : durée totale considérée ( $T = 120\text{mois}$  dans ce cas) ;
- $f$  : nombre de périodes déficitaires (ou de défaillance) (failure periods en anglais).

Vue l'incertitude évidente dans l'estimation des demandes en eau futures des usagers, une période  $t$  est considérée déficitaire si la différence relative entre la demande  $D_t$  et le lâcher  $R_t$  dépasse un certain seuil. Ce dernier est difficile à le quantifier exactement et nécessite une étude approfondie du modèle de prévision de la consommation. Cependant, dans cette étude on a opté pour un seuil de 1% (équation 9.4).

$$\frac{D_t - R_t}{D_t} > 0.01 = 1.00\% \rightarrow \text{Défaillance.} \quad (9.4)$$

#### 9.4.5 Résilience

Il est défini par le temps qu'il faut à un système pour revenir à un état normal suivant un état de défaillance (déficit ou échec) (Hashimoto et al. 1982). Ce critère s'exprime comme suit :

$$\gamma = \frac{f_s}{f} \quad (9.5)$$

Tels que :

- $\gamma$  : indice de résilience ;
- $f$  : nombre de périodes déficitaires ;
- $f_s$  : nombre de séries de périodes déficitaires (failure series en anglais). Une série de périodes déficitaires est une séquence de périodes successives d'échec précédées et suivies de périodes de réussite (satisfaction).

#### 9.4.6 Vulnérabilité

L'indice de vulnérabilité ( $\sigma$ ) mesure la gravité des défaillances (Hashimoto et al. 1982). Il est donné par la formule suivante :

$$\sigma = \frac{\sum_{j=1}^{f_s} \max(Sh_j)}{f_s} \quad (9.6)$$

Tel que :

- $Sh_j$  : le déficit (shurtage en anglais) maximal au cours de la série de périodes déficitaires d'indice  $j$ .

Le critère de vulnérabilité  $\sigma$  peut être exprimé en unité de volume si le déficit est valorisé par la différence simple entre la demande en eau d'une période  $t$  et le lâcher correspondant ( $Sh_t = D_t - R_t$ ). De même, il peut être exprimé en pourcentage (%), si le déficit est valorisé par la différence relative entre la demande et le lâcher ( $Sh_t = \frac{D_t - R_t}{D_t}$ ). Les deux expressions sont utilisées par la suite.

#### **9.4.7 Le degré de couverture aux échelles taux de satisfaction - périodes satisfaites**

Ce critère exprime à quel degré une distribution temporelle des lâchers a maximisé deux (02) objectifs à la fois. Le 1<sup>er</sup> est la maximisation de taux "ou pourcentage" (noté  $\tau_s$ ) de satisfaction "ou de couverture" de la demande ciblée par l'optimisation. Le 2<sup>ème</sup> est la maximisation de nombre (ou la fréquence, notée  $N_{ps}$ ) de périodes satisfaites en demande.

Étant donnée une règle de distribution temporelle des lâchers du réservoir ( $\{R_t\}$ ,  $t = 1, 2, \dots, T$ ). Pour un taux de satisfaction minimal donné  $\tau_s$ , la valeur  $N_{ps}$  est déterminée par le nombre de périodes (années, mois, décades, semaines, jours, ...) dont la satisfaction de la demande est *supérieure ou égale* au taux  $\tau_s$ . Par une autre manière,  $N_{ps}$  est le nombre de périodes qui peuvent être couvertes au minimum par le taux  $\tau_s$ . Par conséquent, une règle parfaite de répartition temporelle de la ressource en eau assure la satisfaction totale ( $\tau_s = 100\%$ ) de la demande des usagers sur la durée totale considérée ( $N_{ps} = 100\%$ ).

Pour quantifier cet indicateur numériquement, nous proposons de mesurer la surface au-dessous de la courbe exprimant la variation de nombre des périodes ( $N_{ps}$ ) couvertes au minimum à un taux de satisfaction donné ( $\tau_s$ ). La courbe est exprimée par la fonction ( $N_{ps} = f(\tau_s)$ ), tandis que la surface (aire) au-dessous de cette courbe est notée  $A_s$ . L'utilisation pratique de ce critère est expliquée dans la suite avec plus de détails.

### **9.5 Optimisation de la gestion du réservoir H. Boghrara**

#### **9.5.1 Scénario 1 : Apports mensuels très faibles**

Le 1<sup>er</sup> scénario est basé sur les apports prévus par le réseau de neurones artificiels évolutionnaires "EANN-HPSOGWO-BP-6 (6, 9)" (figure 9.1). Le réseau de neurones est doté de deux (02) couches cachées, composées respectivement par six (06) et neuf (09) neurones artificiels cachés. Il utilise à l'entrée six (06) apports mensuels ( $Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}, Q_{t-6}$ ) de six (06) mois qui précèdent l'apport à prévoir  $Q_t$  du mois cible  $t$ . Le métaheuristique HP-SOGWO est utilisé dans la phase de recherche, tandis que l'algorithme BP est utilisé dans

la phase d'apprentissage. Les apports générés par ce réseau de neurones sont très faibles ( $\leq 5Mm^3/mois$ ) sur la majorité de la période de prévision et même inférieurs aux apports moyens mensuels de l'historique observé (septembre 1972 au décembre 2016) du barrage Hammam Boghrara. En effet, ce scénario simule une période de sécheresse prolongée sur plusieurs années.

### 9.5.1.1 Résultats et discussion

La figure 9.3 présente l'évolution de la fonction objectif à minimiser (décrite dans la section 9.2) au fil du processus d'optimisation. À partir de cette représentation, il est évident que les algorithmes RGA, GSA et HPSOGWO finissaient la dernière itération ( $k = 10^4$ ) par atteindre la même valeur (approximativement) de cette fonction. Cependant, ce n'est pas le cas pour l'algorithme IGWO. L'examen des valeurs numériques montre que l'algorithme HPSOGWO a convergé vers la valeur minimale trouvée ( $F_{obj} \cong 5.1$ ) à partir de l'itération 181, suivi par le GSA à partir l'itération 1068 et plus loin par le RGA vers l'itération 2091. En général, les 100 premières itérations ont été suffisantes aux algorithmes GSA et HPSOGWO pour atteindre une précision suffisante. En revanche, dès l'itération 9 l'algorithme IGWO a tété piégé dans un minimal local dont la valeur de la fonction objectif est égale à 7.22. À la fin, il a achevé le processus d'optimisation sans amélioration des résultats (tableau 9.2).

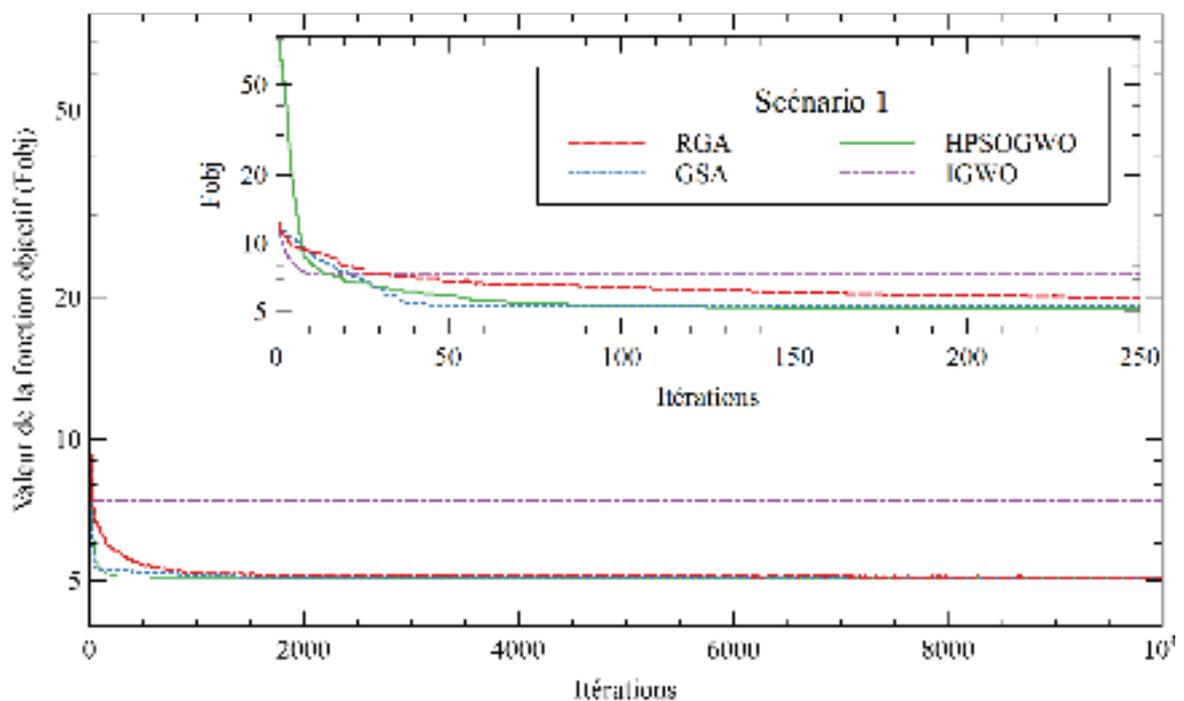


FIGURE 9.3 – Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 1-

La figure 9.4 montre les lâchers d'eau mensuels optimisés par les 04 algorithmes comparés et confrontés à la demande mensuelle cible en aval du barrage Hammam Boughrara.

Premièrement, les lâchers produits par les algorithmes RGA, GSA et HPSOGWO sont nettement inférieurs à la demande mensuelle sur la totalité de la période d'optimisation (120 mois). Deuxièmement, les lâchers générés par ces deux derniers algorithmes (GSA et HPSOGWO) sont parfaitement confondus sur la totalité des 120 mois, et légèrement différents dans certains mois à ceux générés par l'algorithme RGA. Au contraire, les lâchers optimisés par l'algorithme IGWO se diffèrent des autres algorithmes. Dans quelques mois ils couvrent l'allocation en eau ciblée, mais dans d'autres ils amènent à des déficits plus importants.

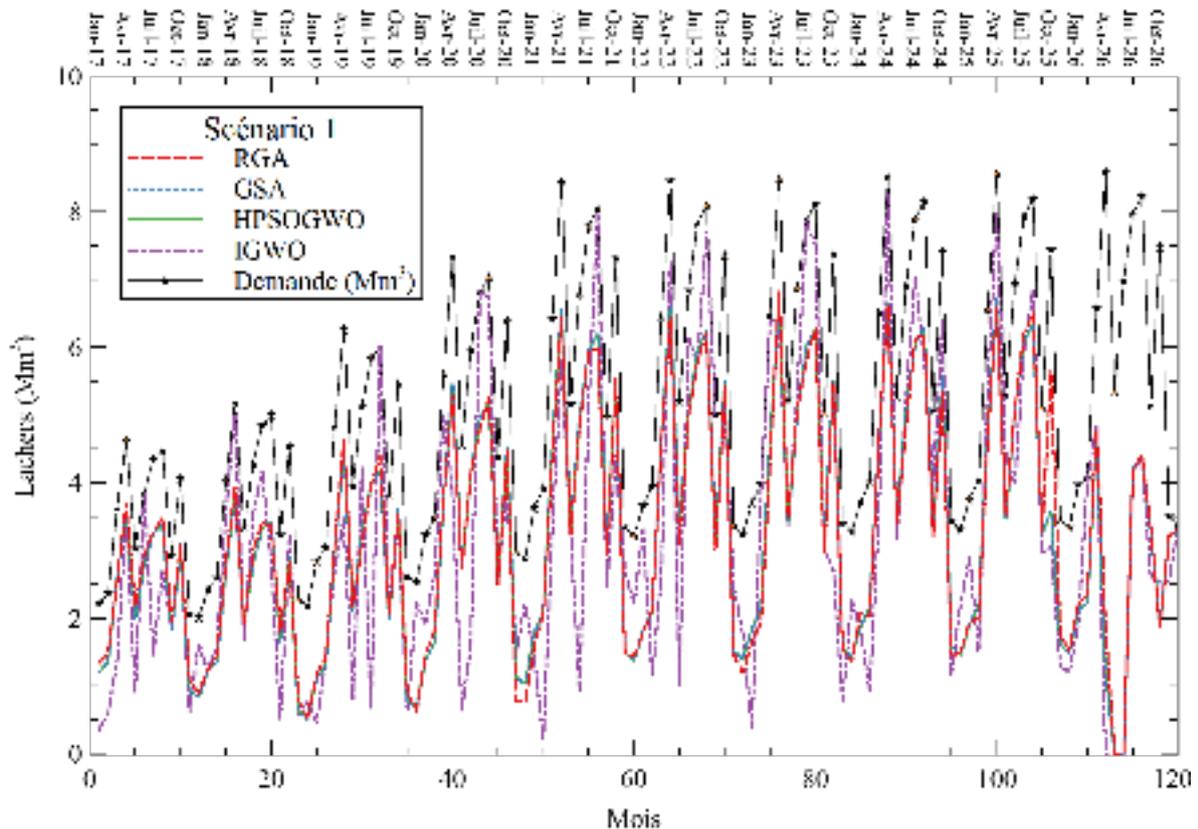


FIGURE 9.4 – Confrontation des lâchers mensuels optimisés par les différents algorithmes à la demande en eau mensuelle ciblée - Scénario 1 -

Les différents indicateurs de performance sont annoncés dans le tableau 9.2. L'examen de ceux-ci est primordial dans le but de quantifier numériquement l'efficacité des distributions temporelles fournies par les différents algorithmes d'optimisation.

Commençant par les volumes déversés durant les 120 mois. L'algorithme RGA a produit le plus faible volume ( $V_{Dev} = 21.87Mm^3$ ), suivi par GSA et HPSOGWO qui ont produit identiquement le même volume ( $V_{Dev} = 23.39Mm^3$ ), et en fin par IGWO ( $V_{Dev} = 26.08Mm^3$ ).

Les quantités d'eau déversées sont considérées perdues, donc un volume déversé plus faible implique une performance plus élevée. Examinant la figure 11 (Annexes), le déversement est produit au début de la durée sujet d'optimisation (mars et avril 2017, puis d'octobre 2017 au mai 2018). Cela s'explique par l'apport relativement fort prévu au mois de mars 2017 ( $18.23Mm^3$  environ) succédant un remplissage important du réservoir au mois de décembre 2016 ( $153Mm^3$  environ) avec une faible consommation dans cette période (voir la figure 9.1). Le regroupement de ces trois facteurs a conduit au remplissage total du réservoir et par conséquent des déversements.

L'échec marqué sur l'ensemble des algorithmes d'optimisation est que même avec le surplus d'eau disponible en ces moments, ils n'ont pas fourni des lâchers couvrant à 100% la demande des consommateurs. Cela s'explique par la stagnation des algorithmes dans des solutions optimales locales.

TABLE 9.2 – Indicateurs de performance relatifs au scénario 1

Scén <sup>a</sup>	Algorithme	$F_{obj}$ <sup>b</sup>	$V_{Dev}$ ( $Mm^3$ )	Fiabilités		Résilience	Vulnérabilité	
				$a_v$ (%)	$a_t$ (%)	$\gamma$ (%)	$\sigma$ (%)	$\sigma(Mm^3/mois)$
1	RGA	5.08	21.87	63.24	0.0	0.0	100	6.98
	GSA	5.06	23.39	62.97	0.0	0.0	100	7.44
	HPSOGWO	5.06	23.39	62.97	0.0	0.0	100	7.43
	IGWO	7.22	26.08	62.34	7.5	8.11	82.09	4.28

<sup>a</sup> Scénario d'apports.

<sup>b</sup> Best fitness.

L'évaluation de la fiabilité volumétrique  $a_v$  (tableau 9.2) exprimant le rapport de la somme des lâchers mensuels sur la demande totale, indique une efficacité approximativement égale de l'ensemble des algorithmes d'optimisation avec une légère supériorité au profit de RGA suivi par GSA et HPSOGWO, dont ils ont prouvé la même fiabilité volumétrique. Cette dernière n'a pas dépassé 63.24% pour l'ensemble des algorithmes, donc un déficit d'eau en aval supérieur à 36.75%.

De fait la convergence entre les résultats aboutis par les trois premiers algorithmes d'optimisation (RGA, GSA et HPSOGWO), ils ont signalé les mêmes valeurs de la fiabilité temporelle  $a_t$ , la résilience  $\gamma$  et de la vulnérabilité  $\sigma$ (%). La fiabilité basée sur le temps et la résilience sont nulles ( $a_t = \gamma = 0\%$ ), car en aucune période (mois) les trois algorithmes ont satisfait la demande en eau mensuelle à 100%. Le système (bassin versant "apports" - réservoir) n'a pas pu reprendre un état de réussite (couverture totale de la demande), donc il n'est pas résilient d'une part, et d'autre part il est totalement vulnérable ( $\sigma = 100\%$ ).

À l'opposé, la fiabilité temporelle relevée de lâchers optimisés par l'algorithme IGWO est meilleure par rapport aux autres. Elle est de 7.5%, ce qui signifie que l'IGWO a généré des apports comblant la demande entière de 9 mois sur 120. Par conséquent, la distribution des lâchers donne une certaine résilience au système de gestion (de l'ordre de 8.11%), et réduit sa vulnérabilité ( $\sigma = 82.09\%$ ).

L'expression de la vulnérabilité par unité de volume ( $\sigma(Mm^3/mois)$ ) indique l'ampleur de la défaillance prévue. La plus faible valeur de vulnérabilité est trouvée pour l'algorithme IGWO, puis RGA et finalement pour HPSOGWO et GSA respectivement. Ces valeurs sont provoquées notamment par les défaillances observées aux mois 113 et 114 correspondant aux mois de mai et juin 2026.

La confrontation des critères FRV (Fiabilité, Résilience et Vulnérabilité) face aux valeurs de la fonction objectif et des volumes déversés surgit une contradiction. La considération des critères FRV seuls réclame la supériorité des résultats aboutis par l'algorithme IGWO devant les autres. Au contraire, la prise en compte des valeurs de la fonction objectif et des volumes déversés seuls conduit au classement de l'algorithme RGA en top, puis GSA et HPSOGWO à la même position et finalement l'IGWO en bas (la dernière position).

L'arbitrage dans cette contradiction nous a conduit à une analyse plus poussée des résultats. La figure 9.5 montre la variation de nombre de périodes (mois) ( $N_{ps}$  exprimé en nombres dans l'intervalle  $[0, 120]$  ou encore en (%) dans l'intervalle  $[0\%, 100\%]$ ) satisfaites - au minimum - à différents taux de satisfaction ( $\tau_s \in [0\%, 100\%]$ ) de la demande en eau des usagers. La lecture du graphique de la figure indique qu'une satisfaction au minimum à  $\tau_s = 70\%$  (par conséquent 30% de restriction) de la demande mensuelle requise, ne touche que 47 mois/120 le cas des lâchers optimisés par RGA, 44 mois/120 le cas de GSA et HPSOGWO simultanément, et finalement 48 mois/120 le cas d'IGWO. Aux taux de satisfaction  $\tau_s \geq 68\%$ , ce dernier algorithme continue à prouver un nombre de mois satisfaits ( $N_{ps}$ ) supérieur aux autres. Néanmoins, pour les taux  $\tau_s < 68\%$  les lâchers produits par les trois premiers algorithmes (RGA, GSA et HPSOGWO) permettent de couvrir la demande d'une durée plus importante que ceux produits par l'IGWO.

La surface (notée  $A_s$ ) au-dessous d'une courbe  $N_{ps} = f(\tau_s)$  permet de quantifier globalement à quel degré les lâchers optimisés peuvent couvrir de périodes de fonctionnement du réservoir. Elle exprime en quelque sorte la maximisation de deux (02) objectifs à la fois, 1) le taux de satisfaction de la demande  $\tau_s$ , et 2) la fréquence de périodes satisfaites  $N_{ps}$ . À partir de là, on peut constater clairement que dans le cas d'une satisfaction entière de la demande cible ( $\tau_s = 100\%$ ) sur la durée totale d'optimisation considérée ( $N_{ps} = 100\%$ ), la surface  $A_s$  au-dessous de la courbe  $N_{ps} = f(\tau_s)$  sera maximale (couvre la totalité du graphe). Dans le cas contraire, c-à-d., lorsque une défaillance absolue se produit (lâchers nuls sur toute la durée

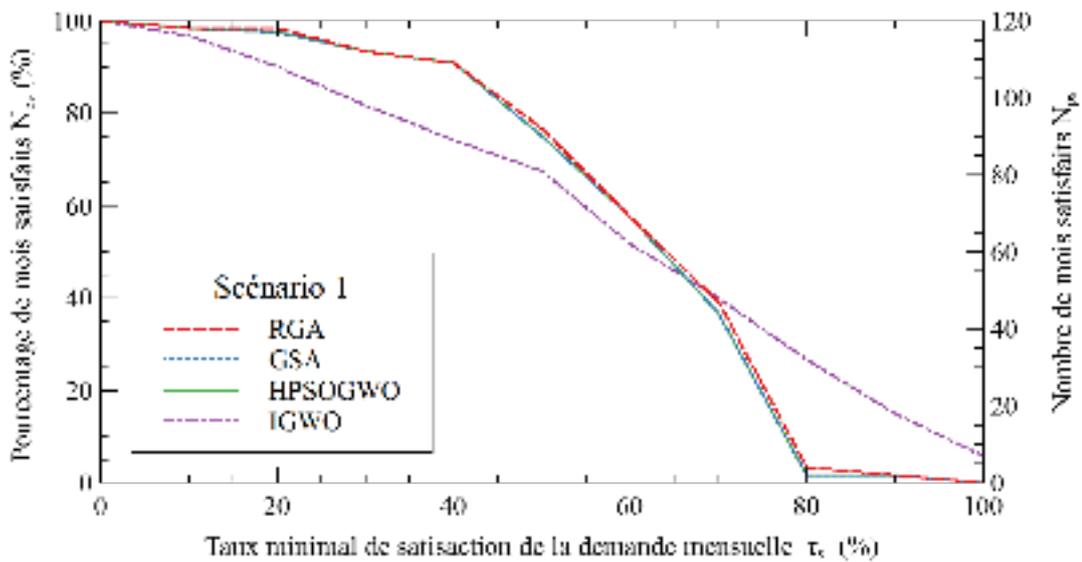


FIGURE 9.5 – Variation de nombre de mois  $N_{ps}$  satisfaits -au minimum- à différents taux de satisfaction de la demande  $\tau_s$  (Scénario 1). Par exemple, 117 demandes mensuelles peuvent être satisfaites au minimum à 20% (algorithmes RGA, GSA et HPSOGWO). La satisfaction de la demande au minimum à 20%, touche 108 mois au maximum (algorithme IGWO)

considérée), la surface  $A_s$  sera minimale.

Le tableau 9.3 énumère les résultats de calcul des surfaces  $A_{si}$  au-dessous des courbes  $N_{ps} = f(\tau_s)$  dérivées des lâchers optimisés par les quatre algorithmes. Il est clair que l'algorithme RGA a démontré la surface  $A_s$  la plus importante, ce qui signifie que ses lâchers peuvent couvrir le nombre de mois  $N_{ps}$  le plus important par les taux de satisfactions  $\tau_s$  les plus importants en comparaison avec les autres algorithmes. En deuxième rang, on trouve les algorithmes GSA et HPSOGWO simultanément à cause de leurs résultats similaires indiqués auparavant. En dernier rang, l'algorithme IGWO est classé. Néanmoins, les différences entre les surfaces ne sont pas importantes.

TABLE 9.3 – Surfaces au-dessous des courbes  $N_{ps} = f(\tau_s)$  relatives aux lâchers optimisés par les quatre algorithmes d'optimisation -Scénario 1-

Algorithme d'optimisation	RGA	GSA	HPSOGWO	IGWO
Surface $A_s$ au-dessous de la courbe $N_{ps} = f(\tau_s)$	73.10	72.30	72.30	71.55

L'inspection de l'évolution du stockage de l'eau dans la retenue du barrage au fil du temps "trajectoires de réserve" (figure 11a, Annexes) indique un remplissage total du réservoir Hammam Boughrara aux mois 4 et 5 (avril et mai 2017), et du mois 10 au 18 (octobre 2017 au juin 2018). Dans ces périodes, des déversements sont produits (figure 11b, Annexes) en fonction de l'état du stockage. Par la suite, une vidange progressive est constatée du réservoir à

partir du mois 19 (juillet 2018) jusqu'à la fin de la période d'optimisation considérée (mois 120 "décembre 2026"). De plus, à partir du mois 113 (mai 2026) la réserve d'eau atteindra le volume mort ( $S_{t=113} = S_{min} = 23.0Mm^3$ ). Cela due au fait de l'insuffisance des apports prévus dans ce scénario d'une part, et l'augmentation de la demande en eau provoquée par l'expansion de la surface agricole du périmètre irrigué de la région de Maghnia d'autre part. Les trajectoires de réserve d'eau dans la retenue issues des algorithmes RGA, GSA et HPSOGWO sont légèrement différentes à celle dérivée par l'IGWO. Cela due à différence entre les lâchers optimisés par ces algorithmes. D'une manière générale, ces résultats proclament le respect des contraintes sur la capacité minimale du stockage et les lâchers maximaux imposées sur les algorithmes au cours du processus d'optimisation la gestion du réservoir. Hormis, les insatisfactions observées de la demande aux moments de déversements indiquent que les solutions (lâchers optimisés) obtenus sont pénalisées d'un part, et ne représente pas la solution optimale globale d'autre part.

Finalement, la figure 11c (Annexes) présente le tracé graphique des estimations de volumes évaporés. Il est certain que ces derniers sont proportionnels au stockage dans la retenue, car l'augmentation de la réserve d'eau dans le réservoir implique l'expansion de la surface du plan d'eau, et par conséquent la hausse des pertes par évaporation. En conséquence, les différences entre ces estimations sont relatives aux celles entre les trajectoires de la réserve restituées des lâchers optimisés.

### **9.5.1.2 Synthèse des principaux résultats**

À travers l'étude de ce scénario on peut ressortir les points essentiels suivants :

— La prise en compte de l'ensemble d'indicateurs de performance implique le classement des algorithmes d'optimisation testés du plus fort au plus faible dans ce scénario dans l'ordre suivant :

1. RGA ;
2. GSA et HPSOGWO (même rang) ;
3. IGWO.

Rappelons que le choix des paramètres intrinsèques des trois premiers algorithmes est fixé à la base d'une étude de leur sensibilité (indiquée dans (Dahmani and Yebdri 2020)). Or, pour l'algorithme IGWO l'unique paramètre intrinsèque  $\mu_{IGWO}$  est fixé à la valeur ( $\mu_{IGWO} = 1.1$ ) adoptée par l'auteur de l'algorithme (Long et al. 2017). Donc, l'étude de sensibilité de l'IGWO par rapport à son paramètre peut conduire à l'amélioration des résultats.

— L'ensemble des algorithmes ont respecté les contraintes imposées par le modèle général de l'optimisation en ce qui concerne la capacité minimale de stockage

et les lâchers maximaux qui ne doivent pas dépasser les demandes en eau mensuelles. Néanmoins, ils ont échoué de délivrer les demandes requises aux moments de déversement. Cela signifie que les solutions atteintes (lâchers optimisés) sont pénalisées et ne constituent pas l'optimal global, malgré que le nombre d'évaluations des solutions au cours d'optimisation est relativement important (fixé à 1.5 Millions =  $1.5 \times 10^6$  évaluations). L'augmentation de ce dernier par l'agrandissement de la taille de population de recherche et/ou le nombre d'itérations peut améliorer les résultats.

- Le recours au nombre limité de critères de performance peut conduire à une estimation illusoire de la performance globale d'un système de gestion. La diversification des critères en fonction des objectifs voulus est indispensable.
- L'évaluation de l'aire ( $A_s$ ) au-dessous de la courbe  $N_{ps} = f(\tau_s)$  exprimant la variation de nombre de périodes  $N_{ps}$  satisfaites à certain taux  $\tau_s$  en fonction de ce dernier, permet l'évaluation des résultats à base de la maximisation de deux (02) objectifs à la fois, le taux de satisfaction de la demande  $\tau_s$  et le nombre de périodes satisfaites  $N_{ps}$ .
- Une succession de périodes sèches (sécheresse prolongée sur plusieurs années) accompagnée d'une consommation irrationnelle (ne subissant pas des restrictions intensives) engendre l'épuisement de la réserve d'eau du barrage Hammam Boughrara et augmente la défaillance du système de gestion de la ressource en eau axé sur ce barrage. Cela induit un impact très néfaste sur les consommateurs de l'eau du barrage Hammam Boughrara, notamment la surface agricole irriguée qui constitue le consommateur le plus exigeant en termes de volumes d'eau à allouer.

### **9.5.2 Scénario 2 : Apports mensuels produisant des volumes proches à la moyenne annuelle**

Ce scénario se base sur les apports prévus par régression au moyen du réseau de neurones artificiels évolutionnaire "EANN-HPSOGWO-LM-4 (8, 13)" (figure 9.1). L'algorithme d'optimisation exploité à fin d'aboutir à ce réseau est le HPSOGWO, pendant que l'algorithme Levenberg-Marquardt "LM" est utilisé pour leur entraînement. Ce réseau de neurones consomme à l'entrée quatre (04) débits mensuels consécutifs ( $Q_{t-1}$ ,  $Q_{t-2}$ ,  $Q_{t-3}$  et  $Q_{t-4}$ ) plus l'indice du mois ciblé par la prévision  $t \in [1, 12]$  réduit dans l'intervalle  $[0, 1]$ , pour produire le débit prévu  $Q_t$  à ce mois. Deux (02) couches cachées constituent la structure de ce réseau de neurones artificiels. Elles comptent respectivement huit (08) et treize (13) neurones artificiels.

Les apports prévus dans ce scénario entraînent des volumes annuels proches à ceux générés par les apports moyens mensuels du réservoir Hammam Boughrara. Ils présentent des pics automnaux forts ( $\cong 18.5Mm^3/mois$ ) (mois de novembre généralement) et des apports

estivaux très faibles ( $< 0.5 Mm^3/mois$ ). Donc, ces apports permet de simuler "approximativement" un remplissage automnal/hivernal, puis une vidange printanière/estivale du réservoir avec des volumes d'apports proches à la moyenne annuelle. De plus, ce scénario peut remplacer un autre basé sur les apports moyens mensuels. Ce dernier est revendiqué par les valeurs négatives du critère *Nash* atteintes par pas mâle de réseaux de neurones artificiels durant la phase de prévision.

### 9.5.2.1 Résultats et discussion

La figure 9.6 communique l'évolution des valeurs de la fonction objectif  $F_{obj}$  au fil du processus d'optimisation parcouru par les quatre algorithmes sujets de comparaison. Pareils au scénario précédent, les métaheuristiques RGA, GSA et HPSOGWO ont convergé vers la même valeur de la fonction objectif ( $F_{obj} \cong 1.1$ , tableau 9.4), avec une légère performance en faveur des deux derniers (GSA et HPSOGWO). Encore, l'algorithme IGWO n'a pas rapproché à cette valeur et resté stagné loin jusqu'à la fin des itérations.

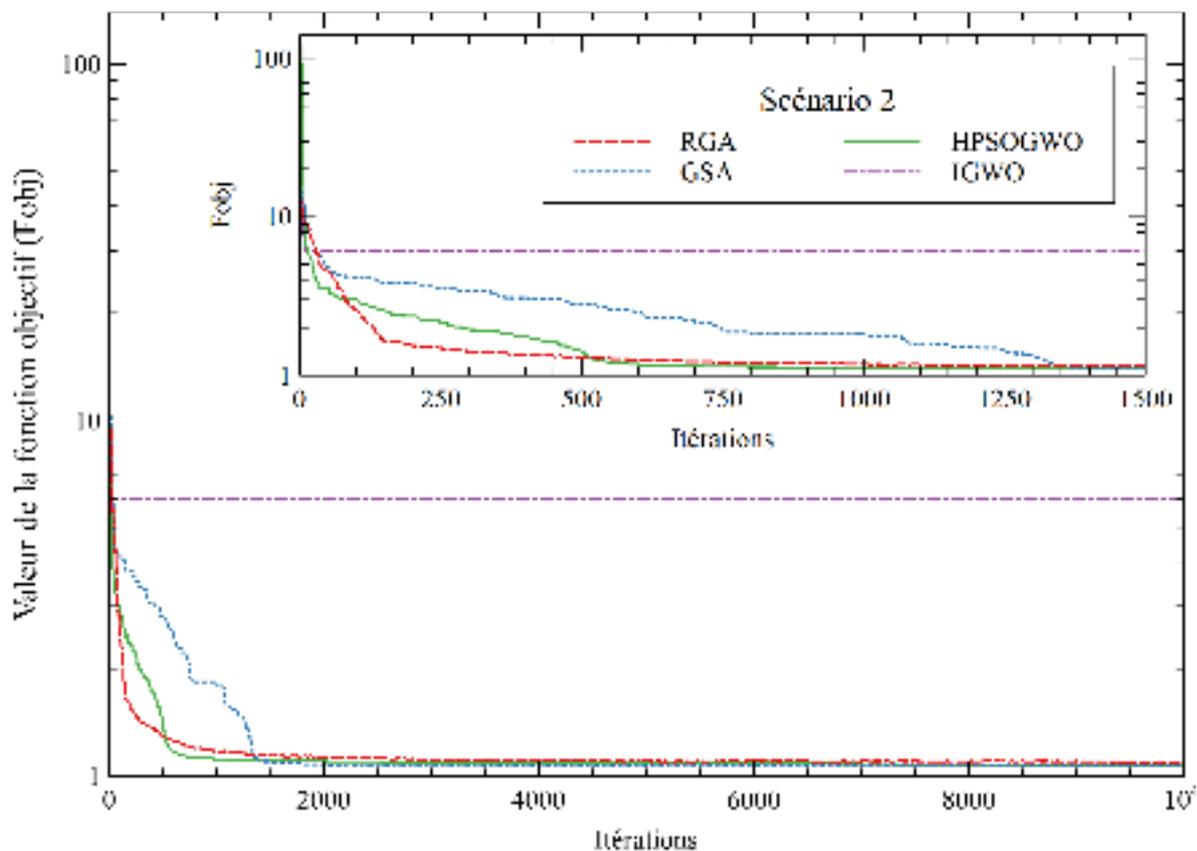


FIGURE 9.6 – Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 2-

L'algorithme RGA a présenté la tendance de convergence la plus rapide dans l'in-

tervalle des 500 premières itérations. Hormis, l'expertise des valeurs numériques prouve la convergence du HPSOGWO en premier (à l'itération 660), secondé par le GSA (à l'itération 1344), puis le RGA (à l'itération 1358). Pour ces algorithmes, 1300 tentatives "itérations" ont été suffisantes pour rejoindre une précision suffisante du calcul. À l'opposé, l'algorithme IGWO a stagné encore dès l'itération 12 dans un minimal local dont la fonction objectif donne  $F_{obj} = 6.05$  (tableau 9.4). Identiquement au premier scénario, l'IGWO n'a pas amélioré le résultat jusqu'à la fin du traitement itératif.

La figure 9.7 montre la comparaison entre les lâchers optimisés résultants et la demandes en eau à satisfaire par le réservoir Hammam Boughrara dans le contexte du 2<sup>ème</sup> scénario. La première constatation tirée de la figure souligne la superposition des trajectoires des lâchers optimisés par les algorithmes RGA, GSA et HPSOGWO, conséquence directe de leurs convergences simultanées à la même valeur de la fonction objectif. La deuxième constatation relève l'insuffisance des lâchers générés par ces trois algorithmes en termes de couverture entière des demandes mensuelles sauf celles de 02 mois (novembre et décembre 2026). Malgré ça, ces lâchers restent plus conséquents par rapport à ceux optimisés par l'algorithme IGWO sur la durée totale (120 mois), à l'exception de certains apports de ce dernier couvrant totalement la demande en eau voulue.

L'inspection des critères de performance relatifs aux résultats du scénario 2 (tableau 9.4) indique des déversements ( $V_{Dev}$ ) nuls résultants des lâchers optimisés par les trois algorithmes RGA, GSA et HPSOGWO, face à  $20.58Mm^3/120mois$  de déversement effet de lâchers optimisés par l'IGWO. Ce volume est la conséquence des déversements apparus aux mois de novembre et décembre des années 2018 et 2019 (figure 12b, Annexes). En addition aux valeurs de la fonction objectif, cet indicateur implique la supériorité des résultats obtenus par le RGA, GSA et HPSOGWO par rapport à l'IGWO. Encore, le tableau signale une fiabilité volumétrique ( $a_v$ ) similaire de l'ordre de 84.4% rapportée aux lâchers produits par les algorithmes RGA, GSA et HPSOGWO. La valeur de cet indicateur n'a pas dépassé 70.1% le cas des lâchers générés par l'algorithme IGWO. Au contraire, la fiabilité temporelle ( $a_t$ ) suggère une évaluation inverse des algorithmes d'optimisation. Les lâchers produits par l'IGWO prouvent une fiabilité temporelle de l'ordre de 12.5%, ce qui signifie une satisfaction entière des demandes mensuelles de  $15mois/120$ . Tandis que les lâchers des autres algorithmes n'assurent qu'une fiabilité temporelle de 1.67%, soit une satisfaction entière des demandes de  $2mois/120$ .

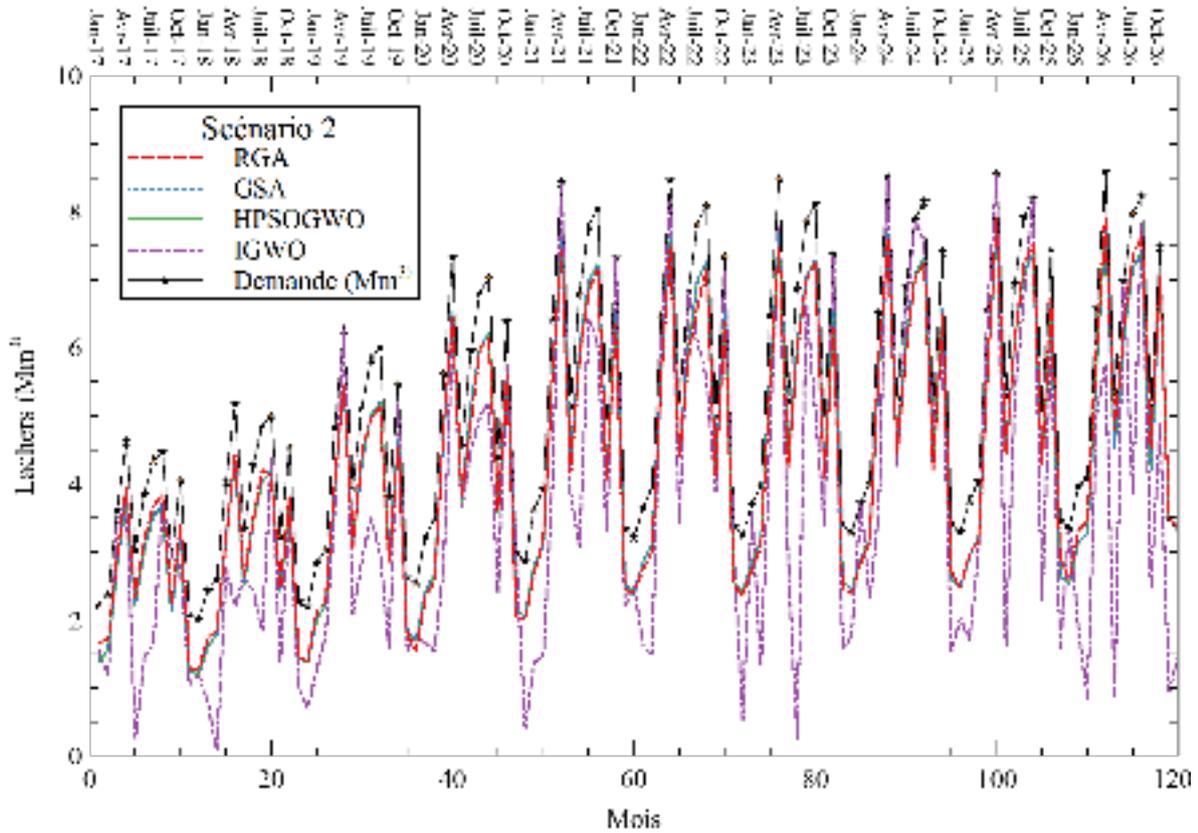


FIGURE 9.7 – Confrontation des lâchers mensuels optimisés par les différents algorithmes à la demande en eau mensuelle ciblée - Scénario 2 -

TABLE 9.4 – Indicateurs de performance relatifs au scénario 2

Scén <sup>a</sup>	Algorithme	$F_{obj}^b$	$V_{Dev}$ ( $Mm^3$ )	Fiabilités		Résilience	Vulnérabilité	
				$a_v$ (%)	$a_t$ (%)	$\gamma$ (%)	$\sigma$ (%)	$\sigma(Mm^3/mois)$
2	RGA	1.09	0.00	84.37	1.67	0.85	38.74	1.21
	GSA	1.07	0.00	84.43	1.67	0.85	40.30	0.95
	HPSOGWO	1.07	0.00	84.35	1.67	0.85	40.80	0.95
	IGWO	6.05	20.58	70.08	12.50	14.29	66.89	3.07

<sup>a</sup> Scénario d’apports.

<sup>b</sup> Best fitness.

Sous l’effet que la résilience juge la capacité d’un système à reprendre un état de réussite après un état de défaillance, elle peut exprimer d’une manière indirecte la fiabilité temporelle. Sur le tableau 9.4 toujours, les lâchers aboutis par les métaheuristiques RGA, GSA et HPSOGWO présentent une résilience très fiable ( $\gamma = 0.85\%$ ) face à celle achevée par les lâchers de l’IGWO ( $\gamma = 14.29\%$ ). Cela signifie une espérance faible de l’allocation parfaite de la

demande en eau en aval du réservoir Hammam Boughrara.

L'indice de vulnérabilité confirme ceux de la fonction objectif, de volumes déversés et de la fiabilité volumétrique. La vulnérabilité ( $\sigma(\%)$ ) classe les sorties des algorithmes d'optimisation de moins vulnérable au plus vulnérable dans l'ordre : RGA, GSA et HPSOGWO, puis IGWO. L'ampleur de vulnérabilité ( $\sigma(Mm^3/mois)$ ) varie d'un algorithme à l'autre. La plus fiable est celle relative aux algorithmes GSA et HPSOGWO ( $\sigma = 0.95Mm^3/mois$ ), suivi par celle liée à l'RGA ( $\sigma = 1.21Mm^3/mois$ ), et finalement celle rapporté à l'IGWO ( $\sigma = 3.07Mm^3/mois$ ).

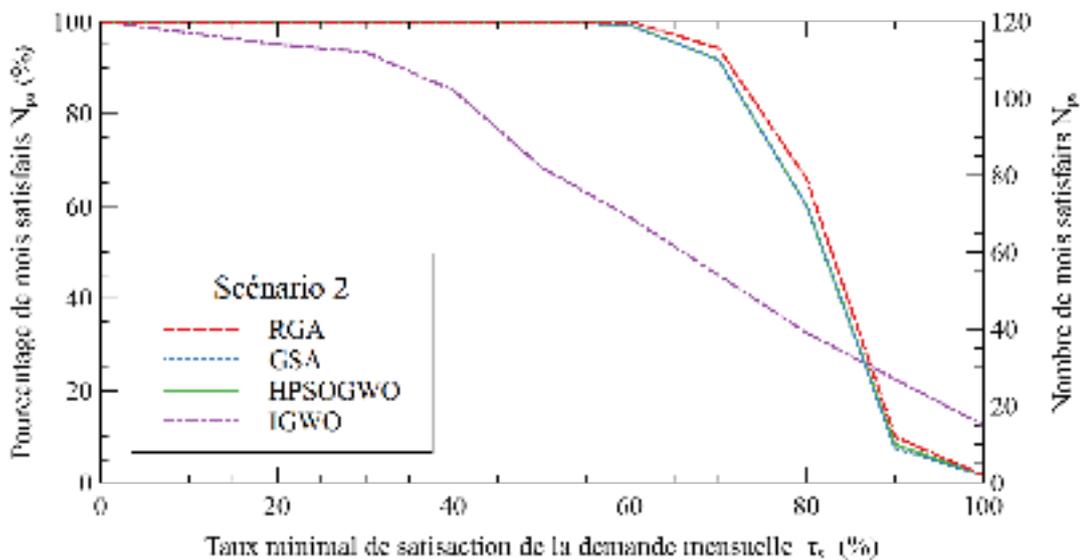


FIGURE 9.8 – Nombre de mois satisfaits en fonction de taux de satisfaction de la demande en eau mensuelle - Scénario 2-

La figure 9.8 illustre la variation de nombre de périodes (en mois)  $N_{ps}$  satisfaites au minimum aux taux d'allocation  $\tau_s(\%)$  ( $N_{ps} = f(\tau_s)$ ). La figure montre clairement la superposition des courbes dérivées de lâchers produits par les algorithmes GSA et HPSOGWO avec un rapprochement très fermé à la courbe produite par les sorties de l'RGA. Encore, elle indique que 100% (120 mois) de la durée concernée par l'optimisation peuvent être satisfaits au minimum à 60% le cas des lâchers produits par ces algorithmes. Au-delà de ce taux, le nombre  $N_{ps}$  diminue rapidement pour atteindre  $N_{ps} = 2mois$  dont l'allocation correspondante est parfaite ( $\tau_s = 100\%$ ).

Sous l'effet des résultats d'optimisation différents aux autres, la courbe  $N_{ps} = f(\tau_s)$  déduite des lâchers générés par l'algorithme IGWO est différente, laquelle se trouve en bas des autres courbes. Même avec un taux de satisfaction minimal ( $\tau_s = 40\%$ ), les lâchers générés par cet algorithme ne concerne que 100 mois (soit 83.33%) de la période totale d'optimisation. À partir de ( $\tau_s = 40\%$ ), le nombre de mois  $N_{ps}$  diminue linéairement jusqu'à aboutir ( $N_{ps} = 15$ )

lorsque le taux de satisfaction de la demande en eau soit parfait ( $\tau_s = 100\%$ ).

TABLE 9.5 – Surfaces au-dessous des courbes  $N_{ps} = f(\tau_s)$  relatives aux lâchers optimisés par les quatre algorithmes d’optimisation -Scénario 2-

Algorithme d’optimisation	RGA	GSA	HPSOGWO	IGWO
Surface $A_s$ au-dessous de la courbe $N_{ps} = f(\tau_s)$	98.50	97.10	97.20	78.35

Le tableau 9.5 affirme les résultats de calcul de la surface  $A_s$  au-dessous de chaque courbe exprimant la relation  $N_{ps} = f(\tau_s)$ . Comme évoqué précédemment, la surface  $A_s$  indique le degré de maximisation de deux (02) objectifs à la fois, le taux d’allocation de la demande et le nombre de périodes satisfaites à ce taux. Les résultats du tableau traduisent numériquement les surfaces au-dessous des courbes montrées dans la figure 9.8. La surface  $A_s$  issue des lâchers optimisés par le RGA est la plus importante. Elle est secondée par celle dérivée de lâchers de l’HPSOGWO et du GSA, puis finalement par celle issue des lâchers de l’IGWO.

Tout bien considéré dans ce scénario, les quatre algorithmes peuvent être classés de plus au moins performant dans l’ordre suivant : RGA en premier rang, GSA et HPSOGWO en deuxième rang, puis l’IGWO en dernier. Ce classement est identique à celui du premier scénario.

Finalement, la figure 12 (Annexes) illustre les trajectoires du stockage, le déversement et les volumes évaporés du réservoir Hammam Boughrara impliqués par les résultats d’optimisation. La superposition des courbes issues des algorithmes RGA, GSA et HPSOGWO d’un côté, ainsi leur déphasage à celle issue de l’IGWO d’autre côté confirme les résultats précédents. De plus, la figure 12a (Annexes) marque l’atteint du volume mort du réservoir par les trajectoires de réserve des trois premiers algorithmes. Ceci confirme le respect des contraintes de l’opération du barrage en premier lieu (la capacité minimale et maximale à ne pas dépasser) et explique la défaillance en deuxième lieu. En d’autre termes, l’épuisement du stockage disponible en addition des apports mensuels n’ont pas permis la satisfaction entière de la demande des usagers. En revanche, la trajectoire de réserve dérivée de l’optimisation par le métaheuristique IGWO indique un stockage plus important. Ceci est la conséquence des lâchers faibles générés par l’algorithme et la cause des volumes importants perdus par évaporation. Sans oublier que la surface du plan d’eau soumise à l’évaporation et par conséquence la perte par évaporation varient proportionnellement avec le stockage dans la retenue.

**9.5.2.2 Synthèse des principaux résultats**

L’étude du scénario 2 permet de tirer les résultats principaux suivants :

- D'une manière générale, les algorithmes d'optimisation RGA, GSA et HPSOGWO ont prouvé une performance relativement égale produisant des résultats de bonne qualité face à l'algorithme IGWO. Cette constatation confirme celle relevée du premier scénario et revendique le même classement d'algorithmes testés en terme d'efficacité dans ce genre de problèmes d'optimisation.
- L'ensemble de contraintes dictées par le modèle d'optimisation de la gestion d'un réservoir sont respectées par les algorithmes RGA, GSA et HPSOGWO, y compris celles des capacités minimale et maximale du stockage, des lâchers et de déversement.
- Le recours au nombre important de critères de mesure de performance offre l'évaluation de la fiabilité d'une politique de gestion sur une multitude de facettes (taux et fréquence de satisfaction de la demande ciblée, taux de restriction, fiabilités aux échelles de volumes et du temps, ... etc). Ceci peut conduire à une estimation globale plus judicieuse de la performance d'un système de gestion.
- Les apports liquides assurant un remplissage proche à celui produit par les apports moyens mensuels engendrent encore le risque de défaillance du système de gestion. Malgré que les apports dans ce scénario sont plus conséquents à ceux du scénario 1, le réservoir est épuisé à la fin de la durée d'optimisation adoptée. Les restrictions sur les lâchers sont inévitables, cependant elles doivent être complétées par une règle de répartition de la ressource en eau entre les usagers.

### **9.5.3 Scénario 3 : Séquence d'années humides**

Le scénario 3 est axé sur les apports mensuels au réservoir Hammam Boughrara prévus par le réseau de neurones artificiels conventionnel "ANN-LM-3 (6, 15)". Ce dernier est constitué de deux (02) couches cachées. La première compte six (06) neurones artificiels, tandis que la deuxième compte quinze (15). Les données à l'entrée du réseau sont les apports mensuels successifs ( $Q_{t-1}$ ,  $Q_{t-2}$  et  $Q_{t-3}$ ) en addition de l'indice du mois visé par la prévision  $t$  transformé dans l'intervalle  $[0, 1]$ . La sortie du réseau est l'apport estimé  $Q_t$  correspondant au mois  $t$ . L'apprentissage du réseau de neurones a été établi par l'algorithme Levenberg-Marquardt "LM".

Les apports prévus dans ce scénario (figure 9.1) représentent une alternance de plusieurs années humides dont les apports mensuels dépassent la moyenne mensuelle ainsi les demandes en eau en aval du réservoir sur la majorité de la période concernée par l'étude envisagée.

### 9.5.3.1 Résultats et discussion

La figure 9.9 présente les tendances de convergence des quatre algorithmes au fur et à mesure de progression du traitement itératif.

La convergence rapide et antérieure de l'algorithme RGA dans les premières 1000 itérations est claire. Au-delà, elle est de très faible ampleur jusqu'à la fin du processus itératif. A ce stade, le RGA a atteint la valeur "best fitness" ( $F_{obj} = 75.80$ , tableau 9.6). En second lieu on trouve celle du GSA. Ce dernier a convergé progressivement au cours des 4000 premières tentations, puis très lentement par la suite. Le GSA a terminé le traitement par la valeur ( $F_{obj} = 81.48$ ), ce qui marque sa stagnation dans un minimal local pas loin du RGA. En revanche, la tendance de convergence de l'algorithme HPSOGWO n'a pas accompagné les précédentes. Elle a éprouvé des stagnations dans des minimas locaux en plusieurs reprises à partir de la 20<sup>ème</sup> itération. À la fin, l'algorithme a terminé le processus par la valeur "best fitness" ( $F_{obj} = 170.64$ ), qui dépasse le double de celle aboutie par le GSA. De plus et en similitude avec les scénarios précédents, l'algorithme IGWO a présenté encore la tendance de convergence la plus défavorable parmi les autres. L'IGWO a été perduré dans un minimal local de la fonction objectif (à  $F_{obj} = 216.32$ ) dès l'itération 14. L'algorithme a achevé le traitement sans amélioration des résultats.

Malgré la convergence meilleure de l'algorithme RGA, les valeurs de la fonction objectif sont loin d'être nulles à cause de l'application de la pénalité liée au dépassement de la capacité du stockage maximale du réservoir Hammam Boughrara (équation 1.14). Concernant les autres métaheuristiques, les décalages entre les lâchers et la demande amplifient encore les valeurs de cette fonction en plus les différentes pénalités appliquées.

L'impact des résultats de convergence sur les lâchers optimisés est illustré dans la figure 9.10. Il est évident que les lâchers produits par l'algorithme RGA présentent une correspondance parfaite avec la demande en eau désirée, ce qui exprime la satisfaction entière de cette dernière. La même observation est relevée pour les lâchers calculés par le GSA, sauf qu'à partir du mois 82 (octobre 2023) une défaillance s'amplifiant au fil du temps est soulignée.

En ce qui concerne les lâchers générés par les algorithmes HPSOGWO et IGWO, ils sont loin de couvrir la demande en eau des usagers sauf de certaines périodes. Néanmoins, les lâchers calculés par l'algorithme HPSOGWO surpassent la demande dans quelques périodes. Ceci témoigne l'infraction de la contrainte visant le non-dépassement des demandes mensuelles par les lâchers correspondants. D'une manière générale, les déficits impliqués par les sorties de l'algorithme IGWO sont plus intenses face à ceux relatifs à l'HPSOGWO.

Le tableau 9.6 annonce les critères de performance du fonctionnement du réservoir

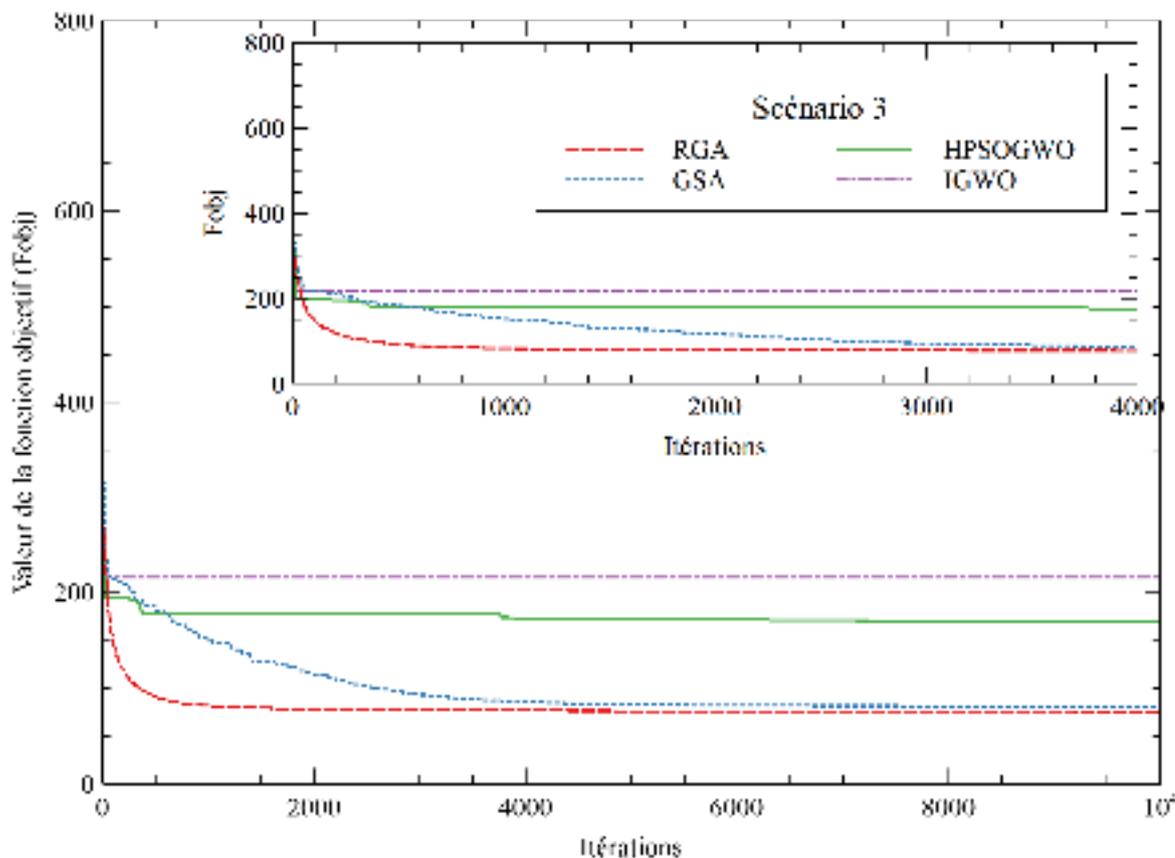


FIGURE 9.9 – Tendances de convergence des algorithmes au cours du processus d'optimisation - Scénario 3-

Hamam Boughrara issus des lâchers rendus par les quatre algorithmes dans le contexte du scénario 3. Il est avéré que les valeurs de la fonction objectif discutées auparavant, classent l'algorithme RGA en première position, GSA en deuxième, HPSOGWO et IGWO en troisième et quatrième positions respectivement. Le même ordre peut être dérivé des volumes déversés ( $V_{Dev}$ ). Ces derniers sont importants, car les apports prévus dans ce scénario expriment des années très humides dans lesquelles l'offre dépasse largement la demande. Par conséquent, un algorithme d'optimisation performant produira l'allocation entière de la demande et minimisera au maximum ces deux indicateurs.

Dans le même volet, les fiabilités volumétrique et temporelle confirment les résultats précédents. Elles signalent des niveaux parfaits relatifs à l'algorithme RGA, et une fiabilité volumétrique très importante et temporelle moyenne relatives au GSA, où les lâchers de ce dernier ne couvrent totalement que 63 mois (soit 52.50%) de la durée sujette à l'optimisation. De plus, les lâchers calculés par l'algorithme HPSOGWO prouvent une fiabilité volumétrique moins conséquente par rapport aux premiers, mais supérieure à celle dérivée de résultats de l'IGWO. Cependant, ce dernier a enregistré une fiabilité temporelle de l'ordre de 20% (soit

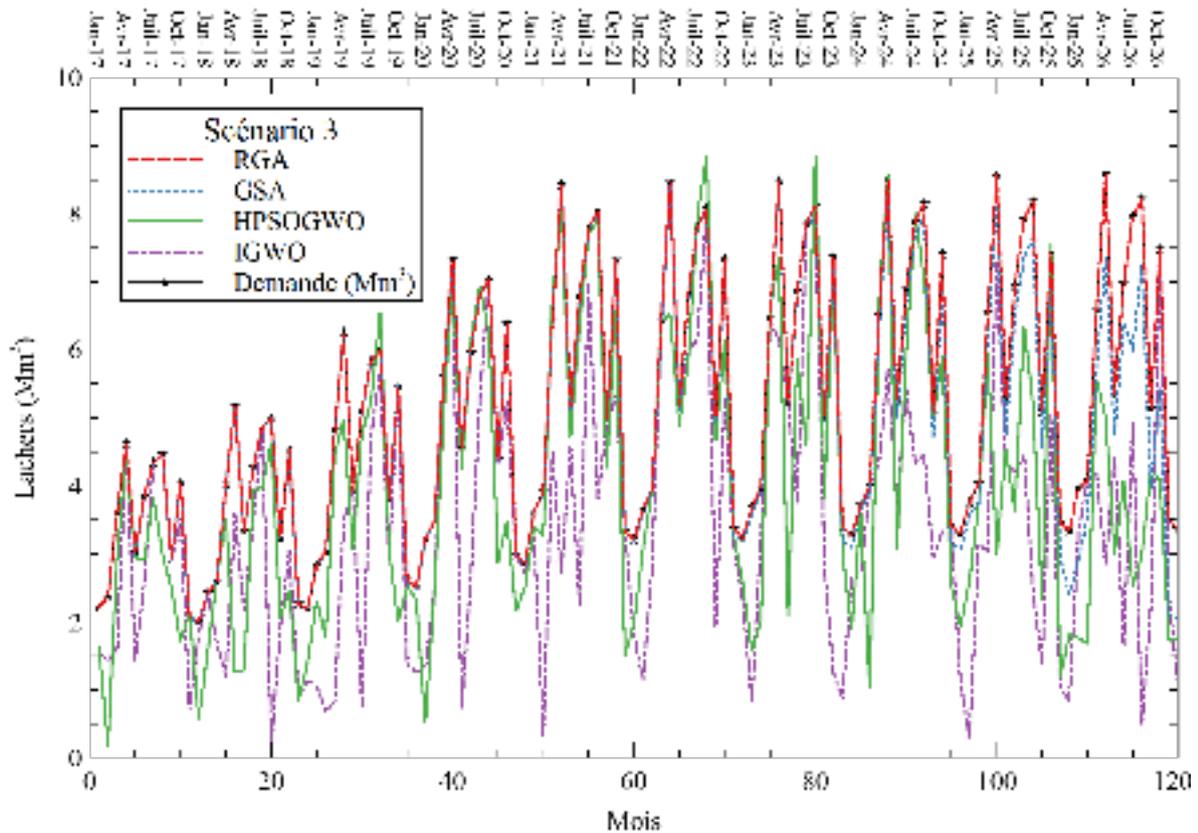


FIGURE 9.10 – Comparaison des lâchers mensuels optimisés par les différents algorithmes avec la demande en eau mensuelle ciblée - Scénario 3 -

une satisfaction totale de la demande de 24 mois/120) contre 14.17% (satisfaction entière de 17 mois/120) liée à l’algorithme HPSOGWO.

Sous l’effet que les indices de performance sont tributaires de résultats d’optimisation, le RGA a démontré une résilience intégrale et par conséquent une vulnérabilité nulle (tableau 9.6). En second lieu on trouve le GSA, dont sa distribution temporelle des lâchers a prouvé une résilience de l’ordre de 21.05%, accompagnée d’une vulnérabilité très faible ( $\sigma = 4.61\%$  correspondant à  $\sigma = 0.24Mm^3/mois$  de déficit). En troisième lieu l’algorithme HPSOGWO se positionne, dont sa distribution temporelle des lâchers a indiqué la résilience la plus faible ( $\gamma \cong 13.6\%$ ), jointe d’une vulnérabilité de l’ordre de 58.40% (vaut  $\sigma = 2.85Mm^3/mois$ ). En dernier lieu l’algorithme IGWO se situe, lequel ses lâchers optimisés conduiront à une résilience d’environ 19.8%, accompagnée de la vulnérabilité la plus intense dans ce scénario ( $\sigma = 65.92\% = 4.00Mm^3/mois$ ).

TABLE 9.6 – Indicateurs de performance relatifs au scénario 3

Scén <sup>a</sup>	Algorithme	$F_{obj}$ <sup>b</sup>	$V_{Dev}$ ( $Mm^3$ )	Fiabilités		Résilience	Vulnérabilité	
				$a_v$ (%)	$a_t$ (%)	$\gamma$ (%)	$\sigma$ (%)	$\sigma(Mm^3/mois)$
3	RGA	75.80	223.50	99.92	100	100	0.00	0.00
	GSA	81.48	244.44	95.89	52.50	21.05	4.61	0.24
	HPSOGWO	170.64	357.60	76.98	14.17	13.59	58.40	2.85
	IGWO	216.32	401.19	70.41	20.00	19.79	65.92	4.00

<sup>a</sup> Scénario d'apports.

<sup>b</sup> Best fitness.

La figure 9.11 indique pour les quatre algorithmes, la fréquence de périodes ( $N_{ps}$  en mois) satisfaites au minimum à différentes proportions ( $\tau_s$ ). Il est notable que la courbe ( $N_{ps} = f(\tau_s)$ ) dérivée des lâchers optimisés par l'algorithme RGA exprime la satisfaction entière de la demande cible pour la durée totale. Elle est suivie par celle du GSA, où les lâchers peuvent couvrir les demandes de  $N_{ps} = 120$  mois, mais à 61% uniquement dans certains. Le nombre de périodes diminue par la suite, où pour une allocation totale ( $\tau_s = 100\%$ ), seulement 63 mois (soit  $N_{ps} = 52.50\% = a_t$ ) sont satisfaits à ce taux. Les courbes attachées aux résultats produits par les algorithmes HPSOGWO et IGWO confirment les retombées précédentes et annoncent des déficits plus graves que ceux du GSA. L'allocation entière de la demande mensuelle visée n'a touché que 17 et 24 mois respectivement. Hormis, l'allocation sur la durée totale (120 mois) ne peut être qu'à 10% dans certains mois.

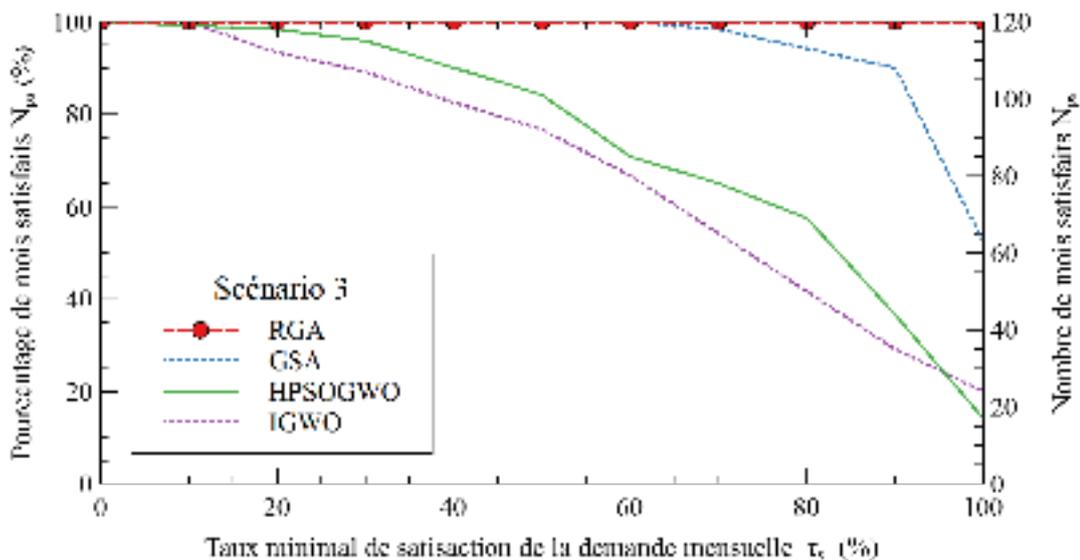


FIGURE 9.11 – Nombre de mois satisfaits en fonction de taux de satisfaction de la demande en eau mensuelle - Scénario 3-

L'estimation des surfaces au-dessous des courbes  $N_{ps} = f(\tau_s)$  est présentée dans le tableau 9.7. Il est clair que la surface la plus importante est celle relative à l'algorithme RGA, suivie par celle du GSA, puis par celle relative à l'HPSOGWO et finalement celle de l'IGWO. Ces résultats valident les aboutissements précédents et arrangent les algorithmes en termes de performance dans le présent scénario selon l'ordre déjà cité.

TABLE 9.7 – Surfaces au-dessous des courbes  $N_{ps} = f(\tau_s)$  relatives aux lâchers optimisés par les quatre algorithmes d'optimisation -Scénario 3-

Algorithme d'optimisation	RGA	GSA	HPSOGWO	IGWO
Surface $A_s$ au-dessous de la courbe $N_{ps} = f(\tau_s)$	120.00	115.05	90.55	83.20

La figure 13 (Annexes) montre l'évolution temporelle du stockage, déversement et des volumes évaporés relatifs au scénario 3. Il est clair que les apports importants prévus dans ce scénario accompagnés d'un remplissage important au début de la période d'optimisation ont induit le remplissage total du réservoir et par conséquent le déversement de surplus. Ce dernier est différent d'un algorithme à l'autre, car il est fortement lié à l'état du stockage et aux lâchers optimisés. En revanche, les pertes par évaporation issues des quatre algorithmes ressemblent l'une à l'autre, car elles sont substituées des surfaces du plan d'eau proches correspondant à la surface maximale de la retenue.

### 9.5.3.2 Synthèse des principaux résultats

Le perfectionnement de l'optimisation de la gestion du réservoir Hammam Boughrara en considérant les apports prévus dans le scénario 3 permet de synthétiser les points suivants :

- Malgré que les apports prévus et le stockage disponible dans le réservoir admettent l'allocation entière de la demande en eau en aval, l'algorithme RGA est le seul qui a généré les lâchers optimaux dans ce scénario. Les autres algorithmes (GSA, HPSOGWO et IGWO) ont été piégés dans des minima locaux de la fonction objectif et ont échoué d'atteindre l'optimal (minimum) global et la par conséquent l'insatisfaction des objectifs cibles.
- Le scénario a marqué la dérogation de la contrainte liée au dépassement des demandes en eau par l'algorithme HPSOGWO. Cela a induit une pénalisation encore plus des solutions générés par cet algorithme.
- D'une manière générale, les résultats d'optimisation associés au scénario 3 entraînent le classement des algorithmes de plus au moins performant dans l'ordre suivant : 1) RGA, 2) GSA, 3) HPSOGWO, 4) IGWO.
- La succession des années humides permettra au gestionnaire du réservoir d'al-

louer les quantités de l'eau exigées par les usagers (domestiques, industriels et agricoles) sans recours aux restrictions.

## **9.6 Convergence, stabilité et temps d'exécution**

La convergence d'un algorithme d'optimisation se définit simplement par sa capacité d'atteindre la solution optimale globale (Talbi 2009) ou des solutions suffisamment bonnes (Gutjahr 2009) d'un problème d'optimisation. Cependant, sa stabilité est peut-être mesurée sur différentes perspectives (Gonzalez 2007), néanmoins celle retenue dans cette analyse est exprimée par l'habilité de reproduire les mêmes solutions ou d'autres proches après chaque essai "lancement ou exécution" en gardant le même paramétrage de l'algorithme. La mesure de stabilité est réalisée dans ce qui suit à la base de l'écart-type et du coefficient de variation appliqués sur les valeurs finales de la fonction objectif "best fitness" de cinq (05) essais. Ceci permet l'évaluation des changements des solutions optimales d'un essai à l'autre d'une manière indirecte.

Les tableaux 14 et 15 (Annexes) montrent les résultats des essais appliqués sur les quatre algorithmes en termes de valeur de la fonction objectif ( $F_{obj}$  = "best fitness") correspondant à la meilleure solution (distribution des lâchers  $R_t$ ) achevée à la fin des itérations ( $k = K_{max} = 10^4$ ), et en termes du temps d'exécution.

L'analyse des résultats relatifs au scénario 1 indique la convergence des algorithmes RGA, GSA et HPSOGWO vers la même solution près, c-à-d., des distributions temporelles des lâchers relativement identiques. Néanmoins, cette solution n'est pas l'optimale globale, car les trois algorithmes n'ont pas produit des lâchers couvrant entièrement la demande en eau pendant les périodes de déversement. Mais en général, elle reste une solution de bonne qualité, car sa valeur de la fonction objectif ( $F_{obj}$ ) "ou son fitness" reste relativement faible malgré les pénalisations appliquées à l'effet de dépassements de la capacité maximale du réservoir et l'atteint du volume mort en plusieurs reprises. Concernant l'algorithme IGWO, ses résultats restent de mauvaise qualité à la fin des cinq essais à cause de la stagnation dans des minima locaux de la fonction objectif.

En termes de stabilité des solutions et par conséquence des algorithmes, le GSA a prouvé l'écart-type et le coefficient de variation les plus faibles, ce qui signifie la stabilité la plus pertinente. Il est suivi par les algorithmes HPSOGWO et RGA respectivement dont la stabilité reste très significative, affirmée par les faibles valeurs des mêmes critères. Finalement, l'IGWO a indiqué un écart-type et un coefficient de variation défavorables par rapport aux autres, donc une stabilité relativement basse.

En considérant les résultats du scénario 2, la même constatation peut être relevée concernant la convergence simultanée des algorithmes RGA, GSA et HPSOGWO qui ont trouvé pratiquement la même solution. De plus, cette dernière est de très bonne qualité et peut constituer la solution optimale globale du problème, car la valeur  $F_{obj}$  "fitness" correspondante a demeuré faible malgré les pénalités appliquées à cause de l'affrontement du volume mort du réservoir Hammam Boughrara. En revanche, les résultats relatifs à l'algorithme IGWO marquent sa stagnation encore dans des minimas locaux en produisant des solutions moins fiables par rapport aux autres.

L'examen des résultats relatifs à l'écart-type et au coefficient de variation confirme les trouvés précédents, sauf que l'algorithme HPSOGWO est déclassé en troisième position après le GSA et le RGA, mais reste meilleur que l'IGWO classé en dernier. Ces résultats signalent l'instabilité occasionnelle de l'algorithme HPSOGWO et confirment les conclusions annoncées dans (Dahmani and Yebdri 2020, Annexes). De plus, ils réaffirment l'instabilité de l'algorithme IGWO.

L'inspection des essais d'optimisation relatifs au scénario 3 proclame la supériorité des résultats trouvés par l'algorithme RGA, dont il a rapproché l'optimal global du problème représenté par le minimal global de la fonction objectif. Encore, le RGA a indiqué la stabilité la plus significative vu les faibles valeurs de l'écart-type et du coefficient de variation achevées. En deuxième position le GSA se situe sur les deux volets convergence et stabilité. Le GSA a abouti des solutions moins fiables par rapport au précédent (RGA) à la fin des essais, ce qui signale son piègeage dans des minimas locaux (pas loin du minimal global) et par conséquence une convergence malle par rapport à l'algorithme RGA. La même constatation est peut-être prolongée à sa stabilité, où l'écart-type et le coefficient de variation sont dégradés par rapport aux résultats précédents (liés aux autres scénarios) d'une part et par rapport à ceux de l'RGA d'autre part.

En considérant les résultats trouvés par le HPSOGWO. Cet algorithme a prouvé une qualité de solutions défectueuse due au retombement dans des minimas locaux. De plus, sa stabilité est dégradée par rapport aux algorithmes RGA et GSA, où l'écart-type et le coefficient de variation sont moins conséquents. En dernier lieu l'algorithme IGWO se classe, dont il a annoncé la convergence et la stabilité des solutions les plus dévalorisées.

L'expertise du temps d'exécution consommé par les quatre algorithmes indique que l'IGWO a dévoilé le temps le plus faible (environ  $13^m40^s$  en moyen), secondé par le HPSOGWO (environ  $15^m03^s$  en moyen), puis le RGA (en moyen  $31^m23^s$ ) qui a nécessité le double par rapport aux premiers (IGWO et HPSOGWO), et finalement le GSA (en moyen  $1^h2^m16^s$  environ). Ce dernier a exigé un temps d'exécution 4 fois plus que les métaheuristiques IGWO et HPSOGWO et 2 fois plus que le RGA. Le temps d'exécution est proportionnel au nombre

d'opérations effectuées par l'algorithme d'optimisation. Conformément à la section (2.6.2.1, chapitre 2), le GSA nécessite un nombre d'opérations mathématiques important par rapport aux autres algorithmes. Sans oublier que la performance de la machine sur laquelle les exécutions sont effectués, impacte le temps requis par les quatre algorithmes. Notons que cette étude comparative a été réalisée sous un ordinateur équipé d'un processeur Intel Core i3-4005U (1.7GHz, 3MB L3 cache), mémoire aléatoire DDR3 4GB, et d'un système d'exploitation Windows 7-32Bit.

Finalement, cette analyse confirme les constatations précédentes et classe les métaheuristiques d'optimisation en termes de performance dans l'ordre suivant :

1. Real-Coded Genetic Algorithm "RGA" ;
2. Gravitational Search Algorithm "GSA" ;
3. Hybrid algorithm of Particle Swarm Optimization and Grey Wolf Optimizer "HP-SOGWO" ;
4. Improved Grey Wolf Optimizer "IGWO".

Néanmoins, la confrontation des résultats trouvés dans cette étude avec ceux annoncés dans (Dahmani and Yebdri 2020, Annexes) revendique la faveur des métaheuristiques GSA et HPSOGWO lorsque le nombre de variables à déterminer est faible ou moyen, où les deux algorithmes ont prouvé une performance significative de point de vue qualité de solutions, convergence et stabilité. Dans le cas contraire, l'algorithme RGA peut constituer un bon alternatif. Aussi, il est trouvé que l'algorithme IGWO n'est pas adéquat à ce genre de problème dans les conditions de paramétrage utilisé dans cette étude. Cependant, une étude de sensibilité de l'algorithme est nécessaire pour confirmer cette constatation ou affirmer d'autres.

## **9.7 Conclusion**

À travers ce chapitre, l'optimisation de fonctionnement du réservoir Hammam Boughrara (W. Tlemcen) est abordée via trois scénarios d'apports couvrant la période du janvier 2017 au décembre 2026 (soit 120 mois). Le premier représente une période de sécheresse temporisée sur une longue durée, dans laquelle les apports sont inférieurs à la demande en eau des usagers. Le deuxième présume des apports produisant des volumes de remplissage approximatifs à ceux induits par les apports moyens. Le troisième scénario suppose un enchaînement des années humides caractérisées par des apports forts dépassant la consommation visée. Le recours aux scénarios sur entrées (apports) dans l'approche d'optimisation adoptée (ISO), est né des intenses incertitudes dans la matière de prévision, particulièrement dans les régions semi-arides où la modélisation fidèle des phénomènes hydrologiques est impossible. De l'autre côté,

le mode de consommation (modélisation de l'usage de l'eau) est gardé invariant d'un scénario à l'autre. Il est basé sur les objectifs tracés par les autorités compétentes ces dernières années. Cette modélisation des apports et des demandes de la ressource en eau n'abrège pas la généralisation du modèle d'optimisation discuté dans cette étude, de fait que les entrées peuvent être changées par d'autres et la qualité de sorties du modèle reste toujours mesurable.

Les quatre algorithmes RGA, GSA, HPSOGWO et IGWO sont confrontés entre eux dans le but d'identifier leurs capacités à résoudre ce type de problèmes d'optimisation dont le nombre d'inconnus est élevé (120 variables dans le présent cas), et dont les entrées (apports et consommation en eau) sont très variables. La fonction objectif choisie vise à maximiser la satisfaction de la demande en eau des usagers (par minimisation des déficits), en tenant compte les différentes contraintes liées au fonctionnement du réservoir. Les variables de décision retenues sont les lâchers mensuels ( $R_t$ ,  $t \in [1, 120]$ ). Par conséquent, les politiques de répartition temporelle de la ressource en eau soutirée du réservoir sont ciblées par l'optimisation. À fin de réaliser une comparaison plus au moins judicieuse, les quatre algorithmes ont été lancé cinq (05) fois individuellement pour faire 1.5 Million ( $1.5 \times 10^6$ ) d'évaluations de solutions dans chaque scénario, dans l'expectative d'atteindre les solutions optimales globales du problème et par conséquent les politiques optimales de répartition temporelle de la ressource en eau du réservoir.

L'évaluation de la qualité des résultats est référée à plusieurs critères tels que la convergence (indiquée par la valeur de la fonction objectif ( $F_{obj}$ ) à la fin du traitement itératif), les volumes déversés ( $V_{Dev}$ ), la fiabilité volumétrique ( $a_v$ ) et temporelle ( $a_t$ ), la résilience ( $\gamma$ ), ... etc. Cela a permis de juger les résultats sur plusieurs angles.

Les principales conclusions issues de cette étude peuvent être résumées dans les points suivants :

- Le classement des algorithmes testés en termes de performance dans la résolution de tels problèmes dans l'ordre suivant : RGA, GSA et HPSOGWO, puis IGWO. Néanmoins, ce dernier algorithme est trouvé carrément inapproprié dans le contexte de paramétrage utilisé.
- Le scénario 1 a relevé l'incapacité des quatre algorithmes à allouer la quantité d'eau visée pendant les périodes de déversement. Ce qui signifie que les distributions temporelles des lâchers dans ce cas ne sont pas optimales certaines, mais proches à l'optimale. Donc, les métaheuristiques ne garantissent pas la solution optimale globale d'un problème d'optimisation et restent des algorithmes du calcul approché.
- La comparaison des résultats obtenus dans cette section avec ceux publiés dans notre manuscrit (Dahmani and Yebdri 2020) montre l'impacte de nombre de va-

riables de décision sur la qualité des résultats. La comparaison montre clairement que les algorithmes GSA et HPSOGWO sont préférables le cas d'optimisation sur une durée relativement courte (nombre faible ou relativement moyen de variables à déterminer). Dans le cas contraire, l'algorithme RGA peut constituer un bon choix.

- Le temps d'exécution faible prouvé par les métaheuristiques HPSOGWO et IGWO face au autres peut encourager l'amélioration de leurs performances (convergence et stabilité) notamment par couplage ou hybridation avec d'autres algorithmes d'optimisation.
- Les scénarios 1 et 2 basés sur des apports faibles et moyens prévus (respectivement) revendiquent l'incapacité du réservoir Hammam Boughrara à répondre aux objectifs tracés par les autorités concernées dans les conditions climatiques exprimées par ces apports. Le réservoir a atteint son volume mort plusieurs fois durant la période d'optimisation envisagée sans satisfaire entièrement la demande ciblée.
- Les différences entre les stratégies de répartition temporelle de la ressource en eau (distribution des lâchers) et les trajectoires des réserves du barrage Hammam Boughrara obtenues par les différents scénarios d'apports indiquent à quel niveau l'optimisation de la gestion d'un réservoir dans l'incertain est sensible aux sorties des modèles de prévision des apports. La même constatation est prolongée aux sorties du modèle de prévision de la consommation de la ressource en eau ainsi aux considérations prises dans ce contexte. Néanmoins, le modèle d'optimisation discuté dans cette thèse reste souple devant les éventuels changements dans les apports à l'entrée et/ou les demandes à la sortie du réservoir. De plus, il reste applicable et valable pour d'autres réservoirs.
- L'appui sur un nombre minime d'indicateurs "ou critères" de performance peut induire à une estimation illusoire de la performance globale d'un système de gestion. L'évaluation de cette dernière exige l'incorporation de plusieurs critères permettant sa mesure sur différentes facettes (volumes alloués, fréquence d'allocation, taux de satisfactions, ampleurs de vulnérabilité, résilience du système, ... etc).
- L'émergence de l'indicateur exprimé par la surface au-dessous de la courbe décrivant la relation entre les deux grandeurs, nombre de périodes allouées et taux minimal d'allocation d'une répartition temporelle donnée de la ressource en eau, peut constituer une autre facette de mesure de l'efficacité de celle-ci.

Finalement, cette étude doit être complétée par l'optimisation de la gestion du réservoir Hammam Boughrara sous d'autres scénarios d'apports liquides. Ceci permettra de générer un ensemble de politiques de gestion du réservoir, sur lesquelles une analyse par régression multiple, réseaux de neurones artificiels, systèmes flous ou autre modèle de régression, pourra être appliquée pour dériver les règles optimales de gestion. Une autre étude visant la règle de

répartition de la ressource en eau entre usagers est nécessaire. Celle-ci permettra de déterminer les différentes restrictions à appliquer sur l'allocation de l'eau aux usagers sous différents contextes socio-économiques et climatiques.

# Chapitre 10

## Présentation de l'application RMOSS

### 10.1 Introduction

Au cours de la préparation de cette thèse, nous avons implémenté les différentes tâches requises à l'élaboration et à la mise en exécution du modèle d'optimisation de la gestion d'un réservoir dans la même application. Cette dernière est nommée "Logiciel d'optimisation et de simulation de la gestion des réservoirs", en anglais "Reservoir Management Optimization and Simulation Software", abrégé "RMOSS". À la limite de connaissances de l'auteur et selon (Rani and Moreira 2010), le domaine de l'optimisation de la gestion opérationnelle des réservoirs marque l'absence des logiciels utilisant les méthodes de calcul intelligent (Computational intelligence), telles que les réseaux de neurones artificiels, les systèmes flous et les métaheuristiques. Ceci nous a encouragé à prendre l'initiative dans la matière en mettant le code de l'application en source ouverte (open source) aussi.

Dans ce qui suit de ce chapitre, une présentation grossière et succincte du logiciel "RMOSS" développé à travers cette thèse, notamment son architecture dans le but de projeter la lumière sur les efforts exercés à fin de ressortir un outil informatique contribuant à la résolution de ce genre de problèmes d'optimisation.

### 10.2 Outils de développement utilisés

L'application en question est développée en moyen des langages de programmation VisualBasic.NET (abrégé VB.NET) et C#.NET (prononcé c sharp dot net ou c sharp tout court). Ces deux langages ciblent le développement d'une grande variété de types d'applications (bureautique, web, mobile, ... etc) fonctionnant sous les plateformes informatiques : Microsoft.NET

(connue aussi sous le nom : .NET "dot net" framework) (Microsoft 2020a), .NET Core (version modulaire, multiplateforme et open source de .NET), Mono (Mono 2020) et Xamarin (Xamarin 2020).

Les deux langages VB.NET et C# sont à usage général "general purpose programming languages" de type orienté objet, structuré et impératif. Ils sont très compatibles entre eux, car le code rédigé par l'un d'eux (ou par autre langage ciblant la plateforme .NET comme C++ CLR, F#, IronPython, ..., etc.) se transforme au même langage dit intermédiaire (Intermediate Language "IL" ou Microsoft intermediate language "MSIL") avant d'être compilé en langage machine (Leblanc 2006). De plus, les deux langages disposent un compilateur commun connu sous le nom "Roslyn Compiler". Ce dernier est disponible librement "en open-source" dans le site d'hébergement et de gestion de développement de logiciels (<https://github.com/>) (Roslyn-Compiler 2020).

Le logiciel éditeur ou l'environnement de développement intégré (Integrated Development Environment "IDE") utilisé dans l'élaboration de cette application est Microsoft Visual Studio (édition communauté "Community Edition", 2010, 2015, 2017, puis 2019). Cette version de l'IDE est offerte gratuitement par la société américaine Microsoft aux étudiants, enseignants, chercheurs, petites entreprises (dont le nombre d'employés est inférieur ou égale à 5) et aux développeurs de source libre "open source" (Microsoft 2020b).

### **10.3 Présentation de l'application**

L'application "RMOSS" est de type bureautique "windows desktop application" dotée d'une fenêtre unique hébergeant d'autres (figure 10.1). Elle fonctionne sous machines équipées de système d'exploitation Microsoft Windows avec la plateforme "Microsoft .NET framework version 4.0" ou plus. Elle vise d'accomplir les tâches principales suivantes dans sa version actuelle :

1. Prévission des apports d'un réservoir (ou les débits d'un cours d'eau) au moyen des réseaux de neurones artificiels conventionnels et évolutionnaires.
2. Optimisation de la gestion opérationnelle d'un réservoir à but unique ou à buts multiples, dont ces derniers peuvent être regroupés dans une seule fonction objectif.

L'application dispose une barre de menus principale permettant de naviguer entre les fenêtres pour réaliser les différentes opérations supportées. Les projets d'optimisation créés peuvent être enregistrés dans des fichiers textuels de format XML (Extensible Markup Lan-

guage) modifiables par les éditeurs de textes. Aussi, l'application dispose des sous-programmes permettant l'échange de données (import/export) ainsi les outils d'affichage tabulaire et graphique facilitant la lecture et la manipulation des entrées et résultats.

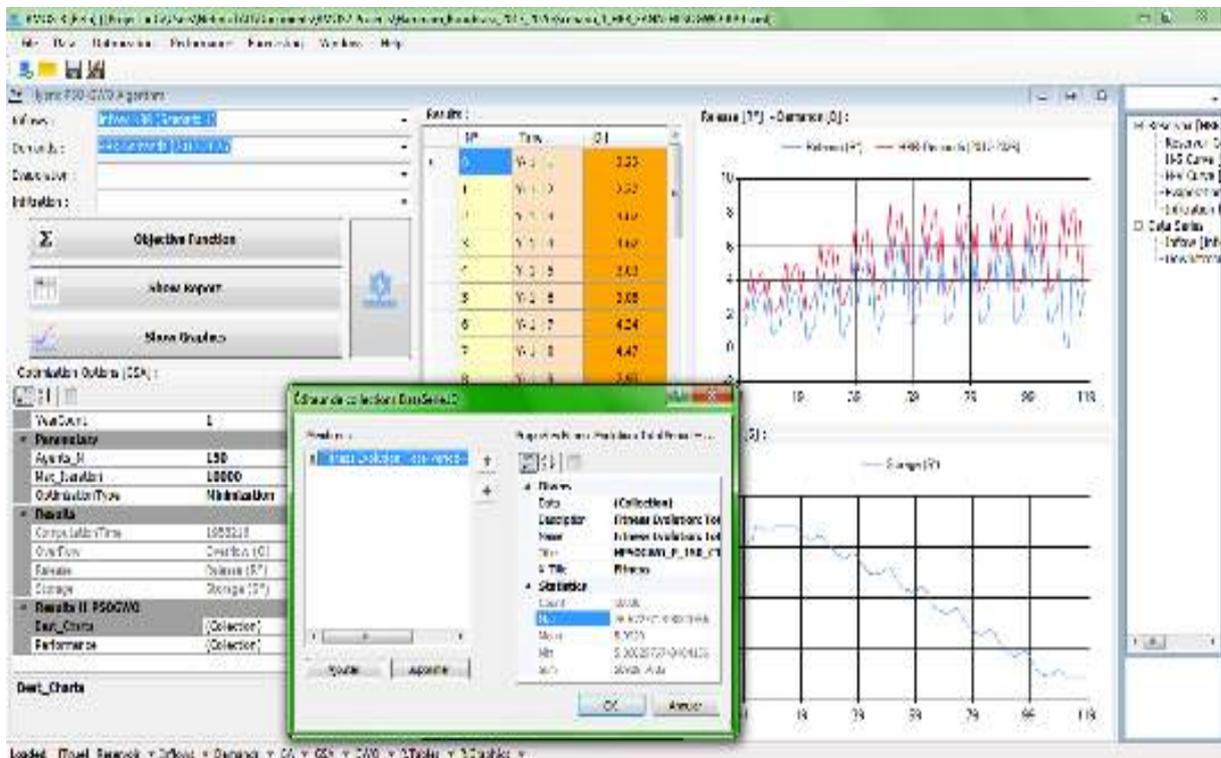


FIGURE 10.1 – L'application "RMOSS" dans l'action d'optimisation de la gestion du réservoir Hammam Bougrara par l'algorithme HPSOGWO

L'application "RMOSS" est développée pour des finalités académiques et elle est disponible en source libre dans le site ([github.com](https://github.com/SaadDAHMANI/RMOSS)) sous le lien (<https://github.com/SaadDAHMANI/RMOSS>), pour probables consultations, vérifications, modifications ou développement.

## 10.4 Architecture de l'application

L'application est réalisée sous l'approche multicouches, où trois (03) couches sont adoptées (figure 10.2).

### 10.4.1 Couche d'interface d'utilisateur

La première couche de l'application est celle de l'interface graphique (interface utilisateur ou Graphic User Interface "GUI" en anglais). Elle est développée au moyen du langage VB.NET et se génère sous forme d'un exécutable ".exe" fonctionnant sur la plateforme ".NET

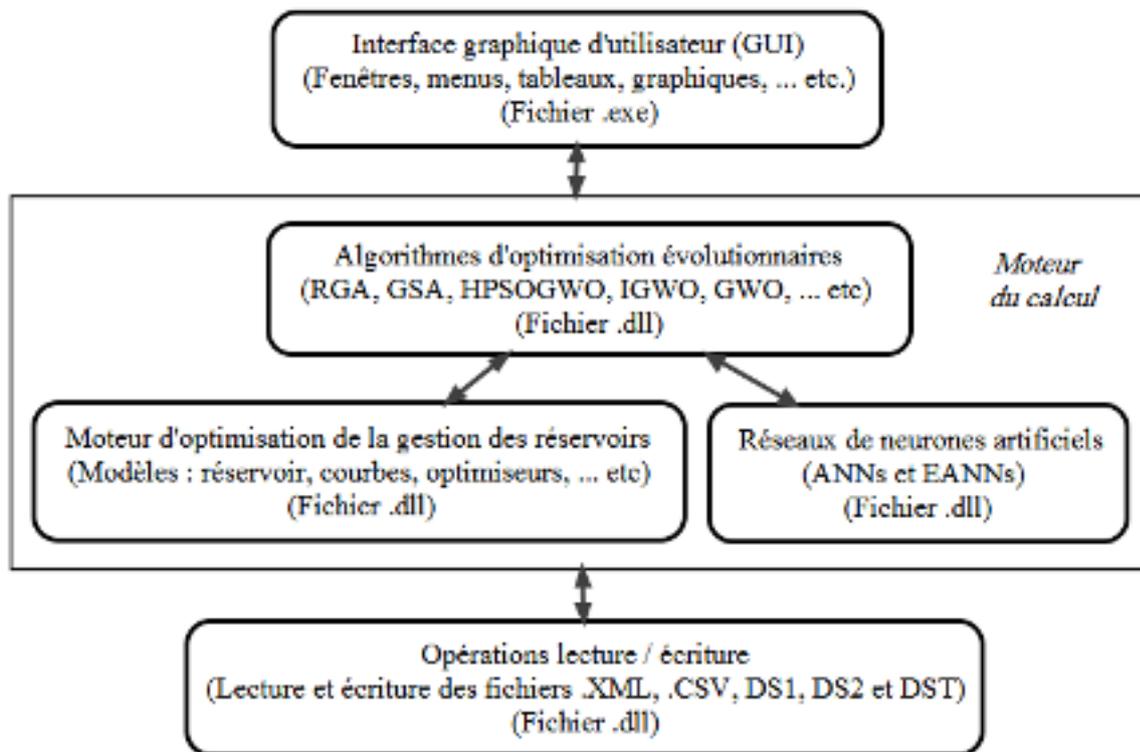


FIGURE 10.2 – Architecture de l'application "RMOSS"

framework version 4.0" ou plus. Cette couche comporte l'ensemble des fenêtres permettant de communiquer avec l'utilisateur en lui donnant la possibilité de saisir, modifier, naviguer entre les différentes fenêtres, lancer les calculs, ... etc. Elle permet l'affichage des résultats sous forme de textes, tableaux et graphiques, ... etc.

#### 10.4.2 Couche moteur du calcul

Elle est responsable sur les différentes opérations mathématiques telles que l'optimisation, l'apprentissage et tests des réseaux de neurones artificiels conventionnels et évolutionnaires, la prévision, ... etc. Elle comporte aussi la modélisation orienté-objet des différents composants du modèle d'optimisation (réservoir, courbes hauteur-surface et hauteur-volume, ... etc). Elle est organisée sous forme de bibliothèques de lien dynamique (Dynamic Link Library, "DLL"). Ces dernières sont des fichiers d'extension ".dll", transportables et utilisables d'une application à l'autre ou d'un langage de programmation à l'autre.

### **10.4.2.1 Bibliothèque des algorithmes évolutionnaires**

Elle réunit l'ensemble des algorithmes métaheuristiques discutés dans cette thèse (et autres) utilisés pour accomplir trois (03) opérations principales, à savoir :

1. utilisés comme moteurs de calcul dans le modèle d'optimisation de la gestion des réservoirs déjà présenté ;
2. l'entraînement des réseaux de neurones artificiels conventionnels "ANNs" ;
3. la recherche des réseaux de neurones artificiels évolutionnaires "EANNs" et leurs entraînements.

Les algorithmes évolutionnaires sont réimplémentés depuis leurs codes sources originellement publiés sous les langages de programmation Matlab, C# ou Python selon le cas, et à la base des articles publiés par leurs auteurs. Les algorithmes pris en charge sont : RGA, GSA, hybrid Particle Swarm Optimization and Gravitational Search Algorithm "PSO-GSA" (Mirjalili et al. 2012), Grey Wolf Optimizer "GWO" (Mirjalili et al. 2014), HPSOGWO, IGWO, Modified Grey Wolf Optimizer "mGWO"(Mittal et al. 2016) et AdaptiveGWO (Dudani and Chudasama 2016). Les algorithmes implémentés supportent la minimisation et la maximisation des fonctions objectifs.

Cette bibliothèque de lien dynamique a été implémentée en langage VB.NET et publiée librement "en open source" dans le site "[github.com](https://github.com/SaadDAHMANI/EOAlgos)" sous le lien : (<https://github.com/SaadDAHMANI/EOAlgos>) pour éventuelles vérifications, réutilisations ou améliorations.

### **10.4.2.2 Bibliothèque des réseaux de neurones artificiels**

Cette bibliothèque de lien dynamique "DLL" regroupe l'ensemble de programmes permettant l'entraînement des réseaux de neurones artificiels conventionnels "ANNs" par les métaheuristiques et la recherche des réseaux de neurones artificiels évolutionnaires "EANNs" destinés à la prévision des apports (figure 10.2). Elle est développée en langage de programmation C# et reliée à la bibliothèque des algorithmes évolutionnaires déjà présentée. Notons que l'implémentation informatique des réseaux de neurones artificiels et les méthodes d'apprentissage classiques (rétropropagation du gradient de l'erreur "BP" et Levenberg-Marquardt "LM") n'est pas effectuée par l'auteur de la thèse. Ces composants sont importés et reliés avec cette bibliothèque depuis la plateforme libre d'apprentissage machine appelée "Accord.NET Framework (Souza et al. 2014)".

### **10.4.2.3 Moteur d'optimisation de la gestion opérationnelle des réservoirs**

Est une bibliothèque de lien dynamique développée en langage de programmation VB.NET. Elle encapsule les programmes modélisant (de point de vue informatique) les composants du modèle d'optimisation tels que : réservoir, courbes hauteurs - surfaces, hauteurs - volumes, hauteurs d'évaporation, ..., et les sous programmes d'optimisation de la gestion des réservoirs basés sur les algorithmes évolutionnaires, ainsi l'implémentation de la programmation dynamique stochastique "SDP" d'optimisation. Notons ici que les sous programmes d'optimisation sont implémentés indépendamment à la fonction objectif, ce qui permet de la modifier ou changer par autre sans nécessité de reprogrammation.

### **10.4.3 Couche de sauvegarde et récupération des données**

Elle inclut l'ensemble des sous programmes permettant l'enregistrement des données dans le disque dur de la machine et leurs récupérations par la suite (figure 10.2). Elle est organisée sous forme d'une bibliothèque "DLL" développée en langage C#. Le premier type de fichiers utilisés suit le format XML. Ces fichiers sont utilisés pour le stockage et la récupération des données sur le réservoir et ses caractéristiques (nom, volume mort, capacité de stockage maximale, ... etc), les apports, la consommation, ... etc. Ces informations sont regroupées dans un "Projet RMOSS" et enregistrées dans un seul fichier XML. Le deuxième type de fichiers est "CSV" (abréviation de "Comma-Separated Values"). Ce type est compatible avec les logiciels tableurs tels que Microsoft Excel, OpenOffice Calc, Sheets, ... etc. Il est employé pour l'importation des données et l'exportation des résultats. Les derniers types sont des fichiers personnalisés (créés par le développeur de l'application) de format binaire et d'extensions ".DS1, .DS2 ou .DST". Ils sont utilisés pour stocker et récupérer les séries de données de longueurs variables qui peuvent être dotées de 1, 2 ou N dimensions (ou colonnes).

## **10.5 Conclusion**

Ce chapitre a été consacré à la présentation brève de l'application RMOSS "Reservoir Management Optimization and Simulation Software" développée par l'auteur de la thèse dans le cadre de la préparation de celle-ci. Cette application peut constituer un noyau de développement d'un logiciel plus complet dans la discipline en question, notamment avec la mise en disposition libre de son code source pour téléchargement, vérification, modification et développement via le site d'hébergement des logiciels "*Github.com*" sous le lien indiqué dans la section 10.3.

L'application "RMOSS" est conçu dans sa version actuelle pour effectuer deux tâches

principales, 1) la prévision des apports par réseaux de neurones artificiels, et 2) l'optimisation de la gestion opérationnelle des réservoirs par la génération des politiques optimales de gestion. Dans le but d'atteindre ces objectifs, l'application implémente un ensemble d'algorithmes utilisés pour réaliser ces tâches. De plus, elle est équipée des procédures permettant le sauvegarde et la récupération des données, ainsi l'affichage des résultats sous forme tabulaire et graphique.

Malgré que cette application est utilisée dans cette étude sur le cas du réservoir Hammam Boughrara, mais elle reste générique et applicable sur d'autres cas obéant au modèle général d'optimisation déjà présenté. Rappelons aussi que la fonction objectif constituant le coeur d'un problème d'optimisation est peut-être changée ou modifiée indépendamment sans toucher les autres composants de l'application.

Les perspectives de développement de cette application touchent principalement :

- L'intégration de diverses techniques de régressions permettant la substitution des règles optimales de gestion ;
- L'implémentation des autres algorithmes évolutionnaires performantes dans la matière ;
- L'élargissement de sa capacité du calcul à l'optimisation multiobjectifs, c-à-d., la minimisation ou la maximisation de plusieurs fonctions objectifs à la fois ;
- L'intégration des autres modèles de prévision tels que ARMA, ARIMA, ..., ainsi l'incorporation des modèles de simulation de fonctionnement des réservoirs qui reste une tâche essentielle ;
- La synchronisation des processus de traitement avec le processus principal de l'application. Cela permet de lancer plusieurs opérations de calculs à la fois, telles que l'entraînement de plus d'un réseau de neurones à la fois, l'optimisation par plus d'un algorithme, ... etc ;
- L'incorporation des autres approches d'optimisation stochastique de la gestion opérationnelle des réservoirs et systèmes de réservoirs notamment la P-S-O.

Finalement, cette application est ouverte librement (en open source) dans l'espérance de constituer une pierre pour bâtir des logiciels plus complets et d'aider les chercheurs ainsi les gestionnaires dans la discipline de management des ressources en eau.



## Conclusion générale

Les barrages réservoirs représentent actuellement la pierre angulaire dans la gestion des ressources eau à travers le monde. Ils assurent d'environ un quart (1/4) des besoins mondiaux en eau de différents secteurs. Sans oublier leurs rôles et importance dans la régulation des cours d'eau permettant la protection des milliers de vies à travers le monde entier.

Dans ce travail, nous avons traité partiellement la problématique de l'optimisation de la gestion opérationnelle d'un réservoir à objectifs multiples. Le fonctionnement de tel ouvrage se caractérise par beaucoup d'incertitudes notamment celles référées aux phénomènes hydrologiques, spécifiquement les apports liquides. L'approche d'optimisation stochastique implicite (ISO) de la gestion opérationnelle des réservoirs est adoptée dans ce travail dans la finalité de tenir en compte la stochasticité des apports. Notons que la problématique a été traitée aussi via l'approche d'optimisation explicite (ESO) (par la méthode SDP) dans notre manuscrit (Dahmani et al. 2018)(Annexes).

Comme évoqué précédemment, nous avons concentré sur les deux premières phases de l'approche ISO, la génération stochastique des apports mensuels et la production des politiques de gestion optimales. Les principales conclusions qui peuvent être tirées à partir de ce travail se résument dans les points suivants.

### A. Prévision des apports mensuels du réservoir Hammam Boughrara

Cette partie s'inscrit sous la problématique de prévision des apports des réservoirs dans les zones caractérisées par un climat semi-aride. Dans ce contexte, les réseaux de neurones artificiels sont utilisés pour traiter six (06) schémas de régression à la base de la série chronologique mensuelle des apports. Le pas mensuel est couramment choisi dans la planification de la gestion opérationnelle des réservoirs.

En premier lieu, les réseaux de neurones artificiels dotés d'une (01) et deux (02) couches cachées entraînés par les méthodes classiques (rétropropagation du gradient de l'erreur "BP" et Levenberg-Marquardt "LM") ont été employées. Les résultats ont montré des corrélations faibles, sauf pour quelques modèles composés de deux (02) couches cachées et entraînés par l'algorithme LM. Dans cette catégorie, le réseau de neurones "ANN-LM-3 (6, 15)", de six (06) et quinze (15) neurones dans la première et la deuxième couches cachées respectivement, a permis une régression  $Q_t = f(Q_{t-1}, Q_{t-2}, Q_{t-3})$  de qualité d'apprentissage ( $R = 82\%$ ,  $Nash = 0.67$ ) et de validation ( $R = 43.8\%$ ,  $Nash = 0.16$ ). Et d'une manière

générale, les RNAs à double couches cachées ont produits des résultats plus remarquables que ceux dotés d'une mono-couche cachée.

En deuxième lieu, les mêmes RNAs ont été entraînés séparément par les deux évolutionnaires RGA et HPSOGWO en profitant la capacité générique d'optimisation de ce type d'algorithmes. Le RGA a produit des résultats relativement meilleurs par rapport à HPSOGWO. Hormis, cet essai n'a pas permis d'améliorer la qualité des résultats en comparaison avec les méthodes classiques.

En dernier lieu, et vu la qualité des résultats aboutis par les deux essais précédents, nous avons constaté la limite du paramétrage manuel (le cas le plus courant) des NAs. Un effort a été fourni ici, afin de tester la possibilité d'exploiter les algorithmes métaheuristiques d'optimisation dans l'espérance de franchir la limite de ce mode de paramétrage. Dans ce cadre, un troisième essai a été élaboré en exploitant les deux métaheuristiques GSA et HPSOGWO pour chercher les meilleurs paramètres de structure des RNAs. Cette famille des réseaux de neurones est connue sous le nom "Réseaux de neurones artificiels évolutionnaires (EANNs)". Les deux algorithmes GSA et HPSOGWO ont été couplés séparément avec ceux classiques d'apprentissage (BP et LM). Les résultats ont montré des améliorations significatives dans la phase d'apprentissage au détriment de la phase de validation, en provoquant le phénomène de sur-apprentissage de la majorité des réseaux de neurones artificiels évalués. A titre d'exemple, l'EANN-HPSOGWO-BP-6(6,9) permettant la régression selon le modèle 6 ( $Q_t = f(Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}, Q_{t-6})$ ) a prouvé une corrélation  $R = 88.6\%$  et  $Nash \cong 0.77$  dans la phase d'apprentissage, contre  $R = 17.1\%$  et  $Nash \cong -0.6$  dans la phase de validation.

En guise de conclusion, la régression par pas mensuel via les réseaux de neurones artificiels conventionnels et évolutionnaires n'a produit des résultats satisfaisants. En effet, le recours à d'autres modèles peut être nécessaire.

## **B. Génération des politiques de gestion optimales du réservoir H.Boughrara**

La problématique de l'optimisation de la gestion opérationnelle d'un réservoir à objectifs multiples a été abordée dans ce travail, où l'approche ISO a été retenue. Un modèle d'optimisation qui tient compte l'ensemble de contraintes liées au fonctionnement de tel ouvrage a été utilisé. Il conduit à la génération des politiques d'exploitation optimisées, lesquelles servent à la confection des règles optimales de la gestion. Les algorithmes métaheuristiques tant que méthodes de calcul approché, ont été exploités ici pour résoudre cette problématique. Quatre algorithmes ont été testés et comparés, à savoir : 1) une version de l'algorithme génétique à codage réel "RGA", 2) l'algorithme de recherche gravitationnelle "GSA", 3) l'algorithme hybride

de l'essaim de particules et l'optimiseur de Loup Gris "HPSOGWO" et 4) l'optimiseur de Loup Gris amélioré "IGWO".

Les quatre algorithmes ont été utilisés séparément pour produire trois (03) politiques de gestion référées à trois scénarios d'apports prévus différents. Ces derniers couvrent la période du janvier 2017 au décembre 2026 (120 mois). Les scénarios d'apports sont considérés à côté d'un seul scénario de consommation, dans lequel l'usage de l'eau est présumé présente une variation mensuelle avec tendance d'évolution au cours du temps jusqu'à l'atteint de la vision d'allocation prévue par le gestionnaire.

Une seule fonction objectif non-linéaire a été optimisée par minimisation, dans laquelle les lâchers mensuels constituent les variables de décision. La fonction regroupe les buts d'exploitation du réservoir Hammam Boughrara (alimentation en eau potable, industrielle et irrigation). L'unique fonction a permis l'emploi des algorithmes d'optimisation mono-objectifs et d'avoir des structures de données et résultats plus simples. L'optimisation a ciblé la minimisation des défaillances du système de gestion, donc la maximisation de la satisfaction des demandes en eau des usagers en respect des contraintes de fonctionnement du réservoir. L'appréciation des résultats ressortis par les quatre métaheuristiques a été fondé sur plusieurs critères et indices, y compris les volumes déversés, fiabilité volumétrique et temporelle, résilience, vulnérabilité, trajectoire de réserve, convergence, stabilité des solutions, ... etc.

La considération de l'ensemble des résultats indiqués dans ce manuscrit avec ceux présentés dans (Dahmani and Yebdri 2020, Annexes) révèle la supériorité de la version utilisée de l'algorithme génétique à codage réel "RGA" face au autres. Le RGA a surmonté le piègeage dans les minima locaux dans la majorité des cas en convergeant vers des solutions de bonne qualité voir meilleure dans la plupart. La précision achevée est largement suffisante pour ce type de problèmes vu les incertitudes qui les caractérisent. Encore, le RGA a prouvé une stabilité pertinente avec un temps d'exécution moyen. Notons que cette version est simple, car elle implémente des mécanismes simplifiés de sélection, croisement et mutation.

L'algorithme de recherche gravitationnelle "GSA" se classe en deuxième position. Il a abouti à des solutions de haute qualité avec une stabilité très élevée lorsque le nombre de variables de décisions était moyen. Hormis, le GSA a éprouvé la stagnation dans des minima locaux dans certain cas en manquant les solutions optimales globales. L'inconvénient majeur de cet algorithme réside dans le temps de traitement long engendré par le nombre d'opérations mathématiques exigées par son processus. Vu ces constatations, le GSA apparaît n'est pas préféré quand nombreux scénarios de longues durées sont à lancer.

L'algorithme HPSOGWO se répertorie en troisième rang après le RGA et GSA. Il a subi des piègeages dans les optimums locaux de la fonction objectif plus que GSA. Or, il a

prouvé une performance très proche et parfois comparable à ce dernier. À l'exception de son instabilité conjoncturelle, le HPSOGWO est deux fois plus rapide que le RGA et quatre fois que le GSA. En considérant les résultats démontrés par cet algorithme, il peut être recommandé lorsque la taille de problème est moyenne ou réduite, et lorsque un nombre important d'exécution est à déclencher. Vu les avantages que présente le HPSOGWO, son amélioration ou hybridation avec une autre métaheuristique de recherche locale (ex., Local Search) peut remédier ses défauts et ressortir un algorithme plus robuste.

En dernière position l'IGWO se range. Cet algorithme a marqué la stagnation dans les optimums locaux dans tous les essais effectués sur les trois scénarios. L'IGWO est trouvé inadéquat à ce genre de problèmes dans le contexte de paramétrage utilisé. L'algorithme a évoqué le temps d'exécution le plus court parmi les métaheuristicues étudiées. En considérant les points favorables de cet algorithme et qui son partagés avec le HPSOGWO, son amélioration notamment par hybridation avec un autre de recherche locale est de grande opportunité. Notons que les algorithmes de recherche locale se caractérisent par une grande capacité d'échappement aux optimums locaux.

D'une manière générale, les résultats présentés dans ce manuscrit et ceux décrits dans (Dahmani and Yebdri 2020) confirment les théorèmes connus sous "*No free lunch theorems for optimization*" proposés par (Wolpert and Macready 1997).

Le fonctionnement probable du réservoir Hammam Boughrara dans le contexte des objectifs d'allocation nouvellement tracés, est examiné sous les trois scénarios en conséquence. Il est trouvé que l'enchaînement des années sèches ou d'apports proches à la moyenne ne permet pas de parvenir les objectifs de gestion et conduit à l'épuisement du réservoir. Des restrictions notamment sur les volumes alloués à l'irrigation pourront être nécessaires dans ces circonstances. Les restrictions influencent directement la surface irriguée du périmètre de Maghnia par réduction si les déficits ne peuvent être pas remboursés par d'autres sources.

L'examen de fonctionnement du réservoir à la base des scénarios d'apports différents montre la forte sensibilité de l'élaboration des politiques optimales de gestion aux sorties de l'opération de prévision des entrées du barrage. La sensibilité est peut être étirée à la prévision de régime de consommation de la ressource en eau aussi.

Une constatation principale tirée à partir de cette étude révèle l'insuffisance de recours à peu de critères de mesure de performance d'un système de gestion sous une politique donnée. Les critères quantitatifs (fiabilité, résilience et vulnérabilité) couramment abrégés par "critères FRV" doivent être complétés par d'autres en fonction des facettes de mesure voulues. La mesure de la fréquences de couverture de la demande ( $N_{ps}$ ) à différents seuils ( $\tau_s$ ) nous a permet de juger mieux la qualité des politiques de gestion déduites par les différents algorithmes.

Finalement, malgré que le modèle d'optimisation décrit dans cette étude a été opéré sur le réservoir Hammam Boughrara, il reste valable et applicable sur d'autres cas similaires.

### **C. Perspectives**

Chaque travail académique de ce genre, s'initialise par la volonté de répondre à une ou plusieurs questions au départ. Cependant, il se termine dans l'ordinaire par la pose de nouvelles questions et l'ouverture des nouvelles perspectives. À travers cette modeste étude, nous avons relevé quelques perspectives que nous estimons méritent d'être entamer à l'avenir.

Premièrement, d'autres modèles de génération stochastique des apports (tels que cités dans Jia and Culver 2006 ; Oliveira and Maia 2018, ... etc), doivent être utilisés et auscultés dans le contexte du climat semi-aride, dans la finalité de produire des entrées plus au moins fiables aux modèles d'optimisation.

Deuxièmement, la discipline de l'optimisation s'enrichit chaque année par des nouvelles techniques qui doivent être examinées dans le cadre de la problématique en question. Sans oublier l'amélioration de méthodes existantes, notamment celles qui présentent des avantages requis dans le domaine de l'ingénierie des ressources en eau. A titre d'exemple, l'hybridation entre les métaheuristiques de recherche globale telles utilisées dans cette étude avec celles de recherche locale ou avec des méthodes exactes représente une opportunité majeure de recherche.

Cette étude doit être complétée sans doute par la phase de génération des règles de gestion optimales. Dans ce volet, plusieurs techniques de régressions peuvent être exploitées, particulièrement celles référées à l'apprentissage machine ("Machine Learning", RNAs, SVR, ... etc). L'usage de ces techniques récentes peut être encouragé par le volume important des données générées par les phases de prévision et d'optimisation.

L'application nommée "RMOSS" développée dans le cadre de cette étude a été mise en source ouverte. L'amélioration de cet outil informatique pour prendre en charge les fonctionnalités évoquées dans la discipline de l'optimisation de la gestion opérationnelle des réservoirs et systèmes de réservoirs, peut constituer une opportunité dans la matière.



# Bibliographie

- Adjim, H.** (2018). *CONTRIBUTION A L'ETUDE DES PHENOMENES HYDROLOGIQUES–CAS DE LA SECHERESSE Application au bassin du barrage Hammam Bouhrara*. Ph.D. thesis, Université Aboubekr Belkaid, Tlemcen.
- Adjim, H.** and **Djedid, A.** (2018). Drought and water mobilization in semi-arid zone : The example of Hammam Bouhrara dam (North-West of Algeria). *Journal of Water and Land Development*, volume 37(1) :pages 3–10.
- Ahmad, A., El-Shafie, A., Razali, S.F.M., and Mohamad, Z.S.** (2014). Reservoir optimization in water resources : a review. *Water resources management*, volume 28(11) :pages 3391–3405.
- Ahmed, J.A.** and **Sarma, A.K.** (2005). Genetic algorithm for optimal operating policy of a multipurpose reservoir. *Water resources management*, volume 19(2) :pages 145–161.
- Akhtar, M., Corzo, G., Van Andel, S., and Jonoski, A.** (2009). River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information : case study of the Ganges river basin. *Hydrology and Earth System Sciences*, volume 13(9) :pages 1607–1618.
- Akter, T.** and **Simonovic, S.P.** (2004). Modelling uncertainties in short-term reservoir operation using fuzzy sets and a genetic algorithm/Modélisation d'incertitudes dans la gestion de barrage à court terme grâce à des ensembles flous et à un algorithme génétique. *Hydrological sciences journal*, volume 49(6).
- Almeida, L.M.** and **Ludermir, T.** (2008a). An improved method for automatically searching near-optimal artificial neural networks. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, (pages 2235–2242).
- Almeida, L.M.** and **Ludermir, T.B.** (2008b). Tuning artificial neural networks parameters using an evolutionary algorithm. In *2008 Eighth International Conference on Hybrid Intelligent Systems*. IEEE, (pages 927–930).

- ANBT** (2012). Rapport d'exploitation du barrage Hammam Boughrara. Agence Nationale des Barrages et Transferts.
- ANBT** (2017). Données fournies par l'Agence Nationale des Barrages et Transferts.
- Ariel, D., Rosegrant, M., and Meinzen-Dick, R.** (1997). Water allocation mechanisms : principles and examples. *World Bank Working Paper*.
- Arunkumar, R. and Jothiprakash, V.** (2012). Optimal reservoir operation for hydropower generation using non-linear programming model. *Journal of The Institution of Engineers (India) : Series A*, volume 93(2) :pages 111–120.
- Atchia, M. and Tropp, S.** (1995). *Environmental Management. Issues and Solutions*. John Wiley and Sons, London.
- Ayad, A., Awad, H., and Yassin, A.** (2018). Integrated approach for the optimal design of pipeline networks. *Alexandria engineering journal*, volume 57(1) :pages 87–96.
- Barricelli, N.A.** (1957). Symbiogenetic evolution processes realized by artificial methods. *Methodos*, volume 9(35-36) :pages 143–182.
- Barricelli, N.A.** (1963). Numerical testing of evolution theories. *Acta Biotheoretica*, volume 16(3-4) :pages 99–126.
- Bellman, R.** (1957). Dynamic programming princeton university press princeton. *New Jersey Google Scholar*.
- Bellman, R.E. and Zadeh, L.A.** (1970). Decision-making in a fuzzy environment. *Management science*, volume 17(4) :pages B–141.
- Bensalem, I. and Bouhenna, N.** (2017). Etude de faisabilité de l'extension du périmètre irrigué de Maghnia. Mémoire Master en Hydraulique. Université Aboubekr Belkaid, Tlemcen.
- Blayo, F. and Verleysen, M.** (1996). *Les réseaux de neurones artificiels. Que sais-je ?* Presses universitaires de France. ISBN 9782130473558.
- Bledsoe, W.W.** (1962). The evolutionary method in hill climbing : Convergence rates. *Panoramic Research, Inc. Technical Report*.
- Borne, P., Benrejeb, M., and Haggège, J.** (2007). *Les réseaux de neurones : présentation et applications. Méthodes et pratiques de l'ingénieur : automatique*. Editions Technip. ISBN 9782710808961. URL <https://books.google.dz/books?id=P1GgimurnGoC>.
- Bouanani, A.** (2004). Hydrologie, Transport solide et Modélisation : Etude de quelques sous bassins de la Tafna (NW-Algérie). *Th. de Doct. d'état, Univ. Abou Bekr BELKAID, Tlemcen*.

- 
- Bounoua, R.** (2014). Gestion Du Barrage Hammam Boughrara : Application De MODSIM. Mémoire Magister -USTO-.
- Bouزيد-Lagha, S. and Djelita, B.** (2012). Etude du phénomène d'eutrophisation dans le Barrage de Hammam Boughrara (Wilaya de Tlemcen, Algérie). *Hydrological sciences journal*, volume 57(1) :pages 186–201.
- Bozorg-Haddad, O., Janbaz, M., and Loáiciga, H.A.** (2016). Application of the gravity search algorithm to multi-reservoir operation optimization. *Advances in water resources*, volume 98 :pages 173–185.
- Bremermann, H.J. et al.** (1962). Optimization through evolution and recombination. *Self-organizing systems*, volume 93 :page 106.
- Carpentier, P. and Cohen, G.** (2017). *Décomposition-coordination en optimisation déterministe et stochastique*. Mathématiques et Applications. Springer Berlin Heidelberg. ISBN 9783662554289. URL <https://books.google.dz/books?id=n64sDwAAQBAJ>.
- Celeste, A.B. and Billib, M.** (2009). Evaluation of stochastic reservoir operation optimization models. *Advances in Water Resources*, volume 32(9) :pages 1429–1443.
- Chang, F.J. and Chen, L.** (1998). Real-coded genetic algorithm for rule-based flood control reservoir management. *Water Resources Management*, volume 12(3) :pages 185–198.
- Chaves, P. and Chang, F.J.** (2008). Intelligent reservoir operation system based on evolving artificial neural networks. *Advances in Water Resources*, volume 31(6) :pages 926–936.
- Chaves, P. and Kojiri, T.** (2007). Stochastic fuzzy neural network : case study of optimal reservoir operation. *Journal of Water Resources Planning and Management*, volume 133(6) :pages 509–518.
- CIGB** (2020). Commission internationale des grands barrages. <https://www.icold-cigb.org>.
- CLEMENT, R., GALAND, A., and MEYLAN, J.** (1978). Systemes d'irrigation.
- Comité Français des Barrages et Réservoirs, C.** (2013). L'histoire des barrages.
- Cybenko, G.** (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, volume 2(4) :pages 303–314.
- Dahmani, S., Ferhait, A., Yebdri, D., Bounoua, R., and Khodja, H.D.** (2018). Development of Reservoir Management Optimal Rules : Case of Hammam Boughrara Dam, Wilaya of Tlemcen, Algeria. In *Advances in Hydroinformatics*. Springer, (pages 347–358).

- Dahmani, S.** and **Yebdri, D.** (2020). Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Reservoir Operation Management. *Water Resources Management* :pages 1–16.
- Darras, T.** (2015). *Prévision de crues rapides par apprentissage statistique*. Ph.D. thesis.
- Devisree, M.** and **Nowshaja, P.** (2014). Optimisation of Reservoir Operations Using Genetic Algorithms. *International Journal of Scientific & Engineering Research*, volume 5(7).
- Dhar, A.** and **Datta, B.** (2008). Optimal operation of reservoirs for downstream water quality control using linked simulation optimization. *Hydrological Processes : An International Journal*, volume 22(6) :pages 842–853.
- Dodge, Y.** (2006). *Optimisation appliquée*. Optimisation appliquée. Springer. ISBN 9782287268359.
- DRE** (2012). Données hydrométriques de l'Oued Mouilah. Direction des ressources en eau de la wilaya de Tlemcen.
- Dudani, K.** and **Chudasama, A.** (2016). Partial discharge detection in transformer using adaptive grey wolf optimizer based acoustic emission technique. *Cogent Engineering*, volume 3(1) :page 1256083.
- Eberhart, R.** and **Kennedy, J.** (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, (pages 39–43).
- Faris, H., Aljarah, I., Al-Betar, M.A., and Mirjalili, S.** (2018). Grey wolf optimizer : a review of recent variants and applications. *Neural computing and applications*, volume 30(2) :pages 413–435.
- Ford Jr, L.R.** and **Fulkerson, D.R.** (2015). *Flows in networks*. Princeton university press.
- François, B.** (2013). *Gestion optimale d'un réservoir hydraulique multiusages et changement climatique. Modèles, projections et incertitudes : Application à la réserve de Serre-Ponçon*. Ph.D. thesis, Université de Grenoble.
- Fraser, A.S.** (1957). Simulation of genetic systems by automatic digital computers I. Introduction. *Australian Journal of Biological Sciences*, volume 10(4) :pages 484–491.
- Ghenim, A.** (2008). *Étude des écoulements et des transports solides dans les régions semi-arides méditerranéennes*. Ph.D. thesis, Thèse de doctorat : Université de Tlemcen (Algérie).
- Ghimire, B.N.** and **Reddy, M.J.** (2013). Optimal reservoir operation for hydropower production using particle swarm optimization and sustainability analysis of hydropower. *ISH Journal of Hydraulic Engineering*, volume 19(3) :pages 196–210.

- 
- Glover, F.** (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, volume 13(5) :pages 533–549.
- Glover, F.** and **Kochenberger, G.** (2006). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer US. ISBN 9780306480560. URL <https://books.google.dz/books?id=P-HpBwAAQBAJ>.
- Goldberg, D.E.** and **Holland, J.H.** (1988). Genetic algorithms and machine learning. *Machine learning*, volume 3(2) :pages 95–99.
- Gonzalez, T.** (2007). *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC Computer and Information Science Series. CRC Press. ISBN 9781420010749. URL [https://books.google.dz/books?id=QK3\\\_VU8ngK8C](https://books.google.dz/books?id=QK3\_VU8ngK8C).
- Gonzalez-Villarreal, F.** and **Solanes, M.** (1999). The Dublin Principles for water as reflected in a comparative assessment of institutional and legal arrangements for integrated water resources management. *Global Water Partnership Technical Advisory Committee, Santiago, Chile GOZ (1998) Water Bill, draft, Ministry of Rural Resource and Water Development. Rural Water Supply and Sanitation–Review of the IRWSSP*, volume 2.
- Gray, N.** (2005). *Water Technology : An Introduction for Environmental Scientists and Engineers*. Elsevier Butterworth-Heinemann. ISBN 9780750666336. URL <https://books.google.dz/books?id=p9e8r1b2UqwC>.
- Gutjahr, W.J.** (2009). Convergence analysis of metaheuristics. In *Matheuristics*. Springer, (pages 159–187).
- Hall, W.A., Butcher, W.S., and Esogbue, A.** (1968). Optimization of the operation of a multiple-purpose reservoir by dynamic programming. *Water Resources Research*, volume 4(3) :pages 471–477.
- Hamlat, A., Tidjani, A.E.B., Yebdri, D., Errih, M., and Guidoum, A.** (2014). Water quality analysis of reservoirs within Western Algeria catchment areas using water quality index CCME WQI. *Journal of Water Supply : Research and Technology—AQUA*, volume 63(4) :pages 311–324.
- Harms, A.A.** and **Campbell, T.H.** (1967). An extension to the Thomas-Fiering Model for the sequential generation of streamflow. *Water Resources Research*, volume 3(3) :pages 653–661.
- Hashimoto, T., Stedinger, J.R., and Loucks, D.P.** (1982). Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation. *Water resources research*, volume 18(1) :pages 14–20.

- Haykin, S.** (2009). *Neural Networks and Learning Machines, 3/E*. Pearson Education India.
- Holland, J.H.** (1975). Adaptation in natural and artificial systems Ann Arbor. *The University of Michigan Press*, volume 1 :page 975.
- Hornik, K., Stinchcombe, M., and White, H.** (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, volume 2(5) :pages 359–366.
- Howe, C.W., Schurmeier, D.R., and Shaw Jr, W.D.** (1986). Innovative approaches to water allocation : the potential for water markets. *Water resources research*, volume 22(4) :pages 439–445.
- Hudson, N.W.** (1990). *Conservation des sols et des eaux dans les zones semi-arides*, volume 57. Food & Agriculture Org FAO.
- Institut Méditerranéen de l'Eau, I.** (2010). RAPPORT REGIONAL SUR LA GESTION DES SEDIMENTS DANS LES BARRAGES EN MEDITERRANEE - RAPPORT NATIONAL ALGERIE.
- Janga Reddy, M. and Nagesh Kumar, D.** (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering : a state-of-the-art review. *H2Open Journal (Open Access Journal)*.
- Janikow, C.Z. and Michalewicz, Z.** (1991). An experimental comparison of binary and floating point representations in genetic algorithms. In *ICGA*, volume 1991. (pages 31–36).
- Jia, Y. and Culver, T.B.** (2006). Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. *Journal of Hydrology*, volume 331(3-4) :pages 580–590.
- Jordan, F.** (2007). Modèle de prévision et de gestion des crues. Technical report, EPFL.
- Jorgensen, S.E., Loffler, H., Rast, W., and Straskraba, M.** (2005). *Lake and reservoir management*, volume 54. Elsevier.
- Jothiprakash, V. and Shanthi, G.** (2006). Single reservoir operating policies using genetic algorithm. *Water Resources Management*, volume 20(6) :pages 917–929.
- Jothiprakash, V. and Shanthi, G.** (2009). Comparison of policies derived from stochastic dynamic programming and genetic algorithm models. *Water resources management*, volume 23(8) :pages 1563–1580.
- Kalteh, A.M.** (2013). Monthly river flow forecasting using artificial neural network and support vector regression models coupled with wavelet transform. *Computers & Geosciences*, volume 54 :pages 1–8.

- 
- Kamodkar, R. and Regulwar, D.** (2010). Multipurpose single reservoir operation with fuzzy technological constraints. *ISH Journal of Hydraulic Engineering*, volume 16(sup1) :pages 49–63.
- Karami, H., Ehteram, M., Mousavi, S.F., Farzin, S., Kisi, O., and El-Shafie, A.** (2018). Optimization of energy management and conversion in the water systems based on evolutionary algorithms. *Neural Computing and Applications* :pages 1–14.
- Karmarkar, N.** (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. (pages 302–311).
- Kelman, J. and Kelman, R.** (2002). Water allocation for economic production in a semi-arid region. *International Journal of Water Resources Development*, volume 18(3) :pages 391–407.
- Kelman, J., Stedinger, J.R., Cooper, L.A., Hsu, E., and Yuan, S.Q.** (1990). Sampling stochastic dynamic programming applied to reservoir operation. *Water Resources Research*, volume 26(3) :pages 447–454.
- Kiranyaz, S., Ince, T., Yildirim, A., and Gabbouj, M.** (2009). Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. *Neural networks*, volume 22(10) :pages 1448–1462.
- Kişi, Ö.** (2004). River flow modeling using artificial neural networks. *Journal of Hydrologic Engineering*, volume 9(1) :pages 60–63.
- Kişi, Ö.** (2007). Streamflow forecasting using different artificial neural network algorithms. *Journal of Hydrologic Engineering*, volume 12(5) :pages 532–539.
- Klemeš, V.** (1977). Discrete representation of storage for stochastic reservoir optimization. *Water resources research*, volume 13(1) :pages 149–158.
- Krause, P., Boyle, D., and Bäse, F.** (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in geosciences*, volume 5 :pages 89–97.
- Labadie, J.W.** (2004). Optimal operation of multireservoir systems : state-of-the-art review. *Journal of water resources planning and management*, volume 130(2) :pages 93–111.
- Laborde, J.** (2000). *Eléments d'hydrologie de surface. Université de Nice-Sophia Antipolis. France (204p).*
- Lebdi, F., Slimani, M., and Parent, E.** (1997). Stratégie empirique d'un système de ressources en eau : l'exemple d'un périmètre irrigué en zone semi-aride. *Revue des sciences de l'eau/Journal of Water Science*, volume 10(1) :pages 121–134.

- Leblanc, G.** (2006). *C# et .NET*. Eyrolles. ISBN 9782212117783. URL <https://books.google.dz/books?id=xuuGtXgJAXwC>.
- Lin, G.F., Chen, G.R., Huang, P.Y., and Chou, Y.C.** (2009). Support vector machine-based models for hourly reservoir inflow forecasting during typhoon-warning periods. *Journal of Hydrology*, volume 372(1-4) :pages 17–29.
- Long, W., Liang, X., Cai, S., Jiao, J., and Zhang, W.** (2017). A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. *Neural Computing and Applications*, volume 28(1) :pages 421–438.
- Mala-Jetmarova, H., Sultanova, N., and Savic, D.** (2017). Lost in optimisation of water distribution systems? A literature review of system operation. *Environmental modelling & software*, volume 93 :pages 209–254.
- Marquardt, D.W.** (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, volume 11(2) :pages 431–441.
- Marshall, L., Nott, D., and Sharma, A.** (2004). A comparative study of Markov chain Monte Carlo methods for conceptual rainfall-runoff modeling. *Water Resources Research*, volume 40(2).
- Mathlouthi, M. and Lebdi, F.** (2008). a fiabilEvaluation de lité de gestion d’un barrage réservoir pour des événements secs/Assessment of the reliability of reservoir dam operation for dry events. *Hydrological sciences journal*, volume 53(6) :pages 1194–1207.
- McKinney, D.C. and Lin, M.D.** (1994). Genetic algorithm solution of groundwater management models. *Water Resources Research*, volume 30(6) :pages 1897–1906.
- Microsoft** (2020a). Microsoft .NET framework. <https://dotnet.microsoft.com/>.
- Microsoft** (2020b). Visual Studio Community. <https://visualstudio.microsoft.com/fr/vs/community/>.
- Mirjalili, S., Hashim, S.Z.M., and Sardroudi, H.M.** (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, volume 218(22) :pages 11125–11137.
- Mirjalili, S., Mirjalili, S.M., and Lewis, A.** (2014). Grey wolf optimizer. *Advances in engineering software*, volume 69 :pages 46–61.
- Mishra, B. and Patnaik, R.K.** (2009). *Genetic Algorithm and its Variants : Theory and Applications*. Ph.D. thesis.

- Mittal, N., Singh, U., and Sohi, B.S.** (2016). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, volume 2016 :page 8.
- Mono** (2020). Mono software platform. <https://www.mono-project.com/>.
- Moré, J.J.** (1978). The Levenberg-Marquardt algorithm : implementation and theory. In *Numerical analysis*. Springer, (pages 105–116).
- Mousavi, S.J., Moghaddam, K.S., and Seifi, A.** (2004). Application of an interior-point algorithm for optimization of a large-scale reservoir system. *Water resources management*, volume 18(6) :pages 519–540.
- Moy, W.S., Cohon, J.L., and ReVelle, C.S.** (1986). A programming model for analysis of the reliability, resilience, and vulnerability of a water supply reservoir. *Water resources research*, volume 22(4) :pages 489–498.
- Muro, C., Escobedo, R., Spector, L., and Coppinger, R.** (2011). Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes*, volume 88(3) :pages 192–197.
- Nagesh Kumar, D. and Janga Reddy, M.** (2007). Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management*, volume 133(3) :pages 192–201.
- Nagesh Kumar, D., Raju, K.S., and Ashok, B.** (2006). Optimal reservoir operation for irrigation of multiple crops using genetic algorithms. *Journal of irrigation and drainage engineering*, volume 132(2) :pages 123–129.
- Nagy, I.V., Asante-Duah, K., and Zsuffa, I.** (2013). *Hydrological dimensioning and operation of reservoirs : practical design concepts and principles*, volume 39. Springer Science & Business Media.
- Nash, J.E. and Sutcliffe, J.V.** (1970). River flow forecasting through conceptual models part I—A discussion of principles. *Journal of hydrology*, volume 10(3) :pages 282–290.
- Oliveira, B. and Maia, R.** (2018). Stochastic Generation of Streamflow Time Series. *Journal of Hydrologic Engineering*, volume 23(10) :page 04018043.
- Pabiot, F.** (1999). Optimisation de la gestion d'un barrage collinaire en zone semi-aride : projet MERGUSIE.
- Panahi, F., Ehteram, M., Kisi, O., Afshar nia, S., and El-Shafie, A.** (2019). Forecasting rainfall in arid regions with an improved Levenberg- Marquardt algorithm based on Taguchi model, crow optimization algorithm and principal component analysis. *In publication*.

- Parent, E.** (1991). *Elaboration des consignes de gestion des barrages-réservoirs*. Ph.D. thesis.
- Pólya, G.** (1945). How to solve it. Princeton. *New Jersey : Princeton University*.
- Rani, D.** and **Moreira, M.M.** (2010). Simulation–optimization modeling : a survey and potential application in reservoir systems operation. *Water resources management*, volume 24(6) :pages 1107–1138.
- Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S.** (2009). GSA : a gravitational search algorithm. *Information sciences*, volume 179(13) :pages 2232–2248.
- Reddy, M.J.** and **Kumar, D.N.** (2007). Optimal reservoir operation for irrigation of multiple crops using elitist-mutated particle swarm optimization. *Hydrological Sciences Journal*, volume 52(4) :pages 686–701.
- Remini, B.** (2005). *La problématique de l'eau en Algérie*. Office des publications universitaires (OPU). ISBN 996100884-7.
- Richter, B.** and **Evrard, O.** (2017). *La crise de l'eau : De la pénurie à la gestion durable*. Plaisirs des sciences. De Boeck Supérieur. ISBN 9782807306356. URL <https://books.google.dz/books?id=Co0zDgAAQBAJ>.
- Ritzel, B.J., Eheart, J.W., and Ranjithan, S.** (1994). Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, volume 30(5) :pages 1589–1603.
- Roche, P., Miquel, J., and Gaume, E.** (2012). *Hydrologie quantitative : Processus, modèles et aide à la décision*. Ingénierie et développement durable. Springer Paris. ISBN 9782817801056. URL [https://books.google.dz/books?id=aMAMzJjQ\\\_nYC](https://books.google.dz/books?id=aMAMzJjQ\_nYC).
- Rosenblatt, F.** (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, volume 65(6) :page 386.
- Roslyn-Compiler** (2020). The Roslyn .NET compiler. <https://github.com/dotnet/roslyn>.
- Roweis, S.** (1996). Levenberg-marquardt optimization. *Notes, University Of Toronto*.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., et al.** (1988). Learning representations by back-propagating errors. *Cognitive modeling*, volume 5(3) :page 1.
- Russell, S.O.** and **Campbell, P.F.** (1996). Reservoir operating rules with fuzzy programming. *Journal of water resources planning and management*, volume 122(3) :pages 165–170.
- Sabri, N.M., Puteh, M., and Mahmood, M.R.** (2013). A review of gravitational search algorithm. *Int. J. Advance. Soft Comput. Appl.*, volume 5(3) :pages 1–39.

- Seifi, A. and Hipel, K.W.** (2001). Interior-point method for reservoir operation with stochastic inflows. *Journal of water resources planning and management*, volume 127(1) :pages 48–57.
- Sharma, P.J., Patel, P., and Jothiprakash, V.** (2015). Stochastic Modelling for Inflow Prediction into Ukai Reservoir, India. In *Proceedings of 20th International Conference on Hydraulics, Water Resources and River Engineering (HYDRO 2015 International)*, IIT Roorkee, India.
- Singh, N. and Singh, S.** (2017). Hybrid algorithm of particle swarm optimization and Grey Wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, volume 2017.
- Smail, N.** (2014). *SUIVI DU COMPORTEMENT DES BARRAGES Etude de cas*. Ph.D. thesis, Université Aboubekr Belkaid, Tlemcen.
- Souza, C., Kirillov, A., Catalano, M.D., and contributors, A.** (2014). The Accord.NET Framework. 10.5281/zenodo.1029480. URL <http://accord-framework.net>.
- Srikanthan, R.** (1978). Sequential generation of monthly streamflows. *Journal of Hydrology*, volume 38(1-2) :pages 71–80.
- swMath.org** (2020). Information service for mathematical software. <http://swmath.org/>.
- Talbi, E.** (2009). *Metaheuristics : From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley. ISBN 9780470496909.
- Talbi, E.G.** (2002). A taxonomy of hybrid metaheuristics. *Journal of heuristics*, volume 8(5) :pages 541–564.
- Talbi, E.G. et al.** (2013). *Hybrid metaheuristics*, volume 166. Springer.
- Thirumalaiah, K. and Deo, M.** (1998). River stage forecasting using artificial neural networks. *Journal of Hydrologic Engineering*, volume 3(1) :pages 26–32.
- Thirumalaiah, K. and Deo, M.C.** (2000). Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering*, volume 5(2) :pages 180–189.
- Torabi, M. and Mobasher, F.** (1973). A STOCHASTIC DYNAMIC PROGRAMMING MODEL FOR THE OPTIMUM OPERATION OF A MULTI-PURPOSE RESERVOIR 1. *JAWRA Journal of the American Water Resources Association*, volume 9(6) :pages 1089–1099.
- TRACTEBEL, E.D.** (2003). Morphologie du barrage Hammam Boughrara. Technical report, TRACTEBEL.

- Valipour, M., Banihabib, M.E., and Behbahani, S.M.R.** (2013). Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *Journal of hydrology*, volume 476 :pages 433–441.
- Vermeiren, L. and Jobling, G.** (1983). *L'irrigation localisée : calcul, mise en place, exploitation, contrôle du fonctionnement*. Bulletin F.A.O. d'irrigation et de drainage. Organisation des Nations Unies pour l'alimentation et l'agriculture. ISBN 9789252009863. URL <https://books.google.dz/books?id=9Fz408knASwC>.
- Votruba, L. and Broža, V.** (1989). *Water management in reservoirs*, volume 33. Elsevier.
- Wilamowski, B.** (2010). Efficient neural network architectures and advanced training algorithms. *Gdańsk University of Technology Faculty of ETI Annals*, volume 18 :pages 345–352.
- Wilamowski, B. and Irwin, J.** (2016). *Intelligent Systems*. Electrical engineering handbook series. CRC Press. ISBN 9781439802847. URL <https://books.google.dz/books?id=NGfMBQAAQBAJ>.
- Willmott, C.J.** (1981). On the validation of models. *Physical geography*, volume 2(2) :pages 184–194.
- Wolpert, D.H. and Macready, W.G.** (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, volume 1(1) :pages 67–82.
- World Commission on Dams, W.** (2000). *Dams and development : A new framework for decision-making : The report of the world commission on dams*. Earthscan.
- Wright, A.H.** (1991). Genetic algorithms for real parameter optimization. In *Foundations of genetic algorithms*, volume 1. Elsevier, (pages 205–218).
- Wurbs, R.A.** (1991). *Optimization of multiple-purpose reservoir system operations : a review of modeling and analysis approaches*. 34. Hydrologic Engineering Center, US Army Corps of Engineers.
- Wurbs, R.A.** (1993). Reservoir-system simulation and optimization models. *Journal of water resources planning and management*, volume 119(4) :pages 455–472.
- Xamarin** (2020). Xamarin software platform. <https://dotnet.microsoft.com/apps/xamarin>.
- Yang, X.S., Gandomi, A.H., Talatahari, S., and Alavi, A.H.** (2012). *Metaheuristics in water, geotechnical and transport engineering*. Newnes.
- Yao, X.** (1993). Evolutionary artificial neural networks. *International journal of neural systems*, volume 4(03) :pages 203–222.

- Yapo, P.O.** (1992). A Markov chain flow model with application to flood forecasting.
- Yaseen, Z.M., Karami, H., Ehteram, M., Mohd, N.S., Mousavi, S.F., Hin, L.S., Kisi, O., Farzin, S., Kim, S., and El-Shafie, A.** (2018). Optimization of reservoir operation using new hybrid algorithm. *KSCE Journal of Civil Engineering*, volume 22(11) :pages 4668–4680.
- Yeh, W.W.G.** (1985). Reservoir management and operations models : A state-of-the-art review. *Water resources research*, volume 21(12) :pages 1797–1818.
- Yeniay, Ö.** (2005). Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and computational Applications*, volume 10(1) :pages 45–56.
- Young, G.K.** (1967). Finding reservoir operating rules. *Journal of the Hydraulics Division*, volume 93(6) :pages 297–322.
- Zadeh, L.A.** (1965). Fuzzy sets. *Information and control*, volume 8(3) :pages 338–353.
- Zenasni, S.** (2013). *Modélisation de la qualité des eaux d'oued mouillah jusqu'au barrage Boughrara –Application du HEC-RAS- Mémoire du Magister*. Ph.D. thesis, Université USTO, Oran.
- Zéphyr, L.** (2015). Optimisation stochastique des systèmes multi-réservoirs par l'agrégation de scénarios et la programmation dynamique approximative.

# **Annexes**

## Annexe A

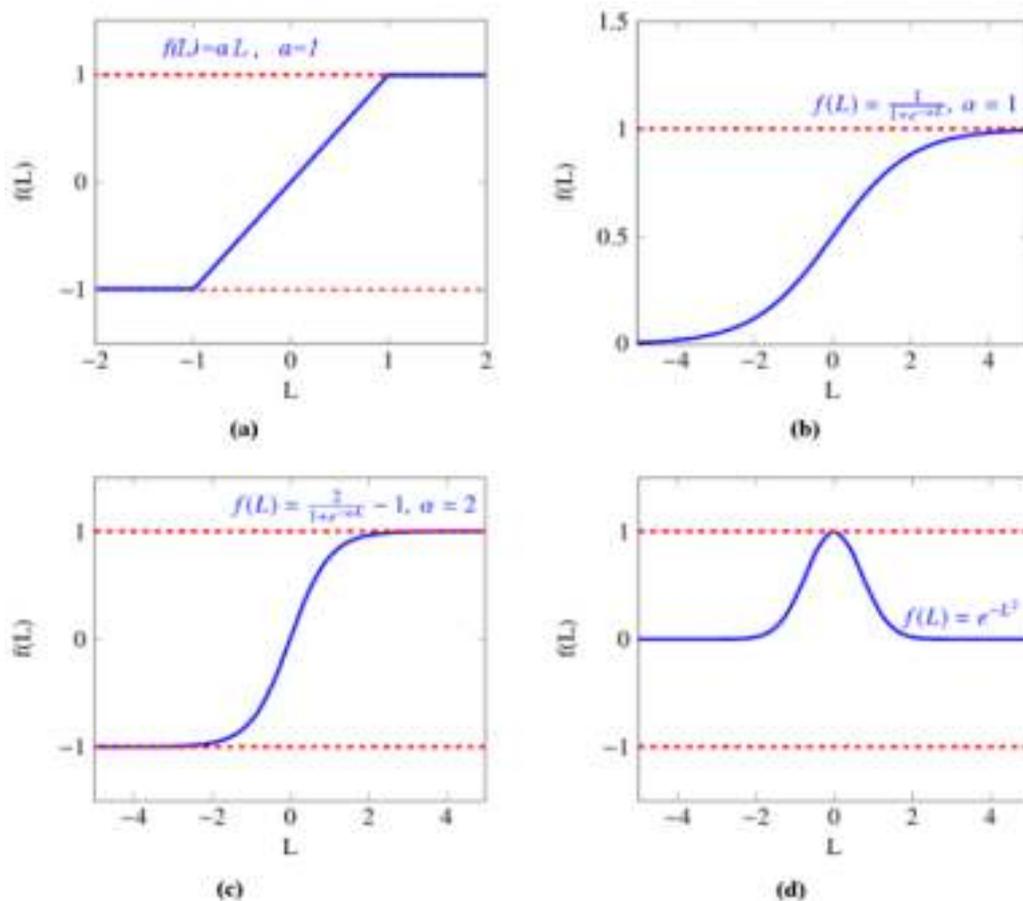


FIGURE 3 – Les fonctions d’activation usuelles ; (a) : Linéaire, (b) : Sigmoide, (c) : Bipolaire Sigmoide, (d) : Gaussienne.

### Principe de l’algorithme de Levenberg-Marquardt

L’algorithme d’apprentissage de Levenberg-Marquardt a été conçu pour approcher la vitesse d’entraînement de second ordre sans avoir à calculer la matrice Hessienne (Moré 1978) définie comme suit :

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial W_1^2} & \frac{\partial^2 E}{\partial W_1 \partial W_2} & \cdots & \frac{\partial^2 E}{\partial W_1 \partial W_N} \\ \frac{\partial^2 E}{\partial W_2 \partial W_1} & \frac{\partial^2 E}{\partial W_2^2} & \cdots & \frac{\partial^2 E}{\partial W_2 \partial W_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 E}{\partial W_N \partial W_1} & \frac{\partial^2 E}{\partial W_N \partial W_2} & \cdots & \frac{\partial^2 E}{\partial W_N^2} \end{bmatrix}$$

Lorsque la fonction d’erreur d’apprentissage est de forme d’une somme de carrés (typiquement aux équations 3.10 et 3.14), la matrice Hessienne peut être approximée en utilisant la matrice Jacobienne  $J$  contenant les dérivées de 1<sup>er</sup> ordre des erreurs d’apprentissage par

rapport aux poids et aux biais. L'intérêt ici, est que la Jacobienne  $J$  est plus facilement calculable par rapport à la matrice Hessienne  $H$  (Kişi 2007).

$$H = J^T J \quad (1)$$

Avec :

$$J = \begin{bmatrix} \frac{\partial e_{11}}{\partial W_1} & \frac{\partial e_{11}}{\partial W_2} & \cdots & \frac{\partial e_{11}}{\partial W_N} \\ \frac{\partial e_{21}}{\partial W_1} & \frac{\partial e_{21}}{\partial W_2} & \cdots & \frac{\partial e_{21}}{\partial W_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{M1}}{\partial W_1} & \frac{\partial e_{M1}}{\partial W_2} & \cdots & \frac{\partial e_{M1}}{\partial W_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{1P}}{\partial W_1} & \frac{\partial e_{1P}}{\partial W_2} & \cdots & \frac{\partial e_{1P}}{\partial W_N} \\ \frac{\partial e_{2P}}{\partial W_1} & \frac{\partial e_{2P}}{\partial W_2} & \cdots & \frac{\partial e_{2P}}{\partial W_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{MP}}{\partial W_1} & \frac{\partial e_{MP}}{\partial W_2} & \cdots & \frac{\partial e_{MP}}{\partial W_N} \end{bmatrix}$$

Et,

$$E = (e_{11} \dots e_{M1} e_{1P} \dots e_{MP})^T \quad (2)$$

Le vecteur d'erreur  $E$  contient les erreurs calculées pour chaque paire d'entraînement. Ces erreurs sont calculées comme suit :

$$e_{ip} = y_{ip} - s_{ip} \quad (3)$$

Afin de s'assurer que la matrice Hessienne Approximative  $J^T J$  est inversible, l'algorithme Levenberg-Marquardt introduit une autre approximation de la matrice Hessienne (Kişi 2007) :

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k^T e \quad (4)$$

Tels que :

- $\mu$  : Paramètre modifié au cours du processus d'apprentissage ;
- $I$  : la matrice d'identité.

Durant le processus d'entraînement, la méthode de Levenberg-Marquardt change alternativement le comportement entre deux méthodes, Gauss-Newton et gradient descent. Lorsque  $\mu = 0$ , la méthode de Levenberg-Marquardt se comporte identiquement comme Gauss-Newton. Cependant, si  $\mu$  est important, il se rapproche au coefficient d'apprentissage  $a$ , et la méthode se transforme en gradient descent.

$$a = \frac{1}{\mu} \quad (5)$$

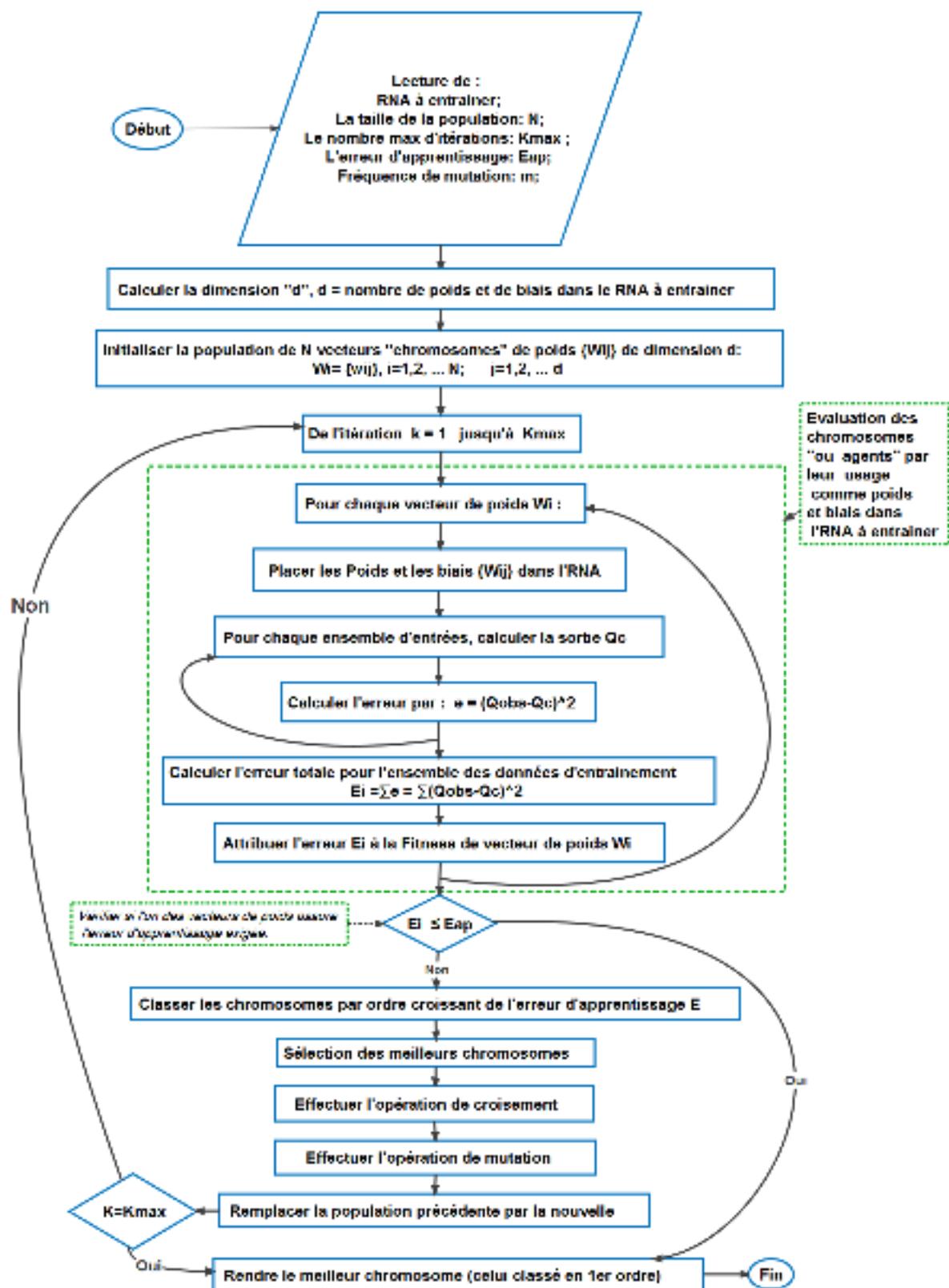


FIGURE 4 – Organigramme explicatif de l'entraînement des RNA par l'algorithme génétique RGA

**Algorithm 1:** Algorithme de rétro-propagation "BP"

---

**Data:**  $exemples, E_{app}, \alpha, K_{max}, RNA$  //Couples d'apprentissage, marge d'erreur d'apprentissage, facteur d'apprentissage, nombre maximal d'itérations, le réseau de neurone à entraîner, de  $L$  couches, poids synaptiques  $W_{ij}$ , et fonction d'activation  $f(\cdot)$  //

**Result:**  $RNA$

```

1 begin
2    $\delta[] \leftarrow [0]$ ; //Vecteur des erreurs indexé par les noeuds "neurones" du RNA//
3   foreach  $W_{ij}$  dans  $RNA$  do
4     |  $W_{ij} \leftarrow Rand(-1, 1)$ ; // Attribuer aux poids des valeurs aléatoires  $\in$ 
      |  $[-1,1]$ //
5   end
6   repeat
7     |  $k \leftarrow 0$ ;  $E \leftarrow 0$ ; //Initialiser le compteur d'itérations, et l'erreur
      | d'apprentissage//
8     foreach  $exemple(x, y)$  dans  $exemples$  do
9       | foreach  $noeud i$  dans la couche d'entrée  $l = 0$  do
10        |  $s_i \leftarrow x_i$ ;
11        end
12        foreach  $l = 1$  jusqu'à  $L$  do
13          | foreach  $noeud j$  dans la couche  $l$  do
14            |  $in_j \leftarrow \sum W_{ij}s_i$ ;  $s_j \leftarrow f(in_j)$ ;
15            end
16          end
17          //Propager l'erreur de la couche finale vers la première couche//
18          foreach  $noeud j$  dans la couche de sortie do
19            |  $\delta[j] \leftarrow y_j - s_j$ ;  $E \leftarrow E + \delta[j]$ ;
20            end
21          foreach  $l = L - 1$  jusqu'à  $1$  do
22            | foreach  $noeud i$  dans la couche  $l$  do
23              |  $\delta[j] \leftarrow f(in_i)(1 - f(in_i)) \sum_j W_{ij}\delta[j]$ ;
24              end
25            end
26            //Mis à jour des poids de l'RNA au moyen des erreurs  $\delta$  //
27            foreach  $poids W_{ij}$  dans  $RNA$  do
28              |  $W_{ij} \leftarrow W_{ij} + \mu\delta[j]s_i$ ;
29            end
30          end
31        until  $E \leq E_{app}$  ou  $k \geq K_{max}$  //Critères d'arrêt//
32 end

```

---

**Algorithm 2:** Algorithme de Levenberg-Marquardt "LM"

**Data:**  $exemples, E_{app}, \mu, K_{max}, RNA$  //Couples d'apprentissage, marge d'erreur d'apprentissage, facteur d'apprentissage, nombre maximal d'itérations, le réseau de neurone à entraîner, composé de couches  $L$ , poids synaptiques  $W_{ij}$ , et fonction d'activation  $f(\cdot)$  //

**Result:**  $RNA$

```

1 begin
2   foreach  $W_{ij}$  dans  $RNA$  do
3      $W_{ij} \leftarrow Rand(-1, 1)$ ; // Attribuer aux poids des valeurs aléatoires  $\in$ 
4      $[-1, 1]$ //
5   end
6   repeat
7      $k \leftarrow 0$ ;  $E_k \leftarrow [0]$ ; //Initialiser le compteur d'itérations, et les erreurs
8     d'apprentissage//
9     foreach  $exemple(x, y)$  dans  $exemples$  do
10      //Propager les entrées dans le RNA pour calculer les sorties (les
11      memes étapes que l'algorithme de BP) //
12      Executer les lignes 10  $\rightarrow$  17 de l'Algorithme 1;
13      //Calcul de l'erreur //
14      foreach  $noeud j$  dans la couche de sortie do
15         $E_k[j] \leftarrow (y_j - s_j)^2$ ;
16      end
17      foreach  $l = L - 1$  jusqu'à 1 do
18        foreach  $noeud i$  dans la couche  $l$  do
19           $\delta[j] \leftarrow f(O_i)(1 - f(O_i)) \sum_j W_{ij} \delta[j]$ ;
20        end
21      end
22      //Mis à jour des poids de l'RNA au moyen des erreurs  $\delta$  //
23      foreach  $poids W_{ij}$  dans  $RNA$  do
24         $W_{ij} \leftarrow W_{ij} + \mu \delta[j] s_i$ ;
25      end
26    end
27  until  $E \leq E_{app}$  ou  $k \geq K_{max}$  //Critères d'arrêt//
28 end

```

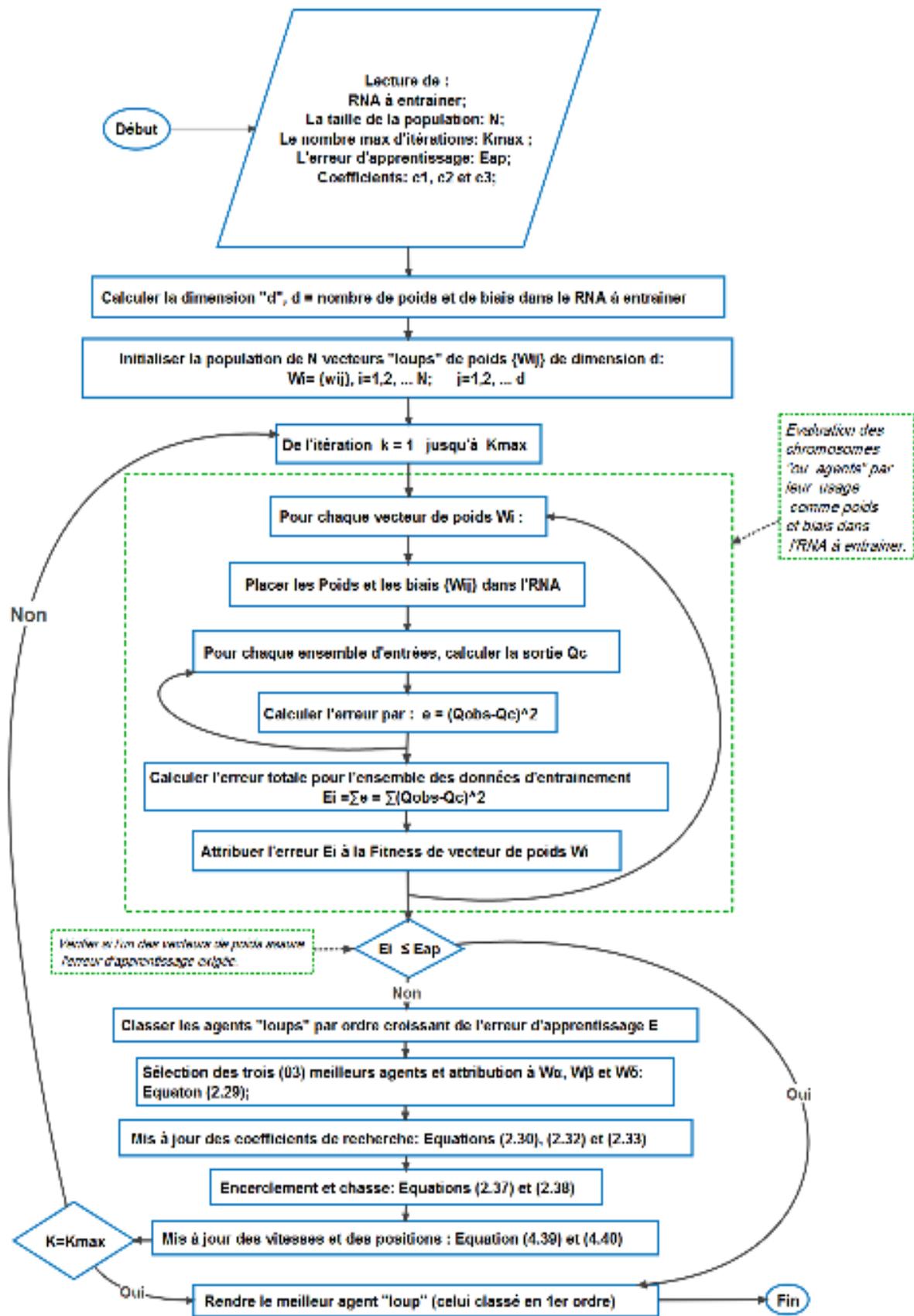


FIGURE 5 – Organigramme explicatif de l'entraînement des RNA par l'algorithme HPSOGWO

## Annexe B

(1,2, 3)	1	ID Fonction d'activation $f(\cdot)$	(1,2, 3)	1	ID Fonction d'activation $f(\cdot)$
[0,1, 5]	2	Valeur de coefficient $\sigma_{k_{k+1}}$	[0,1, 5]	2	Valeur de coefficient $\sigma_{k_{k+1}}$
[30, 120]	3	Taille de population $N$	[20, 120]	3	Taille de population $N$
[0,05, 0,9]	4	Fréquence de mutation $\omega_{\mu}$	[0,05, 5]	4	Valeur de coefficient $C_1$
[0,005, 0,05]	5	Erreur d'apprentissage $E_{k_{k+1}}$	[0,05, 5]	5	Valeur de coefficient $C_2$
[500, 3000]	6	Nombre maximal d'itérations $K_{k_{k+1}}$	[0,05, 5]	6	Valeur de coefficient $C_3$
[1, 5]	7	Nombre de couches cachées $N_c$	[0,005, 0,05]	7	Erreur d'apprentissage $E_{k_{k+1}}$
[1, 15]	8	Nombre de neurones de la couche cachée $N^{c1}$	[500, 3000]	8	Nombre maximal d'itérations $K_{k_{k+1}}$
[1, 15]	9	Nombre de neurones de la couche cachée $N^{c2}$	[1, 5]	9	Nombre de couches cachées $N_c$
[1, 15]	10	Nombre de neurones de la couche cachée $N^{c3}$	[1, 15]	10	Nombre de neurones de la couche cachée $N^{c1}$
[1, 15]	11	Nombre de neurones de la couche cachée $N^{c4}$	[1, 15]	11	Nombre de neurones de la couche cachée $N^{c2}$
[1, 15]	12	Nombre de neurones de la couche cachée $N^{c5}$	[1, 15]	12	Nombre de neurones de la couche cachée $N^{c3}$
			[1, 15]	13	Nombre de neurones de la couche cachée $N^{c4}$
			[1, 15]	14	Nombre de neurones de la couche cachée $N^{c5}$

FIGURE 6 – Codage des solutions et intervalles de définition des variables selon l'algorithme d'apprentissage : (a) l'algorithme génétique à codage réel "RGA", (b) l'algorithme HPSOGWO

**Annexe C**

TABLE 1 – Répartition hypsométrique des surfaces du bassin d'Oued Mouilah (Bouanani 2004)

Intervalles d'altitudes	Surfaces partielles ( $km^2$ )	Surfaces cumulées ( $km^2$ )	Surfaces partielles (%)	Surfaces cumulées (%)
1400-1460	26.8	26.8	1.01	1.01
1200-1400	306	332.8	11.54	12.55
1000-1200	381	713.8	14.37	26.93
800-1000	375	1088.8	14.15	41.07
600-800	414.5	1503.3	15.64	56.71
400-600	904	2407.3	34.10	90.81
285-400	243.5	2650.8	9.19	100.00

TABLE 2 – Classification de l'IRD (ex, ORSTOM) de relief à partir de l'indice de pente globale

R1	Relief très faible	$I_g < 0,002$
R2	Relief faible	$0.002 < I_g < 0.005$
R3	Relief assez faible	$0.005 < I_g < 0.01$
<b>R4</b>	<b>Relief modéré</b>	<b><math>0.01 &lt; I_g &lt; 0.02</math></b>
R5	Relief assez fort	$0.02 < I_g < 0.05$
R6	Relief fort	$0.05 < I_g < 0.1$
R7	Relief très fort	$0.1 < I_g$

TABLE 3 – Classification de relief de l'IRD (ex, ORSTOM) en fonction de la dénivelée spécifique (Laborde 2000)

R1	Relief très faible	$D_s < 10m$
R2	Relief faible	$10 m < D_s < 25m$
R3	Relief assez faible	$25m < D_s < 50m$
R4	Relief modéré	$50m < D_s < 100m$
R5	Relief assez fort	$100m < D_s < 250m$
R6	Relief fort	$250m < D_s < 500m$
R7	Relief très fort	$500m < D_s$

## Annexe D

TABLE 4 – Fuites par infiltration du barrage Hammam Bouhrara ( $Mm^3$ ) (Bounoua 2014)

Mois / Année	00/01	01/02	02/03	03/04	04/05	05/06	06/07	07/08	08/09	09/10	10/11	11/12
Sep	0.001	0.002	0.003	0	0	0	0	0.002	0	0.002	0.051	0.029
Oct	0.001	0.002	0.003	0	0	0	0	0.002	0.002	0.003	0.029	0.06
Nov	0.001	0.002	0.003	0	0	0	0	0.002	0	0.003	0.029	0.03
Dec	0.001	0.002	0.003	0	0	0.004	0	0.002	0	0.001	0.028	0.031
Jan	0.001	0.002	0.003	0	0	0	0	0.002	0	0.003	0.029	0.027
Feb	0.001	0.002	0.003	0	0	0	0	0.002	0	0.003	0.057	0.028
Mar	0.002	0.003	0.003	0	0	0	0	0.003	0	0.003	0.029	0.036
Apr	0.001	0.003	0.003	0	0	0	0	0.003	0	0.002	0.03	0.027
May	0.001	0.003	0.003	0	0	0	0	0.003	0	0.003	0.029	0.03
Jun	0.001	0.003	0.003	0	0	0	0	0.003	0	0.004	0.029	0.005
Jul	0.001	0.003	0.003	0	0	0	0	0.003	0.004	0.003	0.026	0
Aug	0.002	0.003	0.003	0	0	0	0	0.003	0	0.003	0.028	0
Total	0.014	0.03	0.036	0	0	0.004	0	0.03	0.006	0.033	0.394	0.303

TABLE 5 – Coefficients culturels ( $K_c$ ) des cultures (Bensalem and Bouhenna 2017)

Mois	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Céréales	0.45	0.8	0.9	0.9	0.9	0.9	0.9	0.9	0	0	0	0
Maraichère	0.7	0.7	0.7	0.7	0.65	0.65	0.7	0.7	0.7	0.7	0.7	0.7
Olivier	0.2	0	0	0	0	0	0	0.6	0.8	0.8	0.6	0.5
Arbres fruitiers	0.9	0	0	0	0	0	0	0.6	0.6	0.6	0.8	0.8

TABLE 6 – Pluies de fréquence (80%), pluies utiles, données et résultats de calcul de  $ET_0$  (Bensalem and Bouhenna 2017)

Mois	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
P (80%) (mm)	4.7	8.9	19.4	14.8	15.3	14.8	11.8	9	4.6	0	0	0
Pu (mm)	0	7.12	15.52	11.84	12.24	11.84	9.44	7.2	0	0	0	0
Tmin °C	4,1	4,6	6,3	8,4	12	15,4	19,7	20,3	16,6	13,1	8	4,9
Tmax °C	16,6	17,4	20	22,7	26,3	31,8	35	35,2	29,9	26,9	19,9	17
Humidité %	73	72	70	66	61	53	49	52	61	66	73	74
Vent (Km/j)	224	190	241	198	247	226	207	203	188	168	181	165
Insolation (h)	6,6	7,2	7,8	8,5	9,4	16,3	15,5	19,4	8,8	11,6	6,5	6,5
Ray (Mj/cm2/j)	11	13,8	17,4	20,9	23,6	34,1	32,6	36,7	19,5	19,6	11,3	10,2
ETP (mm/mois)	55,49	59,64	94,86	115,8	157,48	221,7	242,42	256,37	141,9	116,25	62,4	47,74

TABLE 7 – Besoins nets, bruts et volumes mensuels de l'eau du périmètre irrigué de Maghnia (Surface totale = 7000 ha)

Mois	Jan	Fev	Mars	Avril	Mai	Juin	Juil	Aout	Sept	Oct	Nov	Déc	Total <sup>c</sup>
$B_{net}$ - Céréales ( $\times 10^3 m^3/ha$ )	0.377	0.418	0.759	0.970	0	0	0	0	0	0.859	0.335	0.311	4.03
$B_{net}$ - Maraichère ( $\times 10^3 m^3/ha$ )	0.238	0.269	0.570	0.739	1.102	1.552	1.697	1.795	0.993	0.743	0.210	0.216	10.12
$B_{net}$ - Olivier ( $\times 10^3 m^3/ha$ )	0	0	0	0.288	0.945	1.330	1.939	2.051	1.277	0	0	0	7.83
$B_{net}$ - Arbres fruitiers ( $\times 10^3 m^3/ha$ )	0	0	0	0.172	1.260	1.774	1.455	1.282	0.284	0	0	0	6.23
$B_{brut}^a$ - Céréales ( $\times 10^3 m^3/ha$ )	0.509	0.565	1.026	1.311	0	0	0	0	0	1.161	0.453	0.421	5.45
$B_{brut}^a$ - Maraichère ( $\times 10^3 m^3/ha$ )	0.322	0.364	0.770	0.998	1.490	2.097	2.293	2.425	1.342	1.003	0.284	0.292	13.68
$B_{brut}^a$ - Olivier ( $\times 10^3 m^3/ha$ )	0	0	0	0.389	1.277	1.798	2.621	2.772	1.726	0	0	0	10.58
$B_{brut}^a$ - Arbres fruitiers ( $\times 10^3 m^3/ha$ )	0	0	0	0.232	1.702	2.397	1.966	1.732	0.384	0	0	0	8.41
Volume d'eau <sup>b</sup> - Céréales- ( $Mm^3$ )	2.140	2.375	4.310	5.507	0	0	0	0	0	4.874	1.902	1.767	22.87
Volume d'eau <sup>b</sup> - Maraichère ( $Mm^3$ )	0.451	0.509	1.078	1.397	2.086	2.936	3.210	3.395	1.879	1.405	0.398	0.408	19.15
Volume d'eau <sup>b</sup> - Olivier ( $Mm^3$ )	0	0	0	0.408	1.341	1.887	2.752	2.910	1.812	0	0	0	11.11
Volume d'eau <sup>b</sup> - Arbres fruitiers ( $Mm^3$ )	0	0	0	0.081	0.596	0.839	0.688	0.606	0.134	0	0	0	2.94
Volume d'eau total mensuel ( $Mm^3$ )	2.591	2.884	5.387	7.393	4.022	5.662	6.650	6.912	3.826	6.279	2.301	2.175	56.08
<b>Volume total (besoins bruts) (<math>Mm^3/an</math>)</b>													<b>56.08</b>

$$B_{brut} = \frac{B_{net}}{R_G}$$

Volume brut =  $B_{brut} \times$  Surface de culture considérées.

Total annuel.

TABLE 8 – Résultats d'entraînement et de validation des réseaux de neurones dotés d'une seule couche cachée

Algo. appreni.	Nombre d'entrées	Meilleur RNA	Indices "Phase d'apprentissage"						Indices "Phase validation"					
			<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>	<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>
BP	1	ANN-BP-1 (15)	0.251	3.309	0.079	0.298	0.089	8.657	0.422	2.863	0.115	0.344	0.118	4.936
	2	ANN-PB-2 (1)	0.236	3.440	0.067	0.260	0.068	8.725	0.389	3.136	0.069	0.300	0.090	5.076
	3	<b>ANN-PB-3 (2)</b>	<b>0.275</b>	<b>3.273</b>	<b>0.085</b>	<b>0.297</b>	<b>0.088</b>	<b>8.650</b>	<b>0.514</b>	<b>2.947</b>	<b>0.154</b>	<b>0.407</b>	<b>0.165</b>	<b>4.852</b>
	4	ANN-PB-4 (2)	0.317	3.118	0.101	0.321	0.103	8.351	0.524	2.953	0.118	0.378	0.143	4.972
	5	ANN-PB-5 (2)	0.272	3.215	0.084	0.292	0.085	8.438	0.377	3.043	0.090	0.312	0.098	5.064
	6	ANN-PB-6 (2)	0.345	3.066	0.112	0.338	0.114	8.299	0.516	2.985	0.059	0.342	0.117	5.163
LM	1	ANN-LM-1 (2)	0.397	3.461	0.127	0.357	0.128	8.426	0.466	2.885	0.035	0.304	0.092	5.156
	2	ANN-LM-2 (1)	0.292	3.452	0.079	0.281	0.079	8.669	0.398	3.166	0.023	0.246	0.060	5.201
	3	<b>ANN-LM-3 (10)</b>	<b>0.308</b>	<b>3.420</b>	<b>0.088</b>	<b>0.298</b>	<b>0.089</b>	<b>8.635</b>	<b>0.482</b>	<b>3.107</b>	<b>0.085</b>	<b>0.338</b>	<b>0.114</b>	<b>5.046</b>
	4	ANN-LM-4 (5)	0.574	2.838	0.248	0.514	0.265	7.637	0.596	3.437	-1.187	0.443	0.196	7.828
	5	ANN-LM-5 (8)	0.946	2.524	0.792	0.917	0.842	4.021	0.266	6.183	-7.410	0.152	0.023	15.392
	6	ANN-LM-6 (3)	0.378	3.255	0.108	0.329	0.108	8.315	0.520	3.010	0.042	0.331	0.110	5.208
RGA	1	ANN-RGA-1 (5)	0.301	3.442	0.088	0.297	0.088	8.615	0.422	3.096	0.047	0.280	0.078	5.122
	2	ANN-RGA-2 (1)	0.293	3.455	0.079	0.281	0.079	8.669	0.399	3.167	0.023	0.247	0.061	5.201
	3	ANN-RGA-3 (8)	0.386	3.412	0.130	0.361	0.130	8.438	0.466	3.216	0.051	0.303	0.092	5.141
	4	<b>ANN-RGA-4 (14)</b>	<b>0.384</b>	<b>3.216</b>	<b>0.127</b>	<b>0.358</b>	<b>0.128</b>	<b>8.226</b>	<b>0.509</b>	<b>2.908</b>	<b>0.124</b>	<b>0.368</b>	<b>0.135</b>	<b>4.955</b>
	5	ANN-RGA-5 (4)	0.377	3.235	0.119	0.346	0.120	8.274	0.528	3.191	0.027	0.341	0.116	5.236
	6	ANN-RGA-6 (6)	0.362	3.248	0.121	0.350	0.123	8.254	0.493	3.084	0.077	0.337	0.114	5.114

Résultats d'entraînement et de validation des réseaux de neurones dotés d'une seule couche cachée (suite)

Algorithme apprentissage	Nombre d'entrées	Meilleur RNA	Indices "Phase d'apprentissage"						Indices "Phase de validation"					
			<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>	<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>
HPSOGWO	1	ANN-HPG <sup>a</sup> 1 (5)	0.117	3.621	0.015	0.136	0.019	8.951	0.310	3.115	0.089	0.361	0.130	5.009
	2	ANN-HPG-2 (9)	0.129	3.641	0.023	0.154	0.024	8.929	0.340	3.141	0.097	0.366	0.134	5.001
	3	ANN-HPG-3 (6)	<b>0.121</b>	<b>3.603</b>	<b>0.026</b>	<b>0.161</b>	<b>0.026</b>	<b>8.926</b>	<b>0.312</b>	<b>3.092</b>	<b>0.101</b>	<b>0.384</b>	<b>0.148</b>	<b>5.002</b>
	4	ANN-HPG-4 (4)	0.121	3.443	0.027	0.166	0.027	8.684	0.277	3.068	0.090	0.360	0.130	5.050
	5	ANN-HPG-5 (7)	0.120	3.705	0.023	0.160	0.026	8.713	0.338	3.373	0.075	0.368	0.135	5.105
	6	ANN-HPG-6 (7)	0.134	3.317	0.029	0.171	0.029	8.678	0.293	2.938	0.104	0.361	0.130	5.036

<sup>a</sup> Abréviation de "HPSOGWO".

TABLE 9 – Résultats d'entraînement et de validation des réseaux de neurones dotés de deux (02) couches cachées

Algo. apprenti.	Apports "entrés"	Meilleur RNA	Indices "Phase d'apprentissage"						Indices "Phase validation"					
			<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>	<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>
BP	1	ANN-BP-1 (3, 13)	0.143	3.293	0.038	0.221	0.049	8.847	0.289	2.837	0.099	0.340	0.116	4.982
	2	ANN-BP-2 (1, 14)	0.175	3.340	0.049	0.233	0.054	8.806	0.340	2.939	0.098	0.327	0.107	4.997
	<b>3</b>	<b>ANN-BP-3 (2, 7)</b>	<b>0.323</b>	<b>3.051</b>	<b>0.100</b>	<b>0.328</b>	<b>0.108</b>	<b>8.579</b>	<b>0.602</b>	<b>2.690</b>	<b>0.149</b>	<b>0.435</b>	<b>0.190</b>	<b>4.868</b>
	4	ANN-BP-4 (5, 13)	0.322	2.885	0.099	0.342	0.117	8.357	0.542	2.595	0.141	0.397	0.158	4.906
	5	ANN-BP-5 (5, 13)	0.320	2.911	0.099	0.331	0.109	8.368	0.517	2.774	0.115	0.366	0.134	4.993
	6	ANN-BP-6 (7, 13)	0.320	2.913	0.100	0.335	0.112	8.354	0.498	2.728	0.117	0.359	0.129	5.000
LM	1	ANN-LM-1(7,6)	0.094	3.490	0.024	0.176	0.031	8.911	0.226	3.045	0.068	0.338	0.114	5.066
	2	ANN-LM-2(8,8)	0.115	3.376	0.027	0.194	0.038	8.909	0.220	2.888	0.076	0.326	0.106	5.058
	<b>3</b>	<b>ANN-LM-3 (6, 15)</b>	<b>0.893</b>	<b>2.675</b>	<b>0.671</b>	<b>0.820</b>	<b>0.673</b>	<b>5.190</b>	<b>0.591</b>	<b>2.940</b>	<b>0.161</b>	<b>0.438</b>	<b>0.192</b>	<b>4.832</b>
	4	ANN-LM-4(14,14)	0.851	2.935	0.571	0.757	0.573	5.769	0.367	2.685	0.056	0.266	0.071	5.142
	5	ANN-LM-5(3,6)	0.695	2.938	0.361	0.601	0.361	7.050	0.583	3.648	-1.285	0.429	0.184	8.023
	6	ANN-LM-6(3,13)	0.639	2.882	0.307	0.554	0.307	7.332	0.672	2.808	-0.100	0.487	0.237	5.581
RGA	1	ANN-RGA-1(9,3)	0.325	3.447	0.099	0.315	0.099	8.565	0.467	3.114	0.077	0.328	0.108	5.041
	2	ANN-RGA-2(5,14)	0.325	3.505	0.102	0.321	0.103	8.560	0.429	3.245	0.041	0.288	0.083	5.153
	3	ANN-RGA-3(3,14)	0.352	3.406	0.103	0.322	0.104	8.564	0.564	3.090	0.105	0.394	0.155	4.992
	4	ANN-RGA-4(6,1)	0.310	3.374	0.099	0.319	0.102	8.358	0.499	3.003	0.152	0.407	0.166	4.874
	5	ANN-RGA-5(3,14)	0.568	3.083	0.232	0.482	0.232	7.727	0.560	2.897	0.106	0.385	0.148	5.019
	<b>6</b>	<b>ANN-RGA-6(9,9)</b>	<b>0.604</b>	<b>3.057</b>	<b>0.276</b>	<b>0.526</b>	<b>0.277</b>	<b>7.491</b>	<b>0.530</b>	<b>2.751</b>	<b>0.178</b>	<b>0.422</b>	<b>0.178</b>	<b>4.825</b>

Résultats d'entraînement et de validation des réseaux de neurones dotés de deux (02) couches cachées (suite)

Algo. apprenti.	Nombre d'entrées	Meilleur RNA	Indices "Phase d'apprentissage"						Indices "Phase validation"					
			<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>	<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>
HPSOGWO	1	ANN-HPG <sup>a</sup> 1(5,15)	0.138	3.800	0.017	0.150	0.023	8.942	0.369	3.332	0.081	0.366	0.134	5.030
	2	ANN-HPG-2(9,5)	0.154	3.494	0.025	0.166	0.027	8.917	0.384	2.936	0.122	0.368	0.135	4.930
	<b>3</b>	<b>ANN-HPG-3(12,15)</b>	<b>0.160</b>	<b>3.573</b>	<b>0.040</b>	<b>0.200</b>	<b>0.040</b>	<b>8.863</b>	<b>0.441</b>	<b>3.128</b>	<b>0.145</b>	<b>0.419</b>	<b>0.176</b>	<b>4.878</b>
	4	ANN-HPG-4(3,15)	0.195	3.653	-0.008	0.133	0.018	8.839	0.476	2.906	0.147	0.387	0.150	4.889
	5	ANN-HPG-5(2,2)	0.150	3.407	0.030	0.175	0.031	8.685	0.336	2.964	0.113	0.363	0.132	4.997
	6	ANN-HPG-6(11,8)	0.147	3.481	0.027	0.166	0.027	8.686	0.327	3.007	0.112	0.379	0.144	5.014

<sup>a</sup> Abréviation de "HPSOGWO".

TABLE 13 – Résultats d'entraînement et de validation des réseaux de neurones artificiels (Optimisation par HPSOGWO, apprentissage par BP et LM)

Algorithme d'optimisation	EANN	Phase d'apprentissage						Phase de validation					
		<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>	<i>d</i>	<i>MAE</i>	<i>Nash</i>	<i>R</i>	<i>R</i> <sup>2</sup>	<i>RMSE</i>
HPSOGWO	EANN-HPG <sup>a</sup> BP-1 (5,13)	0.694	2.937	0.383	0.628	0.394	7.085	0.213	3.090	-0.849	0.066	0.004	7.135
	EANN-HPG-BP-2 (7,12)	0.826	3.181	0.558	0.775	0.601	6.008	0.329	3.168	-0.316	0.055	0.003	6.036
	EANN-HPG-BP-3 (7,13)	0.689	3.168	0.386	0.637	0.406	7.088	0.198	3.176	-0.351	0.016	0.000	6.132
	EANN-HPG-BP-4 (6,11)	0.863	2.465	0.634	0.820	0.673	5.327	0.209	2.869	-0.244	-0.031	0.001	5.902
	EANN-HPG-BP-5 (4,6)	0.833	2.728	0.579	0.769	0.592	5.722	0.152	3.262	-0.921	0.009	0.000	7.357
	<b>EANN-HPG-BP-6 (6,9)</b>	<b>0.936</b>	<b>2.519</b>	<b>0.767</b>	<b>0.886</b>	<b>0.786</b>	<b>4.250</b>	<b>0.407</b>	<b>3.730</b>	<b>-0.595</b>	<b>0.171</b>	<b>0.029</b>	<b>6.720</b>
	EANN-HPG-LM-1 (5,11)	0.938	2.325	0.795	0.892	0.796	4.085	0.307	4.783	-5.087	0.229	0.052	12.946
	EANN-HPG-LM-2 (4,9)	0.952	2.175	0.831	0.912	0.832	3.717	0.145	4.095	-2.310	0.009	0.000	9.573
	EANN-HPG-LM-3 (8,7)	0.960	2.071	0.861	0.930	0.866	3.368	0.177	4.631	-3.889	0.041	0.002	11.667
	<b>EANN-HPG-LM-4 (8,13)</b>	<b>0.974</b>	<b>1.635</b>	<b>0.903</b>	<b>0.951</b>	<b>0.905</b>	<b>2.740</b>	<b>0.524</b>	<b>4.392</b>	<b>-2.109</b>	<b>0.377</b>	<b>0.142</b>	<b>9.333</b>
	EANN-HPG-LM-5 (6,12)	0.959	1.989	0.852	0.924	0.853	3.389	0.111	3.713	-2.706	0.006	0.000	10.217
	EANN-HPG-LM-6 (9,6)	0.967	1.608	0.878	0.938	0.879	3.075	0.457	4.218	-3.122	0.356	0.127	10.804

<sup>a</sup> HPG : Abréviation de "HPSOGWO".

TABLE 10 – Résultats de recherche des EANNs (Optimisation par GSA, Apprentissage par BP et LM)

Paramètre	EAN-GSA-BP-1	EAN-GSA-BP-2	EAN-GSA-BP-3	EAN-GSA-BP-4	EAN-GSA-BP-5	<b>EAN-GSA-BP-6</b>
Fonction d'activation	Sigmoïde	Sigmoïde	Sigmoïde	Sigmoïde	Sigmoïde	<b>Sigmoïde</b>
$\alpha_{activ}$	2.057	2.141	2.203	2.062	2.375	<b>3.387</b>
Taux d'apprentissage $\mu$	0.535	0.639	0.734	0.785	0.278	<b>0.479</b>
Erreur d'apprentissage $E_{app}$	0.494	0.610	0.576	0.386	0.735	<b>0.615</b>
Nbr. d'itérations $K_{max-app}$	19279	16152	19300	13832	19047	<b>18442</b>
Nbr de couches cachées $N_c$	2	2	2	2	2	<b>2</b>
Nbr neurones - couche 1	7	5	5	9	6	<b>7</b>
Nbr neurones - couche 2	12	6	11	10	4	<b>11</b>
Nbr neurones - couche 3	0	0	0	0	0	<b>0</b>
Nbr neurones - couche 4	0	0	0	0	0	<b>0</b>
Nbr neurones - couche 5	0	0	0	0	0	<b>0</b>

Paramètre	EAN-GSA-LM-1	<b>EAN-GSA-LM-2</b>	EAN-GSA-LM-3	EAN-GSA-LM-4	EAN-GSA-LM-5	EAN-GSA-LM-6
Fonction d'activation	Sigmoïde	<b>Sigmoïde</b>	Sigmoïde	Sigmoïde	Sigmoïde	Sigmoïde
$\alpha_{activ}$	1.348	<b>1.849</b>	1.043	0.701	0.218	1.312
Taux d'apprentissage $\mu$	0.525	<b>0.543</b>	0.729	0.586	0.636	0.671
Erreur d'apprentissage $E_{app}$	0.385	<b>0.032</b>	0.409	0.429	0.258	0.039
Nbr. d'itérations $K_{max-app}$	49	<b>49</b>	50	61	53	54
Nbr de couches cachées $N_c$	2	<b>2</b>	1	3	2	2
Nbr neurones - couche 1	5	<b>7</b>	5	6	9	7
Nbr neurones - couche 2	3	<b>8</b>	0	5	5	10
Nbr neurones - couche 3	0	<b>0</b>	0	2	0	0
Nbr neurones - couche 4	0	<b>0</b>	0	0	0	0
Nbr neurones - couche 5	0	<b>0</b>	0	0	0	0

TABLE 11 – Résultats de recherche des EANNs (Optimisation par HPSOGWO, apprentissage par BP et LM)

Paramètres	EAN-HPG-BP-1	EAN-HPG-BP-2	EAN-HPG-BP-3	EAN-HPG-BP-4	EAN-HPG-BP-5	<b>EAN-HPG-BP-6</b>
Fonction d'activation	Sigmoïde	Sigmoïde	Sigmoïde	Sigmoïde	Sigmoïde	<b>Sigmoïde</b>
$\alpha_{activ}$	1.466	2.088	1.980	2.094	1.266	<b>1.538</b>
Taux d'apprentissage $\mu$	0.803	0.713	0.584	0.716	0.878	<b>0.761</b>
Erreur d'apprentissage $E_{app}$	0.733	0.644	0.317	0.322	0.478	<b>0.270</b>
Nbr. d'itérations $K_{max-app}$	19153	19830	19208	19704	18353	<b>18653</b>
Nbr de couches cachées $N_c$	2	2	2	2	2	<b>2</b>
Nbr neurones - couche 1	5	7	7	6	4	<b>6</b>
Nbr neurones - couche 2	13	12	13	11	6	<b>9</b>
Nbr neurones - couche 3	0	0	0	0	0	<b>0</b>
Nbr neurones - couche 4	0	0	0	0	0	<b>0</b>
Nbr neurones - couche 5	0	0	0	0	0	<b>0</b>
Paramètres	EAN-HPG-LM-1	EAN-HPG-LM-2	EAN-HPG-LM-3	<b>EAN-HPG-LM-4</b>	EAN-HPG-LM-5	EAN-HPG-LM-6
Fonction d'activation	Sigmoïde	Sigmoïde	Sigmoïde	<b>Sigmoïde</b>	Sigmoïde	Sigmoïde
$\alpha_{activ}$	1.414	1.160	0.824	<b>1.117</b>	2.420	0.464
Taux d'apprentissage $\mu$	0.622	0.653	0.303	<b>0.448</b>	0.289	0.365
Erreur d'apprentissage $E_{app}$	0.284	0.193	0.068	<b>0.087</b>	0.110	0.113
Nombre d'itérations	69	68	68	<b>64</b>	65	60
$K_{max-app}$						
Nbr de couches cachées $N_c$	2	2	2	<b>2</b>	2	2
Nbr neurones - couche 1	5	4	8	<b>8</b>	6	9
Nbr neurones - couche 2	11	9	7	<b>13</b>	12	6
Nbr neurones - couche 3	0	0	0	<b>0</b>	0	0
Nbr neurones - couche 4	0	0	0	<b>0</b>	0	0
Nbr neurones - couche 5	0	0	0	<b>0</b>	0	0

TABLE 12 – Résultats d’entraînement et de validation des réseaux de neurones artificiels (Optimisation par GSA, apprentissage par BP et LM)

Algorithme d’optimisation	EANN	Phase d’apprentissage						Phase de validation					
		$d$	$MAE$	$Nash$	$R$	$R^2$	$RMSE$	$d$	$MAE$	$Nash$	$R$	$R^2$	$RMSE$
GSA	EANN-GSA-BP-1 (7,12)	0.706	2.834	0.394	0.644	0.415	7.020	0.200	3.090	-0.904	0.066	0.004	7.241
	EANN-GSA-BP-2 (5,6)	0.733	2.935	0.400	0.646	0.417	6.994	0.224	3.475	-1.073	0.078	0.006	7.575
	EANN-GSA-BP-3 (5,11)	0.844	2.687	0.599	0.803	0.645	5.725	0.283	3.279	-0.737	0.079	0.006	6.955
	EANN-GSA-BP-4 (9,10)	0.836	2.572	0.594	0.808	0.653	5.612	0.199	2.943	-0.724	0.003	0.000	6.949
	EANN-GSA-BP-5 (6,4)	0.615	3.511	0.206	0.480	0.231	7.855	0.081	3.335	-0.434	-0.006	0.000	6.355
	<b>EANN-GSA-BP-6 (7,11)</b>	<b>0.836</b>	<b>3.553</b>	<b>0.478</b>	<b>0.757</b>	<b>0.572</b>	<b>6.363</b>	<b>0.496</b>	<b>3.572</b>	<b>-0.496</b>	<b>0.228</b>	<b>0.052</b>	<b>6.509</b>
	EANN-GSA-LM-1 (5,3)	0.886	2.577	0.664	0.815	0.664	5.233	0.221	3.746	-2.596	0.107	0.012	9.951
	<b>EANN-GSA-LM-2 (7,8)</b>	<b>0.863</b>	<b>2.830</b>	<b>0.629</b>	<b>0.798</b>	<b>0.637</b>	<b>5.504</b>	<b>0.349</b>	<b>3.195</b>	<b>-0.143</b>	<b>0.122</b>	<b>0.015</b>	<b>5.624</b>
	EANN-GSA-LM-3 (5)	0.910	2.893	0.715	0.854	0.730	4.828	0.251	6.516	-10.400	0.224	0.050	17.815
	EANN-GSA-LM-4 (6,5,2)	0.946	2.028	0.813	0.902	0.813	3.811	0.201	3.132	-0.240	0.010	0.000	5.895
	EANN-GSA-LM-5 (9,5)	0.941	2.063	0.804	0.897	0.805	3.903	0.185	3.398	-2.290	0.047	0.002	9.628
	EANN-GSA-LM-6 (7,10)	0.968	1.688	0.881	0.939	0.881	3.042	0.312	5.480	-7.413	0.285	0.081	15.435

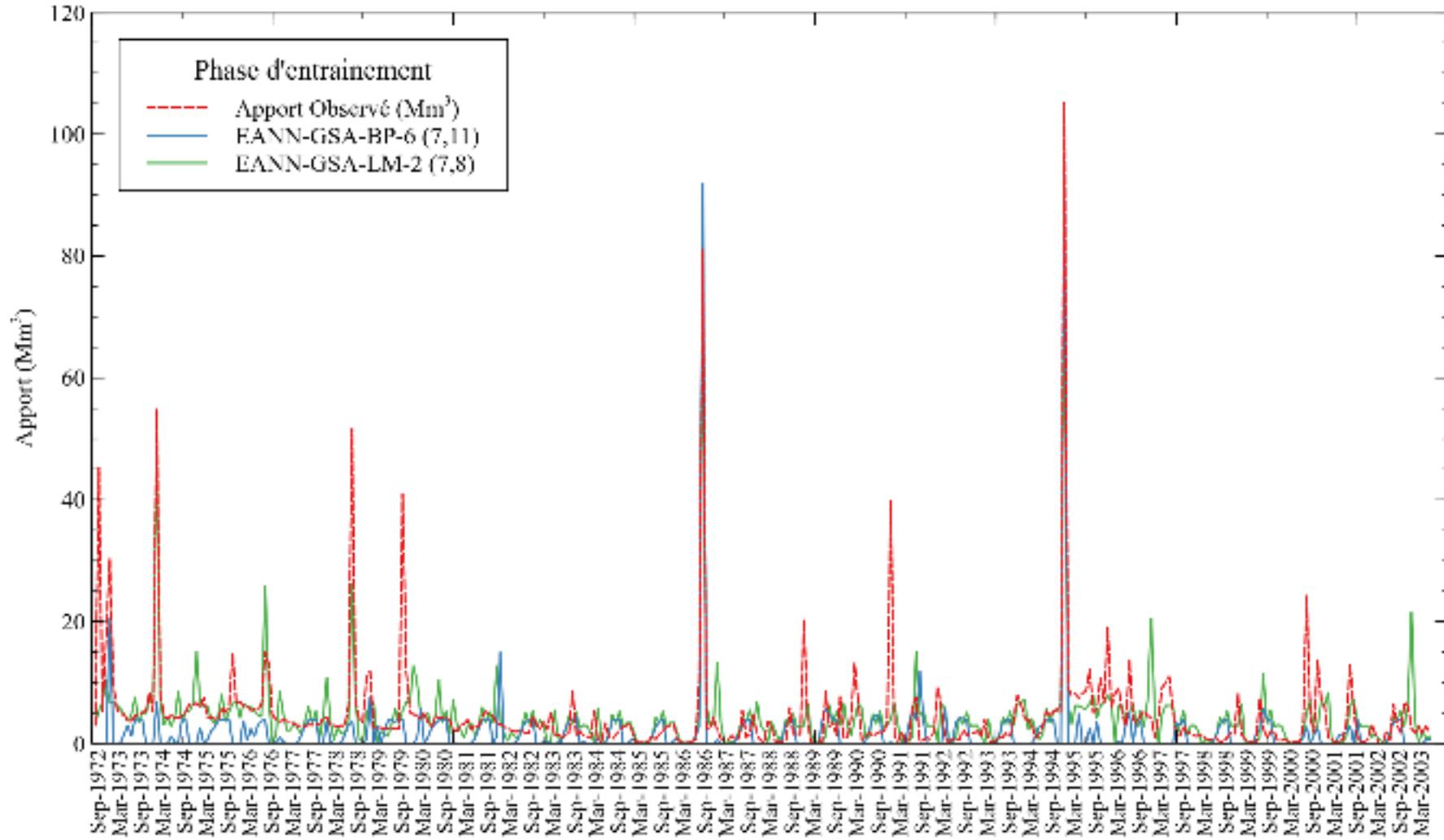


FIGURE 7 – Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase d'apprentissage, optimisation par l'algorithme GSA)

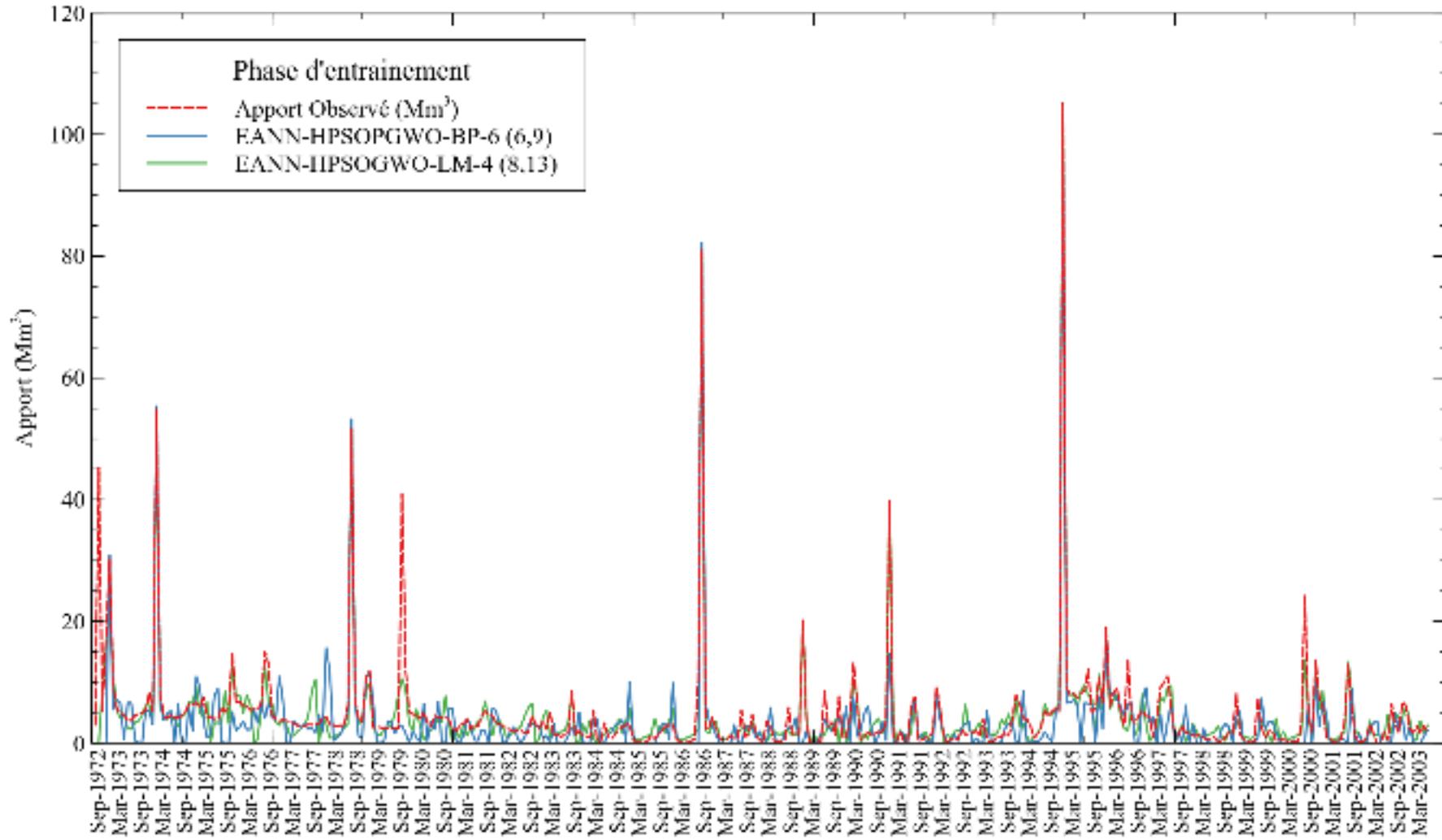


FIGURE 8 – Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase d'apprentissage, optimisation par HPSOGWO)

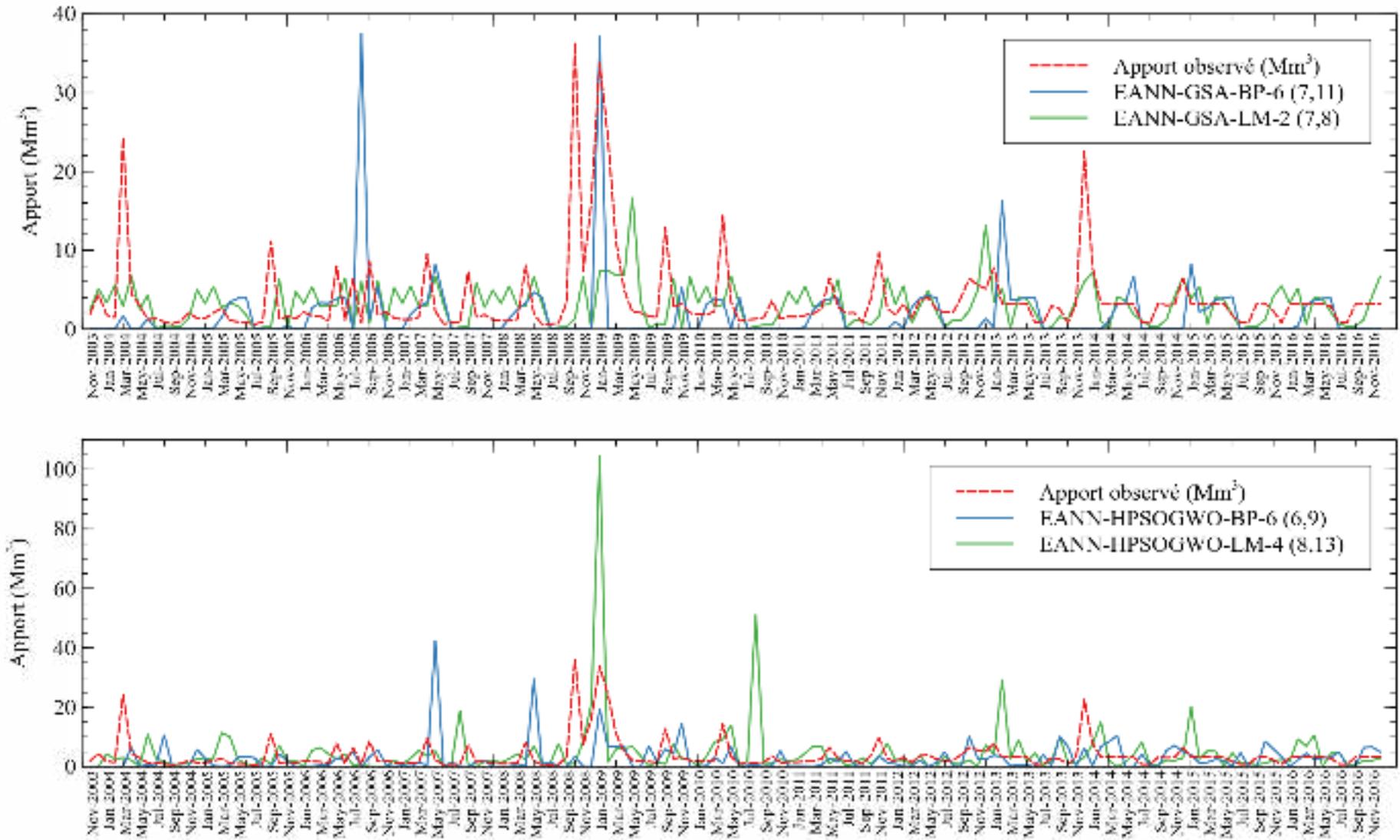


FIGURE 9 – Apports observés et ceux calculés par les réseaux de neurones proches-optimaux (Phase du test)

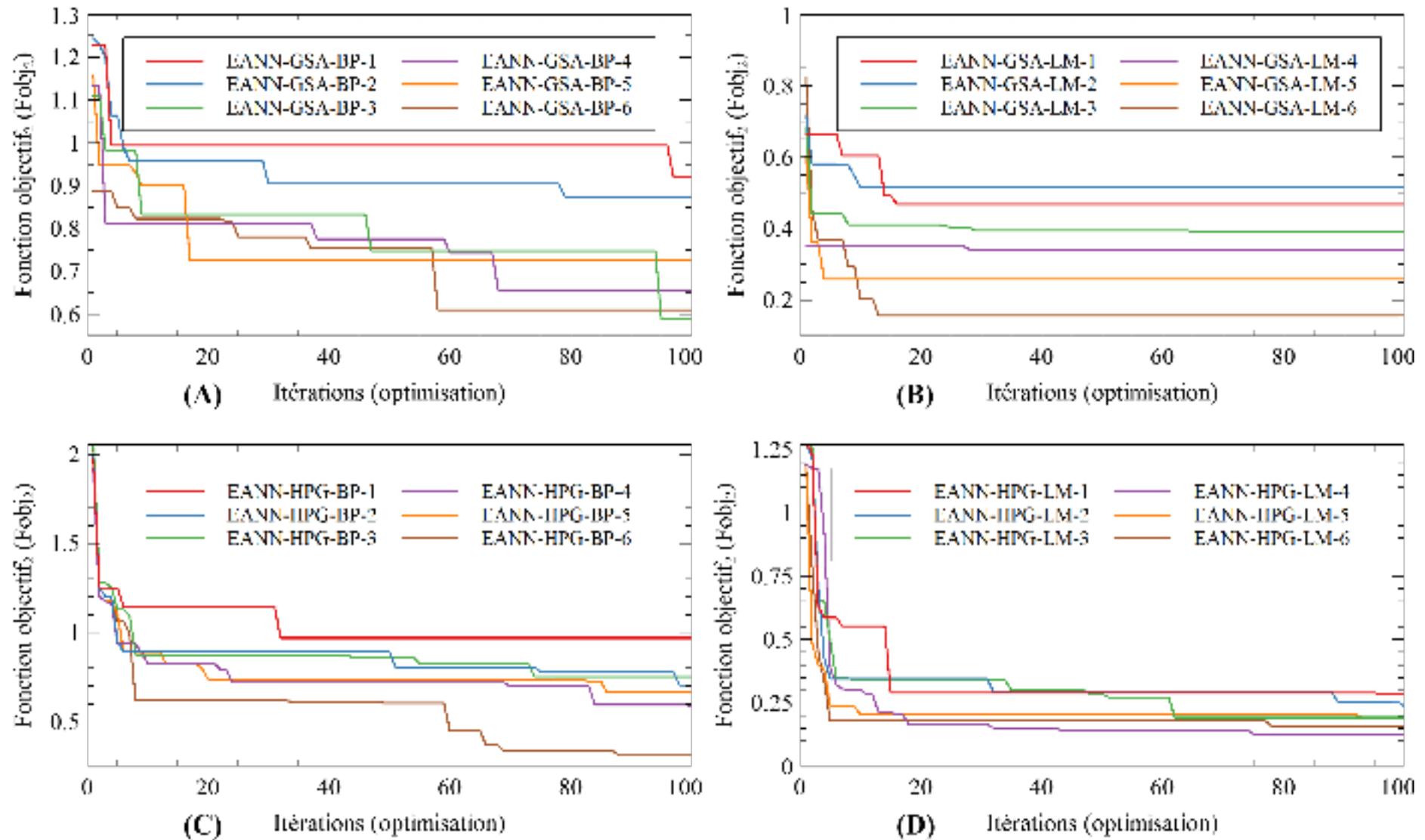


FIGURE 10 – Évolution de la valeur de la fonction objectif (équation (4.1)) en fonction de nombre d'itérations pour les différents modèles

TABLE 14 – Performance et stabilité des algorithmes d'optimisation RGA, GSA, HPSOGWO et IGWO (scénarios 1 et 2)

Scénario	Essai	RGA		GSA		HPSOGWO		IGWO	
		$F_{obj}^a$	Temps <sup>b</sup>	$F_{obj}$	Temps <sup>c</sup>	$F_{obj}$	Temps <sup>b</sup>	$F_{obj}$	Temps <sup>b</sup>
1	1	5.08	26 :29	5.06	1 :03 :51	5.06	14 :52	8.18	16 :26
	2	5.08	27 :11	5.06	1 :03 :02	5.06	17 :46	7.42	12 :06
	3	5.09	27 :41	5.06	1 :10 :21	5.06	18 :11	7.22	16 :31
	4	5.08	29 :46	5.06	1 :04 :39	5.06	9 :51	8.18	10 :58
	5	5.09	28 :47	5.06	1 :06 :28	5.06	10 :41	7.42	15 :01
	Moyenne	5.09	27 :59	5.06	1 :05 :40	5.06	14 :16	7.68	14 :13
	Écart-type	4.38E-03	-	1.77E-13	-	1.27E-03	-	0.46	-
Coefficient de variation	8.61E-04	-	3.49E-14	-	2.50E-04	-	6.00E-02	-	
2	1	1.09	27 :50	1.07	59 :04	1.07	12 :25	6.05	12 :30
	2	1.09	28 :10	1.07	61 :36	1.07	13 :17	6.67	15 :26
	3	1.09	27 :19	1.07	56 :20	2.25	12 :24	7.52	17 :40
	4	1.09	31 :14	1.07	58 :58	1.07	17 :47	8.29	11 :13
	5	1.09	29 :38	1.07	59 :59	1.07	14 :58	6.05	15 :13
	Moyenne	1.09	28 :50	1.07	59 :11	1.30	14 :10	6.92	14 :25
	Écart-type	2.68E-03	-	9.34E-07	-	0.53	-	0.98	-
Coefficient de variation	2.46E-03	-	8.75E-07	-	4.05E-01	-	1.42E-01	-	

TABLE 15 – Performance et stabilité des algorithmes d’optimisation RGA, GSA, HPSOGWO et IGWO (scénario 3)

Scénario	Essai	RGA		GSA		HPSOGWO		IGWO	
		$F_{obj}$ <sup>a</sup>	Temps <sup>b</sup>	$F_{obj}$	Temps <sup>c</sup>	$F_{obj}$	Temps <sup>b</sup>	$F_{obj}$	Temps <sup>b</sup>
3	1	75.87	39 :42	81.48	54 :11	170.64	16 :19	246.86	11 :53
	2	75.81	33 :10	82.41	1 :21 :57	187.42	17 :02	246.40	11 :32
	3	75.84	34 :08	81.51	57 :35	170.64	17 :14	216.32	13 :31
	4	75.80	41 :36	82.21	52 :55	187.42	15 :28	247.74	12 :00
	5	75.83	38 :09	81.72	1 :02 :43	170.64	17 :31	246.40	13 :14
	Moyenne	75.83	37 :21	81.87	1 :01 :55	177.35	16 :43	240.74	12 :26
	Écart-type	2.50E-02	-	0.42	-	9.19	-	13.67	-
	Coefficient de variation	3.30E-04	-	5.13E-03	-	5.18E-02	-	5.68E-02	-

<sup>a</sup> "Best fitness" ou valeur de la fonction objectif à l’itération finale ( $k = K_{max} = 10^4$ ) correspondant à la meilleure solution (meilleur agent de recherche : "chromosome en RGA", "Masse en GSA", "loup gris en HPSOGWO et IGWO").

<sup>b</sup> Temps d’exécution des algorithmes donné en (minutes : secondes).

<sup>c</sup> Temps d’exécution des algorithmes donné en (heurs : minutes : secondes) ou (minutes : secondes).

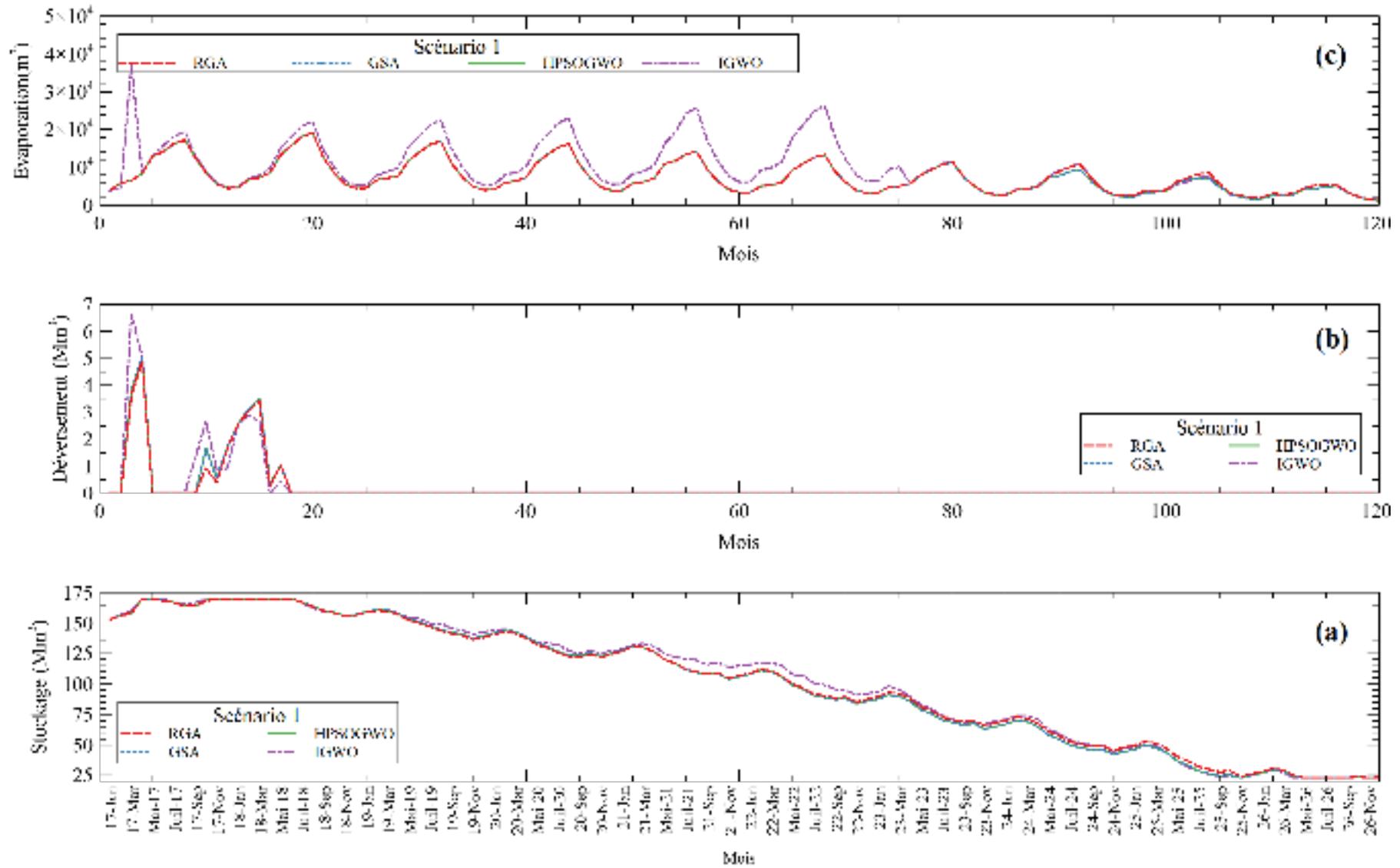


FIGURE 11 – Variation de stockage, déversement et évaporation, scénario 1

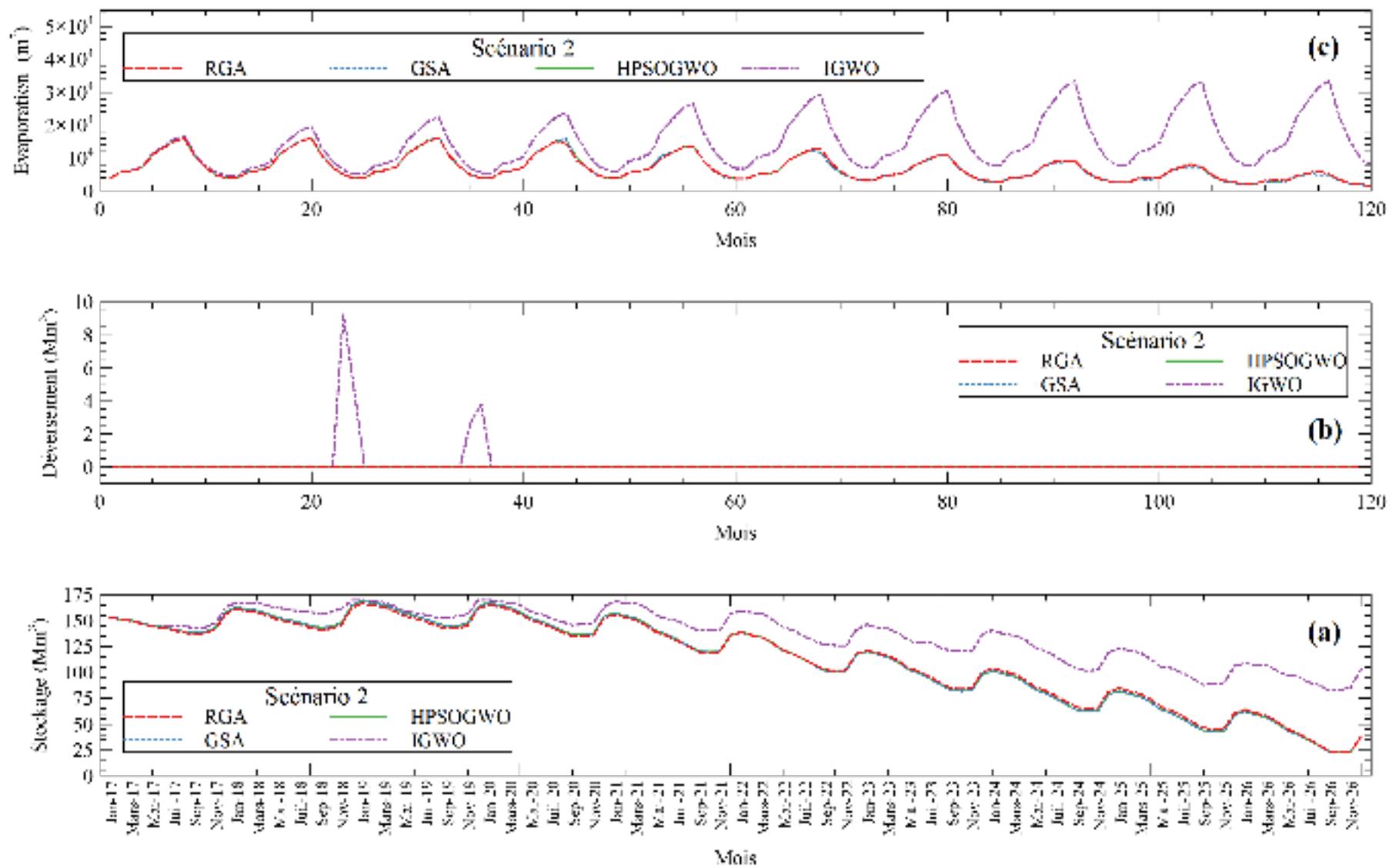


FIGURE 12 – Variation de stockage, déversement et évaporation, scénario 2

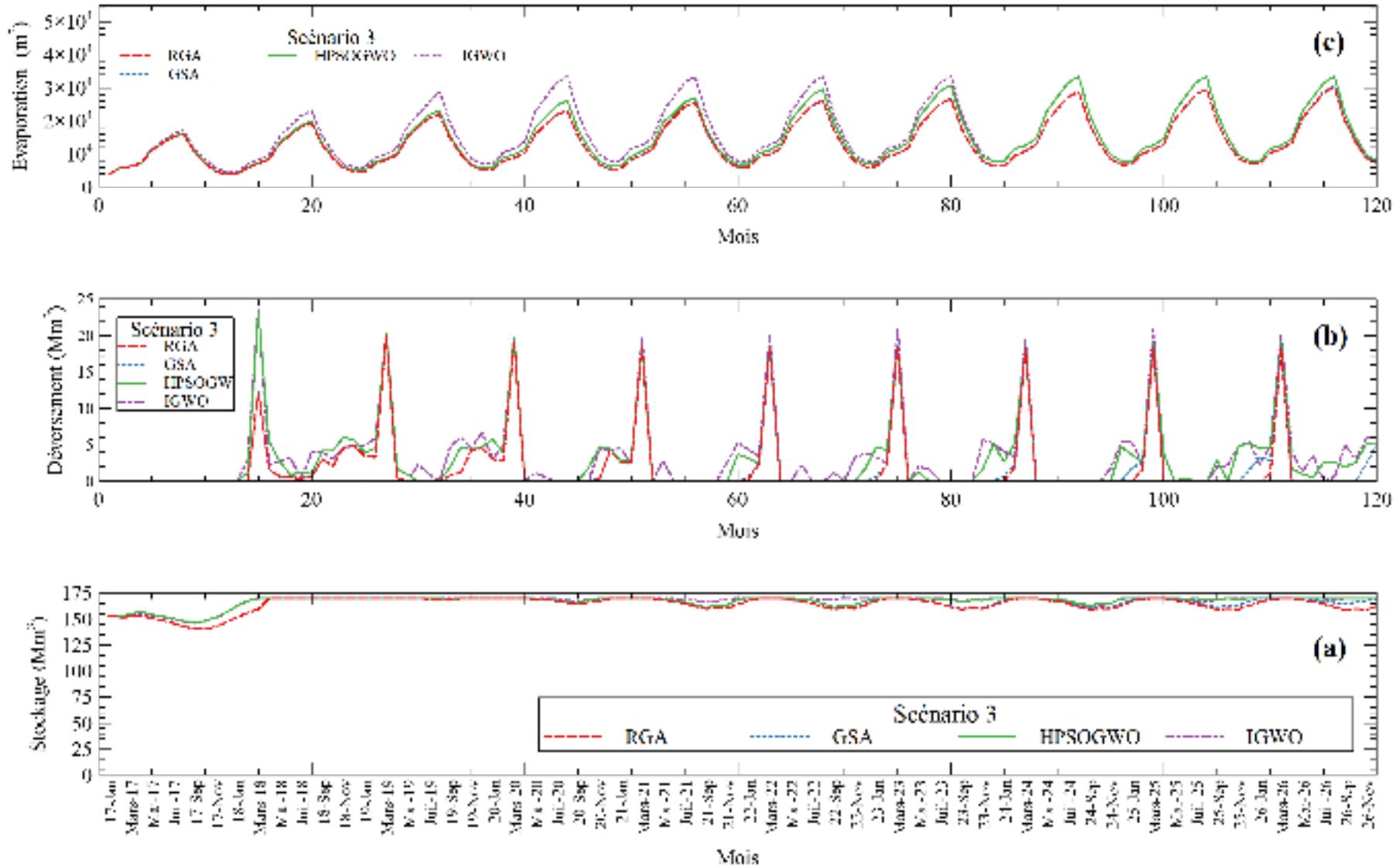


FIGURE 13 – Variation de stockage, déversement et évaporation, scénario 3

**Manuscrit 1 : Hybrid Algorithm of  
Particle Swarm Optimization and Grey  
Wolf Optimizer for Reservoir Operation  
management**



# Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Reservoir Operation Management

Saad Dahmani<sup>1</sup> · Djilali Yebdri<sup>1</sup>

Received: 24 April 2019 / Accepted: 14 October 2019 / Published online: 14 October 2020  
© Springer Nature B.V. 2020

## Abstract

Metaheuristics are highly efficient optimization methods that are widely used today. However, the performance of one method cannot be generalized and must be examined in each class of problems. The hybrid algorithm of particle swarm optimization and grey wolf optimizer (HPSOGWO) is new swarm-based metaheuristic with several advantages, such as simple implementation and low memory consumption. This study uses HPSOGWO for reservoir operation optimization. Real-coded genetic algorithm (RGA) and gravitational search algorithm (GSA) have been used as efficient methods in reservoir optimization management for comparative analysis between algorithms through two case studies. In the first case study, four benchmark functions were minimized, in which results revealed that HPSOGWO was more competitive compared with other algorithms and can produce high-quality solutions. The second case study involved minimizing the deficit between downstream demand and release from the Hammam Boughrara reservoir located in Northwest Algeria. A constrained optimization model with non-linear objective function was applied. Based on the average solutions, HPSOGWO performed better compared with RGA and was highly competitive with GSA. In addition, the reliability, resiliency, and vulnerability indices of the reservoir operation, which was derived from the three algorithms, were nearly similar to one another, which justified the usability of HPSOGWO in this field.

**Keywords** Reservoir optimization model · Evolutionary algorithm · Genetic algorithm · Gravitational search algorithm · Constrained optimization

---

✉ Saad Dahmani  
saad.dahmani@univ-usto.dz; sd.dahmani2000@gmail.com

Djilali Yebdri  
djillali.yebdri@univ-usto.dz; dyebdri@yahoo.fr

<sup>1</sup> Laboratoire de Gestion et Traitement des Eaux (LGTE), Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf (USTO-MB), BP 1505, El M'naouer, 31000 Oran, Algérie

## 1 Introduction

Optimization algorithms aim to determine values of decision variables that maximize or minimize an objective function with or without constraints. Several algorithms, such as linear programming (LP; Loucks 1968), non-linear programming (NLP; Arunkumar and Jothiprakash 2012), and stochastic dynamic programming (SDP; Stedinger et al. 1984), have been used to solve problems in reservoir optimization management for a single or a system of reservoirs. Despite the performance and wide usability of the classical methods, they suffer from several disadvantages, such as inability to solve problems with non-linear and non-convex objective functions (i.e., in the case of LP) and the curse of dimensionality (Sharif and Swamy 2014).

Evolutionary optimization algorithms (EOAs) are new methods for searching the approached solutions for optimization problems with reasonable computation time and acceptable accuracy. Several EOAs, such as genetic algorithm (GA; Chang and Chen 1998; Ahmed and Sarma 2005), particle swarm optimization (PSO; Kumar and Reddy 2007; Ghimire and Reddy 2013), gravitational search algorithm (GSA; Bozorg-Haddad et al. 2016), and weed optimization algorithm (WOA; Karami et al. 2019), have been successfully used and efficiently addressed critical issues in reservoir operation management.

EOAs have many drawbacks despite their advantages. First, no evidence demonstrates the generalizability of the performance and superiority of a metaheuristic method across applications (Wolpert et al. 1997). Second, a large part of EOAs can overlook the global optimal solution and become trapped in a local one. Third, several algorithms suffer from imbalance between exploration and exploitation capabilities during a search. Fourth, the majority of EOAs cannot detect the stochastic nature of certain variables (e.g., inflow, evaporation, and non-linear form of constraints; Karami et al. 2019). Finally, EOAs are sensitive to parameter setting. In this context, exploring other algorithms that can surpass these drawbacks is necessary.

Singh and Singh (2017) developed the hybrid algorithm of particle swarm optimization and grey wolf optimizer (HPSOGWO), which is a recent nature-inspired algorithm. It is a combination of PSO (Eberhart and Kennedy 1995) and grey wolf optimizer (GWO) (Mirjalili et al. 2014) that exploits the benefits of the two algorithms and outperform their drawbacks. During the optimization process, search agents in HPSOGWO mimic the hunting behavior of grey wolves (*Canis lupus*) via the GWO and update their positions by using the PSO approach. Similarities exist between GWO and HPSOGWO in the majority of the structure. However, HPSOGWO retains the main advantages of the GWO, such as the simple design, ease of coding, and low memory consumption (Faris et al. 2018). The authors of HPSOGWO compared it with PSO and GWO by minimizing 23 benchmark functions from the work of Yao et al. (1999). Based on statistical results, HPSOGWO displays better performance in terms of solution quality, stability, computational time, and capability to achieve the global optimum.

The water resource management field has yet to apply HPSOGWO, which may encourage researchers to explore its capabilities in this field. Thus, this study tests and compares HPSOGWO with RGA and GSA. Comparison was based on minimizing four benchmark functions and solving the reservoir operating optimization problem with a constrained non-linear objective function. The algorithms were evaluated based on the statistical results of analysis on the one hand and the reliability, vulnerability, and resiliency indices of reservoir operation on the other hand. Examining the performance of HPSOGWO in dealing with the aforementioned problems is the ultimate objective of this study.

## 2 Materials and Methods

### 2.1 Optimization Techniques

#### 2.1.1 Hybrid PSOGWO

HPSOGWO is a novel hybrid metaheuristic algorithm, which merges the functionalities of PSO with GWO to improve the exploration and exploitation abilities of GWO and PSO, respectively (Singh and Singh 2017). HPSOGWO imitates the hunting behavior of grey wolves in nature in the form of GWO. Grey wolves live in groups of 5 to 12 members with a strict hierarchy organized from top to bottom in four categories, namely, alpha ( $\alpha$ ), beta ( $\beta$ ), delta ( $\delta$ ), and omega ( $\omega$ ). The alpha is considered the dominant member of the group and is responsible for making decisions regarding the social behavior of the group especially during hunting. The beta and delta are subordinate to the alpha and can control the remainder of the group (omega). The hunting mechanism of grey wolves has specific steps as follows: (1) tracking, chasing, and approaching, (2) pursuing, encircling, and harassing the prey until it stops moving, and (3) attacking.

In the mathematical model, HPSOGWO runs in the following steps:

1.  $N$  grey wolves are defined and placed randomly in the search space as follows:

$$X_i = \{x_i^1, x_i^2, \dots, x_i^l, \dots, x_i^d\} \quad i = 1, \dots, N \quad (1)$$

where  $x_i^l$  denotes the position of the  $i$ th wolf at the  $l$ th dimension, and  $d$  refers to the search space dimension.

2. Fitness is computed for the search agents by using the objective function. Then, the agents are ranked in descending order according to fitness, and the first three best agents are provided with  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$  attributes, respectively.
3. Coefficient vectors  $\vec{a}$ ,  $\vec{A}$ , and  $\vec{C}$  are updated to guide the search process by using the following equations:

$$\vec{a} = 2(1 - \frac{k}{K_{max}}) \quad (2)$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

where  $k$  is the current iteration, and  $K_{max}$  denotes the maximum number of iterations.  $r_1$  and  $r_2$  are random vectors in  $[0,1]$ . Coefficient  $\vec{a}$  will decrease linearly from 2 to 0 at the end of the process.

4. Wolves  $\alpha$ ,  $\beta$ , and  $\delta$  are considered closest to the prey because the position of the prey is unknown. The encircling and hunting behaviors can be performed as follows:

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - w * \vec{X}|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - w * \vec{X}|, \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - w * \vec{X}| \end{aligned} \quad (5)$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \quad (6)$$

5. The velocities and positions of the wolves are updated by using the PSO approach as follows:

$$v_i^{k+1} = w * (v_i^k + c_1 r_1 (x_1 - x_i^k) + c_2 r_2 (x_2 - x_i^k) + c_3 r_3 (x_3 - x_i^k)) \quad (7)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (8)$$

where  $w$  represents the inertia constant generated randomly in  $[0,1]$ , and  $r_3$  is a random value in  $[0,1]$ .

6. Repeat steps 2 to 5 until the maximum number of iterations is reached. Afterward, the search agents stop moving, and final position  $X_\alpha$  is considered the best solution.

### 2.1.2 Real-coded Genetic Algorithm

Wright (1991) introduced the real-coded GA (RGA), which is another method of coding individuals using either real or binary values. In RGA, each gene represents a variable of the problem, and the size of genome is kept the same as the length of the solution. Wright (1991), Janikow and Michalewicz (1991), and Chang and Chen (1998) compared the two versions and concluded that RGA produces superior results compared with binary-coded GA.

In RGA and for a given optimization problem, a population of  $N$  possible solutions  $X_i = (x_i^1, x_i^2, \dots, x_i^d)$  is commonly generated in a pseudo-random manner. During iterations, the population evolves by forming generations of solutions through three operations, namely, selection, crossover, and mutation.

In this version of RGA, the stochastic tournament technique is adopted to perform the selection step, where individuals or "genomes" from the population are randomly selected for crossover. To perform the crossover operation, the random single-point technique is used. It consists of generating a random index  $l$  less than  $d$ . Therefore, genomes of the couple exchange genes from index  $l$  to  $d$ . For example, a pair of genomes  $X_1 = (x_1^1, \dots, x_1^l, \dots, x_1^d)$  and  $X_2 = (x_2^1, \dots, x_2^l, \dots, x_2^d)$  becomes  $X'_1 = (x_1^1, \dots, x_2^l, \dots, x_1^d)$  and  $X'_2 = (x_2^1, \dots, x_1^l, \dots, x_2^d)$  after performing a crossover.

To operate mutation, a random single-point mutation is employed, where one gene  $x_i^l$  is randomly chosen for a selected offspring  $X'_i$  and changed by a random feasible value  $y_i^l$ . Thus, a set of size  $2N$  includes fathers, and an offspring is obtained. To guarantee convergence,  $N$  individuals with low performance are eliminated, and the process is repeated until the maximum number of iterations is reached.

### 2.1.3 Gravitational Search Algorithm

GSA is one of the novel population-based metaheuristic algorithms. The population in GSA pertains to a set of search agents, which are considered masses that interact with one another by the laws of gravity and movement (Rashedi et al. 2009). The GSA process starts with initializing a population by considering a set of  $N$  agents  $X_i = (x_i^1, \dots, x_i^d)$  defined in the search space with dimension  $d$ . The  $i$ th position  $x_i^l$  of agent  $X_i$  in the  $l$ th dimension is randomly defined with a feasible value. Then, the fitness of each agent is calculated by using the objective function. Therefore, gravitational and inertia masses are computed as follows:

$$m_i(k) = \frac{fit_i(k) - worst(k)}{best(k) - worst(k)} \quad (9)$$

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^N m_j(k)} \tag{10}$$

$$M_{ai} = M_{pi} = M_{ii} = M_i \tag{11}$$

where  $fit_i(k)$  is the fitness value of the  $i$ th agent,  $best(k)$  and  $worst(k)$  are the best and worst fitness, respectively, at iteration  $k$ .  $M_{ai}$ ,  $M_{pi}$ , and  $M_{ii}$  are active, passive, and inertia masses respectively. The gravitational force acting on agent  $i$  by agent  $j$  is computed as follows:

$$F_{ij}^l(k) = G(k) \frac{M_{pi}(k) \cdot M_{aj}(k)}{R_{ij}(k) + \epsilon_g} (x_j^l(k) - x_i^l(k)) \tag{12}$$

where  $R_{ij}(k)$  is the Euclidean distance between agents  $i$  and  $j$ .  $\epsilon_g$  is a small constant, and  $G(k)$  is the gravitational constant updated at each iteration as follows:

$$G(k) = G_0 e^{\left(\frac{-\alpha_g k}{K_{max}}\right)} \tag{13}$$

where  $G_0$  and  $\alpha_g$  are initialized at the beginning, and  $K_{max}$  is the maximum number of iterations.  $G$  will be reduced with the increase in iterations. Therefore, the total force acting on agent  $i$  is calculated as follows:

$$F_i^l(k) = \sum_{j \in K_{best}, j \neq i} rand_j \cdot F_{ij}(k) \tag{14}$$

where  $rand_j$  is a random number in  $[0,1]$ , and  $K_{best}$  is the set of first  $K$  agents with the best fitness.  $K_{best}$  will decrease linearly, and only one agent will be applying force on the others by the end of the process.

Newton's law of movement denotes that object acceleration  $a$  is dependent only on applied force  $F$  and object mass  $M$ . In GSA, the following equation is used to calculate acceleration.

$$a_i^l(k) = \frac{F_i^l(k)}{M_{ii}(k)} \tag{15}$$

To move agents through the search space, velocities and positions are updated as follows:

$$v_i^l(k+1) = rand_i \cdot v_i^l(k) + a_i^l(k) \tag{16}$$

$$x_i^l(k+1) = x_i^l(k) + v_i^l(k+1) \tag{17}$$

where  $v_i^l(k)$  and  $x_i^l(k)$  denote the velocity and position, respectively, of agent  $i$  in dimension  $l$  at iteration  $k$ , whereas  $rand_i$  is a uniform random variable in  $[0,1]$ . The process is repeated until iterations reach their maximum limit, where the best fitness value is the global fitness, whereas the position of the corresponding agent is the global solution.

## 2.2 Case Study of Benchmark Functions

Before comparing HPSOGWO with RGA and GSA in reservoir operating optimization, four standard mathematical functions as reported by Yao et al. (1999) are used to evaluate the performance of algorithms. Table 1 shows the benchmark functions and their characteristics. Function  $F_1$  is unimodal, which has one global minimum and no local minima.  $F_2$ ,  $F_3$ , and  $F_4$  are multimodal functions with one global minimum and many local minima. The algorithms are launched 20 times independently for each function minimization, where the population size is set to 30, and the maximum iteration number is set to 3,000.

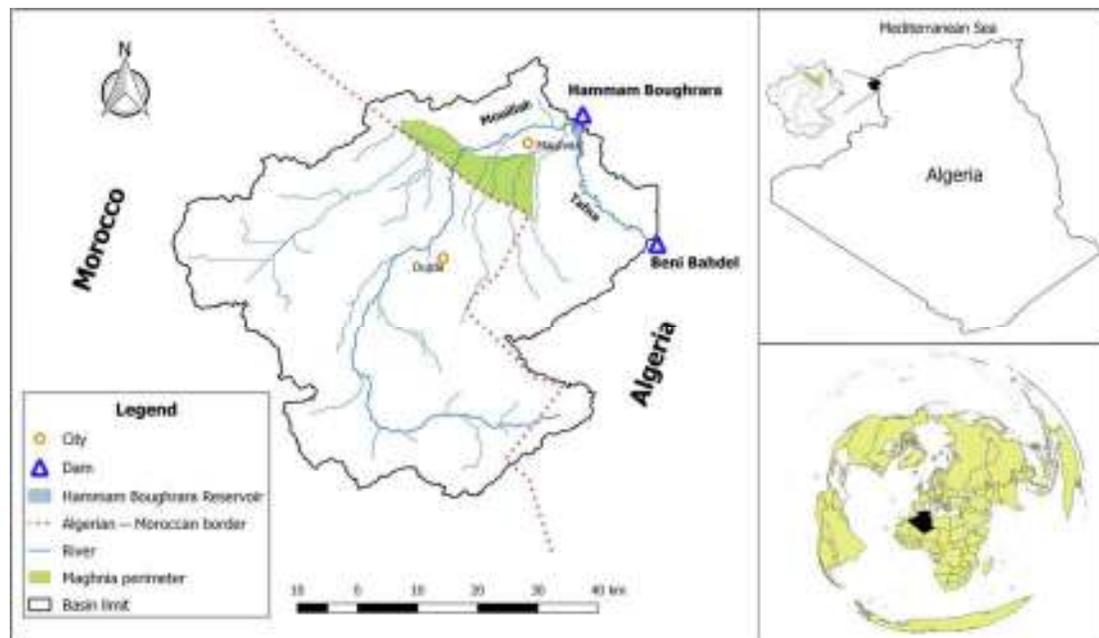
**Table 1** Mathematical benchmark functions

Function	$d^a$	Range	Global minimum
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0
$F_4(x) = \frac{1}{500} + \sum_{j=1}^{25} [\frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}]^{-1}$	2	[-65, 65]	1

<sup>a</sup>Search space dimension.

### 2.3 Case Study of Hammam Boughrara Reservoir

The Hammam Boughrara reservoir is the most important reservoir in Tlemcen Wilaya (far Northwest of Algeria). It is located on the confluence between Mouillah and Tafna Rivers at 1°39'61''W longitude and 34°52'23''N latitude, 13 km downstream from Maghnia Town. Figure 1 shows the geographic location of the study area. The reservoir is characterized by 177 million cubic meters (MCM) of capacity and 23.3 MCM of dead storage. It drains a basin shared between the Algerian and Moroccan territories with an area of 2,914 km<sup>2</sup>, a perimeter of 241 km, and a main thalweg of 104.4 km. The climate of the basin is semi-arid and experiences a wet season from November to April and a dry season from May to October. Maximum precipitations are observed in December with an average of 64.2 mm, whereas the minimal precipitations occur in July with an average of 2.36 mm. The average annual temperature and wind speed are 18.5°C and 24 m/s, respectively. Monthly inflows are variable with a maximum average of 6.29 MCM recorded frequently in November, and a minimum of 0.88 MCM in June. The reservoir started operations in September 2000 for multiple purposes, such as (1) water supply for Maghnia at 17 MCM/year, (2) water supply reinforcement for Oran Town (located approximately 140 km Northeast of Maghnia) at 33



**Fig. 1** Location of Hammam Boughrara reservoir

MCM/year, and (3) irrigation for the Maghnia perimeter (6,000 ha) at 9 MCM/year. The perimeter is supplied with 38 MCM/year of water by the Beni Bahdel reservoir, which is located approximately 30 km upstream from the Hammam Boughrara reservoir.

## 2.4 Reservoir Optimization Model

In general, the operation of the reservoir is governed by the following constraints:

### 2.4.1 Mass Balance Constraints

$$S_{t+1} = S_t + Q_t - R_t - Ev_t - I_t - O_t \quad t = 1, \dots, T \quad (18)$$

where  $S_t$  and  $S_{t+1}$  are the starting and ending reservoir storage (in MCM), respectively, at time period  $t$ .  $Q_t$ ,  $R_t$ ,  $Ev_t$ ,  $I_t$ , and  $O_t$  are inflow, release, evaporation, infiltration, and overflow (MCM), respectively, during time period  $t$ .  $T$  pertains to operating duration.

The evaporation loss is computed as follows:

$$Ev_t = ev_t A_t \quad (19)$$

where  $ev_t$  and  $A_t$  are evaporation and reservoir area, respectively, at time period  $t$ .

### 2.4.2 Storage Constraints

$$S_{min} \leq S_t \leq S_{max} \quad t = 1, \dots, T \quad (20)$$

where  $S_{min}$  and  $S_{max}$  are the minimum and maximum storage volumes, respectively, of the reservoir (MCM).

### 2.4.3 Release Constraints

$$0 \leq R_t \leq D_t \quad t = 1, \dots, T \quad (21)$$

where  $D_t$  is downstream demand (MCM) at time period  $t$ .

### 2.4.4 Overflow Constraints

During the operating period of the reservoir, overflow  $O_t$  is estimated as follows:

$$O_t = \text{Max}((S_{t+1} - S_{max}), 0) \quad t = 1, \dots, T \quad (22)$$

### 2.4.5 Constraints Application

If constraints (Equations [20] and [21]) are not satisfied, then the following penalties are applied (Karami et al. 2019):

$$P_{1,t} = \begin{cases} \frac{(S_{min} - S_{t+1})^2}{S_{min}} & \leftarrow \text{if } S_{t+1} < S_{min} \\ 0 & \leftarrow \text{otherwise} \end{cases} \quad (23)$$

$$P_{2,t} = \begin{cases} \frac{(S_{t+1} - S_{max})^2}{S_{max}} & \leftarrow \text{if } S_{t+1} > S_{max} \\ 0 & \leftarrow \text{otherwise} \end{cases} \quad (24)$$

$$P_{3,t} = \begin{cases} \frac{(R_t - D_t)^2}{D_{max}} & \leftarrow \text{if } R_t > D_t \\ 0 & \leftarrow \text{otherwise} \end{cases} \quad (25)$$

where  $D_{max}$  denotes the maximum downstream demand (MCM) over the considered operating duration  $T$ .

### 2.4.6 Objective Function

This study aims to minimize water supply deficits from the reservoir. As a result, volumes of release are considered decision variables. By applying penalties, the optimization function is expressed as follows:

$$Minimize(Z) = \sum_{t=1}^T \left( \left( \frac{R_t - D_t}{D_{max}} \right)^2 + P_{1,t} + P_{2,t} + P_{3,t} \right) \quad (26)$$

where  $T$  is considered 36 months. Figure 2 shows the monthly inflow, downstream demand, and evaporation loss curves from September 2009 to August 2012 (36 months). These data are used as input to the optimization model. In addition, the average of monthly infiltration loss (0.033 MCM) is considered. Notably, hydrological and water consumption data are provided by the water service office of Tlemcen.

### 2.5 Evaluation Indices

The reliability, resiliency, and vulnerability indices proposed by Hashimoto et al. (1982) are considered to evaluate the performance of HPSOGWO compared with RGA and GSA.

#### 2.5.1 Reliability

Volumetric reliability ( $a_v$ ) reflects the relation between release and downstream demand for total operation period  $T$ :

$$a_v = \frac{\sum_{t=1}^T R_t}{\sum_{t=1}^T D_t} \quad (27)$$

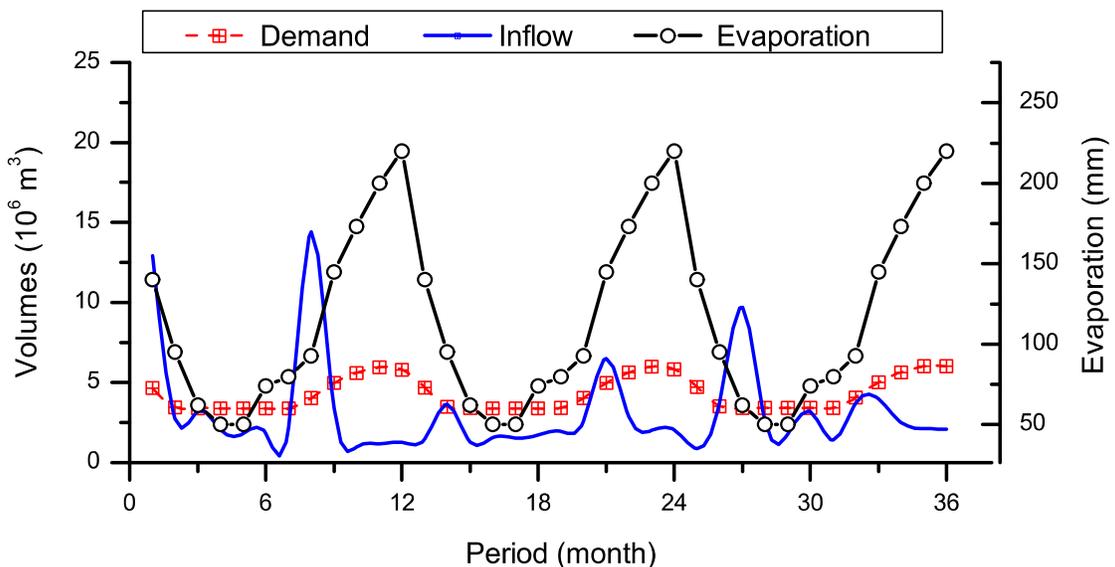


Fig. 2 Monthly inflow, evaporation, and downstream demand of Hammam Boughrara reservoir (September 2009 to August 2012)

## 2.5.2 Resiliency

It is defined by the time a system takes to return to normal state following a failure state:

$$\gamma = \frac{f_s}{f} \quad (28)$$

where  $\gamma$  represents the resiliency index,  $f$  is the number of failure periods, and  $f_s$  pertains to the number of failure series, which is a sequence of successive failure periods preceded and followed by non-failure periods.

## 2.5.3 Vulnerability

Vulnerability index ( $\sigma$ ) measures the severity of failures:

$$\sigma = \frac{\sum_{j=1}^{f_s} \max(Sh_j)}{f_s} \quad (29)$$

where  $Sh_j$  is the maximum shortage during the  $j$ th failure series.

# 3 Results and Discussion

## 3.1 Mathematical Functions

HPSOGWO was compared with RGA and GSA by minimizing four benchmark functions. Twenty runs were applied for each function provided in Table 1 to examine the effectiveness of HPSOGWO. A population of 30 agents and 3,000 iterations were employed, which indicates that a  $9.0E + 04$  evaluation function was performed for each run. The parameters of HPSOGWO were set as follows:  $C_1 = C_2 = C_3 = 0.5$  and  $w = 0.5(Rand() + 1)$ . However, the mutation frequency was set to 0.2 for RGA, whereas those of GSA were set as follows:  $G_0 = 100$  and  $\alpha_g = 20$ .

Table 2 reports the statistical comparison of the solutions found over 20 runs. Clearly, HPSOGWO found the best solutions, which were closest to the global minimum of functions  $F_1$ ,  $F_2$ , and  $F_3$ . However, the best solutions of the three algorithms were closer to one another for function  $F_4$ . In addition, the average and standard deviation of the HPSOGWO solutions were best for  $F_3$  and second best for the remainder.

According to the mean values of the  $F_2$  solutions, the three algorithms failed to reach the global minimum of function  $F_2$  over several runs, which indicates that they were trapped in the local minima. Results confirm those by Rashedi et al. (2009) and Singh and Singh (2017). Moreover, the results obtained by RGA largely differed from the overall minima of  $F_1$ ,  $F_2$ , and  $F_3$ , which may be caused by the small population size ( $N = 30$ ). Rashedi et al. (2009) and Bozorg-Haddad et al. (2016) revealed comparable findings when they used small population sizes in RGA.

Figure 3 displays the average values of the convergence trends. Clearly, HPSOGWO converged slower than RGA and GSA for  $F_1$ ,  $F_2$ , and  $F_3$ , but converged simultaneously with RGA and was faster than GSA for  $F_4$ . However, a stagnation of the RGA convergence trend was observed far from the global minimum for  $F_1$  and  $F_3$ , whereas it has progressed slowly for  $F_2$ . In addition, another stagnation was observed for  $F_2$  and  $F_3$  by GSA, which indicates the trapping of RGA and GSA in the local minima. On the contrary, the average convergence curve of HPSOGWO was improved with the progress in iterations in all cases, which

**Table 2** HPSOGWO, RGA, and GSA statistical results of 20 runs for benchmark functions

Algorithm	Stat.criterion <sup>a</sup>	Mathematical function			
		$F_1$	$F_2$	$F_3$	$F_4$
HPSOGWO	Min	$5.61E - 132$	$0.00E + 00$	$0.00E + 00$	0.9980
RGA		$5.69E - 01$	$7.94E - 01$	$5.88E - 01$	0.9980
GSA		$2.60E - 17$	$1.09E + 01$	$1.05E - 03$	0.9980
HPSOGWO	Max	$1.23E - 05$	$2.81E + 01$	$8.85E - 03$	0.9985
RGA		$1.93E + 00$	$3.59E + 00$	$1.03E + 00$	0.9980
GSA		$9.18E - 17$	$3.88E + 01$	$1.98E - 01$	6.6955
HPSOGWO	Mean	$6.19E - 07$	$7.88E + 00$	$2.82E - 03$	0.9980
RGA		$1.30E + 00$	$1.78E + 00$	$8.98E - 01$	0.9980
GSA		$5.24E - 17$	$2.29E + 01$	$4.74E - 02$	2.1429
HPSOGWO	SD	$2.76E - 06$	$1.00E + 01$	$3.95E - 03$	$1.04E - 04$
RGA		$3.61E - 01$	$6.29E - 01$	$1.27E - 01$	$1.50E - 07$
GSA		$1.99E - 17$	$6.72E + 00$	$6.82E - 02$	$1.47E + 00$

<sup>a</sup>Statistical criterion.

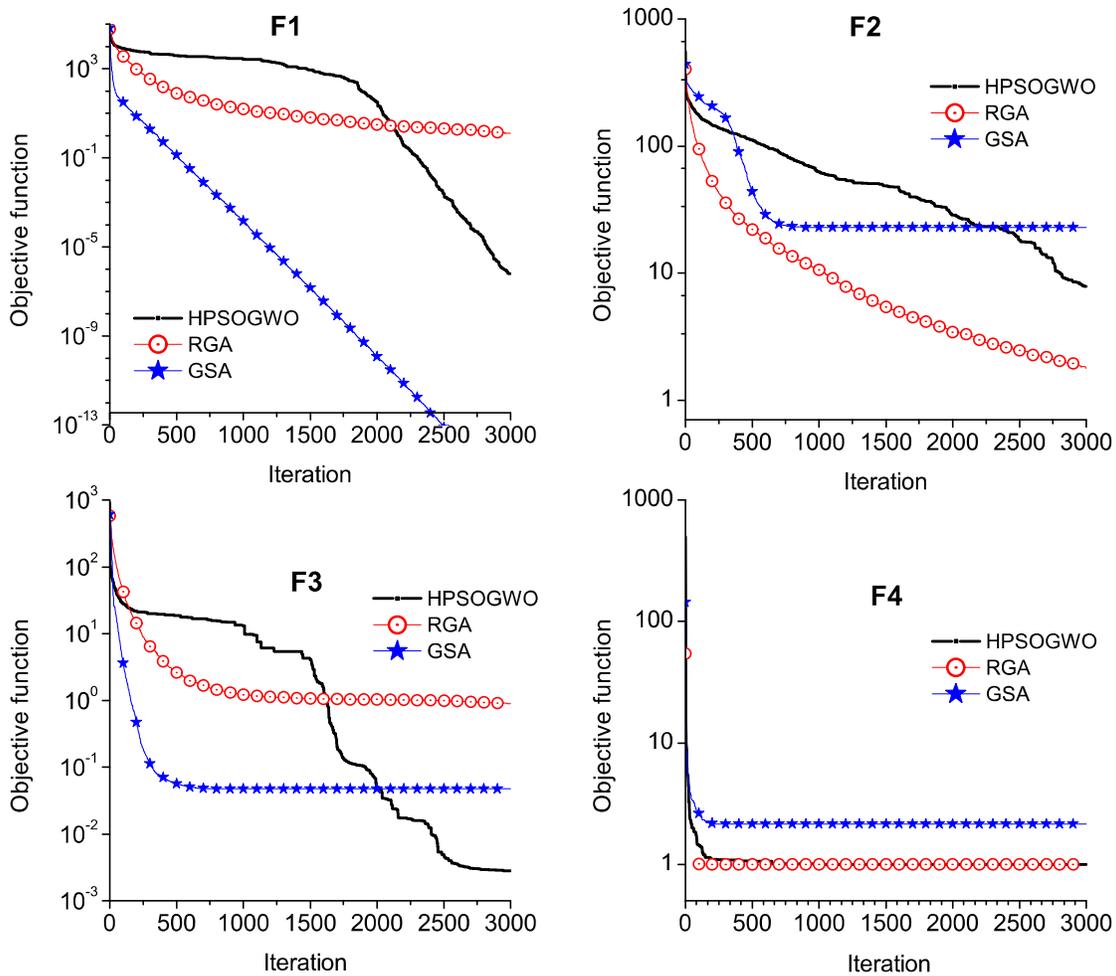
indicates that it could escape from local minima trapping. Thus, the results demonstrate the effectiveness of HPSOGWO in solving similar problems.

### 3.2 Case of Hammam Boughrara Reservoir

Table 3 reports the sensitivity analysis of HPSOGWO, RGA, and GSA. The test was carried out to determine the suitable values of algorithm parameters. For HPSOGWO, the best objective function value ( $3.05E - 08$ ) was achieved by a population size of 30 and  $C_1 = C_2 = C_3 = 0.6$ . For RGA, a population size of 100 and a mutation frequency of 0.2 provided the best objective function value ( $8.57E - 05$ ). Typically for GSA, a population size of 100,  $\alpha_g = 10$ , and  $G_0 = 20$  are necessary to obtain the best value of the objective function ( $1.31E - 07$ ). Thus, the HPSOGWO found the best solution using the small search population size ( $N = 30$ ). In addition, Table 3 highlights that the solutions achieved by RGA and GSA were improved by increasing the population size and, consequently, function evaluation. Notably, the maximum number of iterations equals 3,000 was applied.

Table 4 provides the results of 10 random runs of the three algorithms, where the tuning parameters values cited in Table 3 that gave the best solutions were used. Results in Table 4 also indicate that HPSOGWO found the best solution, whereas the worst, average, and standard deviation of HPSOGWO were worse than those of GSA, but better than those of RGA. The variation coefficient results of HPSOGWO was less than those of RGA and GSA, which indicate that HPSOGWO is less stable compared with the others. Despite the high value of variation coefficient, the average and standard deviation values prove the efficiency of HPSOGWO.

Figure 4 shows a comparison between the convergence trends of the algorithms and the minimum, maximum, and average solution curves for HPSOGWO. Results show that



**Fig. 3** Average convergence trends of HPSOGWO, RGA, and GSA for mathematical functions

the convergence of HPSOGWO occurred earlier than RGA and GSA, and their minimum, maximum, and average solution curves were converged to one another.

Figure 5 presents the results of the root square mean error (RMSE) between monthly release and downstream demand. The RMSE of the HPSOGWO results were in the order of  $2.5E - 04$  and better than RGA for each month by 100%, which is also better than those of GSA in the majority of months by 75%.

Figure 6 illustrates the optimal releases from the Hammam Boughrara reservoir as calculated by the three algorithms and storage evolution during the operating period. Results indicate the superposition of the release curves with downstream demand for the three algorithms. The derived curves for storage presented the same trend and found between the minimum and maximum storage volumes of the reservoir. In fact, they evidently show the algorithms' consideration of the minimum and maximum storage constraints.

Table 5 presents high reliability and resiliency, and low vulnerability for all algorithms. The values of the indices were identical for HPSOGWO and GSA and better than RGA. These finding confirms the previous results and proves the capability of HPSOGWO to optimize reservoir operation.

**Table 3** Sensitivity analysis of RGA, GSA, and HPSOGWO

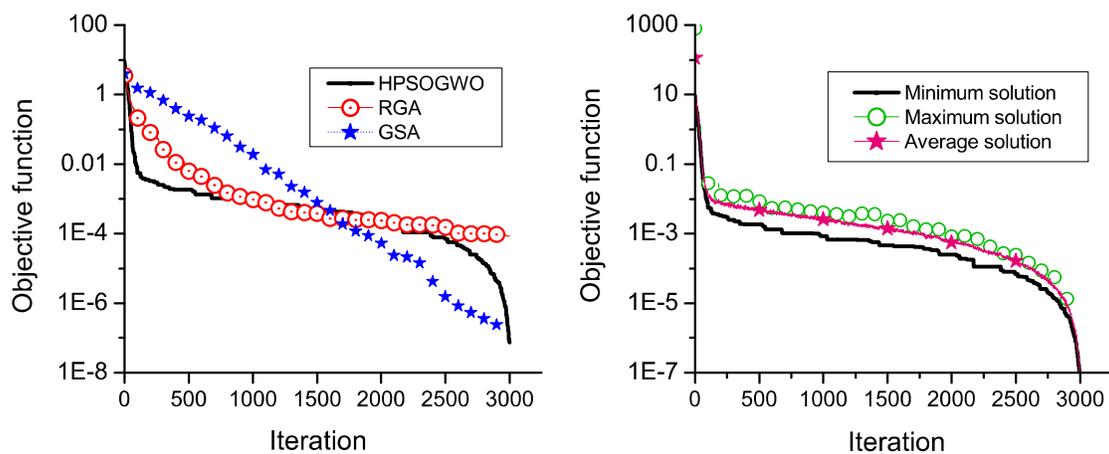
HPSOGWO						
Pop. size <sup>a</sup>	O. function <sup>b</sup>	C <sub>1</sub>	O. function <sup>b</sup>	C <sub>2</sub>	O. function <sup>b</sup>	O. function <sup>b</sup>
10	4.49E - 06	0.2	7.05E - 07	0.2	4.99E - 07	5.44E - 07
30	3.05E - 08	0.4	9.26E - 07	0.4	4.02E - 07	1.97E - 07
50	1.97E - 07	0.5	1.62E - 07	0.5	1.62E - 07	2.35E - 07
70	2.52E - 07	0.6	3.05E - 08	0.6	3.05E - 08	3.05E - 08
100	1.62E - 07	0.8	3.05E - 06	0.8	5.41E - 06	5.64E - 06
RGA						
Population size	Objective function	Mutation probability	Objective function			
10	1.43E - 03	0.20	8.57E - 05			
30	2.33E - 04	0.40	1.23E - 03			
50	1.70E - 04	0.50	3.94E - 02			
70	1.05E - 04	0.60	9.99E - 02			
100	8.57E - 05	0.80	3.88E - 01			
GSA						
Population size	Objective function	G <sub>0</sub>	Objective function	α <sub>g</sub>	Objective function	Objective function
10	2.01E - 02	20	1.31E - 07	1	2.00E - 02	
30	1.61E - 04	50	1.30E - 06	5	1.03E - 04	
50	4.54E - 05	70	1.95E - 06	10	1.31E - 07	
70	9.41E - 07	100	2.20E - 06	20	1.28E - 06	
100	1.31E - 07	150	1.90E - 06	30	2.21E - 04	

<sup>a</sup>Population size.

<sup>b</sup>Objective function.

**Table 4** Ten random results for HPSOGWO, RGA, and GSA

Run	HPSOGWO	RGA	GSA
1	$7.63E - 06$	$6.34E - 05$	$2.74E - 07$
2	$8.20E - 06$	$3.67E - 05$	$6.35E - 07$
3	$1.01E - 05$	$4.01E - 05$	$6.28E - 07$
4	$1.06E - 05$	$5.58E - 05$	$6.14E - 07$
5	$7.50E - 08$	$5.52E - 05$	$2.77E - 07$
6	$3.17E - 05$	$5.58E - 05$	$6.49E - 07$
7	$6.54E - 06$	$3.50E - 05$	$3.53E - 07$
8	$2.42E - 06$	$5.95E - 05$	$2.83E - 07$
9	$1.32E - 06$	$5.31E - 05$	$2.10E - 07$
10	$1.02E - 07$	$4.79E - 05$	$3.81E - 07$
Best	$7.50E - 08$	$3.50E - 05$	$2.10E - 07$
Worst	$3.17E - 05$	$6.34E - 05$	$6.49E - 07$
Average	$7.87E - 06$	$5.02E - 05$	$4.30E - 07$
Standard deviation	$9.29E - 06$	$9.86E - 06$	$1.79E - 07$
Variation coefficient	1.18	0.196	0.417



**Fig. 4** **a** convergence trends of HPSOGWO, RGA, and GSA for Hammam Boughrara reservoir and **b** minimum, maximum, and average solutions for HPSOGWO algorithm

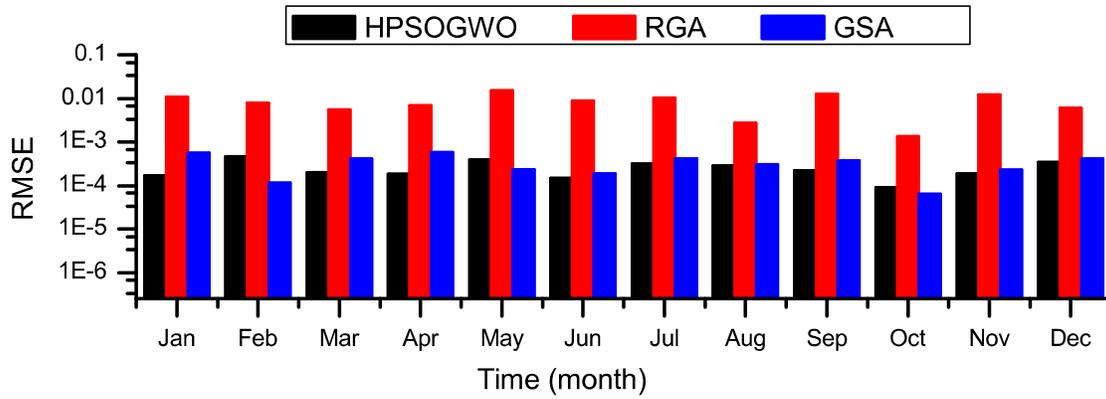


Fig. 5 Monthly RMSE of algorithms during the operation period (MCM)

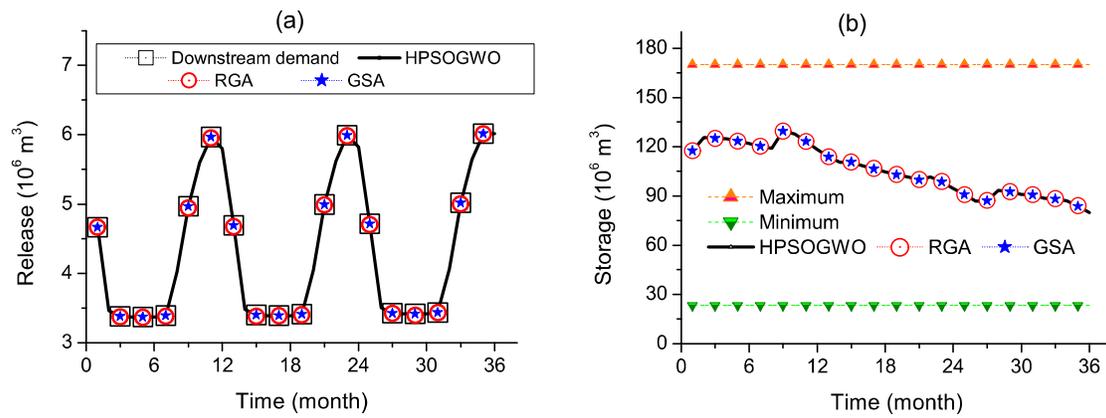


Fig. 6 a released water volume and b storage volume evolution

Table 5 Results of reliability, resiliency, and vulnerability indices for the Hammam Boughrara reservoir

Index	Hybrid PSOGWO	Real-coded GA	GSA
Reliability (%)	99.99	99.84	99.99
Resiliency (%)	100	90	100
Vulnerability (%)	0.02	0.58	0.02

## 4 Conclusion

Evolutionary algorithms have recently been used to solve a wide range of optimization problems. This study evaluated a hybrid algorithm of PSO and GWO, which is collectively called HPSOGWO. The study used two case studies, namely, minimizing four benchmark functions and solving the single reservoir optimization problem using the non-linear objective function. In addition, HPSOGWO was compared with two efficient algorithms, namely, RGA and GSA, whose performances were verified in the literature.

In the first case study, the algorithms were independently run for 20 times, where HPSOGWO proved its competitive capability of minimizing unimodal and multimodal mathematical functions. It was ranked the most powerful among half of the studied benchmarks and second for the remainder.

Optimizing the Hammam Boughrara reservoir operation was selected as the second case study, where the three algorithms were independently run 10 times. Sensitivity analysis demonstrated that HPSOGWO required a small search population size and can achieve high-quality solutions. Moreover, the average monthly RMSE indicated the high precision of release computation by HPSOGWO compared with the other algorithms. However, the average and standard deviation of the results revealed that HPSOGWO ranked second to GSA. The reliability and resiliency indices for the algorithms were high, whereas the vulnerability index was low. Furthermore, the indices for HPSOGWO and GSA were identical and better than RGA. Hence, the previous results prove the usability of HPSOGWO for such problems.

Despite the capability of HPSOGWO to overcome trapping in the local minima, it suffers from drawbacks, such as low stability, which is indicated by the high value of variation coefficient compared with RGA and GSA. Thus, additional efforts are needed to improve its performance.

## References

- Ahmed JA, Sarma AK (2005) Genetic algorithm for optimal operating policy of a multipurpose reservoir. *Water resources management* 19(2):145–161
- Arun Kumar R, Jothiprakash V (2012) Optimal reservoir operation for hydropower generation using non-linear programming model. *Journal of The Institution of Engineers (India): Series A* 93(2):111–120
- Bozorg-Haddad O, Janbaz M, Loáiciga HA (2016) Application of the gravity search algorithm to multi-reservoir operation optimization. *Advances in water resources* 98:173–185
- Chang F-J, Chen L (1998) Real-coded genetic algorithm for rule-based flood control reservoir management. *Water Resour Manag* 12(3):185–198
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE*, pp 39–43
- Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications* 30(2):413–435
- Ghimire BN, Reddy MJ (2013) Optimal reservoir operation for hydropower production using particle swarm optimization and sustainability analysis of hydropower. *ISH Journal of Hydraulic Engineering* 19(3):196–210
- Hashimoto T, Stedinger JR, Loucks DP (1982) Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation. *Water resources research* 18(1):14–20
- Janikow CZ, Michalewicz Z (1991) An experimental comparison of binary and floating point representations in genetic algorithms. In: *ICGA*, pp 31–36
- Karami H, Ehteram M, Mousavi S-F, Farzin S, Kisi O, El-Shafie A (2019) Optimization of energy management and conversion in the water systems based on evolutionary algorithms. *Neural Comput & Applic* 31(10):5951–5964

- Kumar D, Reddy M (2007) Multipurpose reservoir operation using particle swarm optimization. *J Water Resour Plan Manag* 133(3):192–201
- Loucks DP (1968) Computer models for reservoir regulation. *J Sanit Eng Div* 94(4):657–670
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Advances in engineering software* 69:46–61
- Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. *Information sciences* 179(13):2232–2248
- Sharif M, Swamy VSV (2014) Development of lingo-based optimization model for multi-reservoir systems operation. *International Journal of Hydrology Science and Technology* 4(2):126–138
- Singh N, Singh SB (2017) Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J Appl Math*, 1–15
- Stedinger JR, Sule BF, Loucks DP (1984) Stochastic dynamic programming models for reservoir operation optimization. *Water resources research* 20(11):1499–1505
- Wolpert DH, Macready WG et al (1997) No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1(1):67–82
- Wright AH (1991) Genetic algorithms for real parameter optimization. In: *Foundations of genetic algorithms*, vol 1, Elsevier, pp 205–218
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation* 3(2):82–102

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Manuscrit 2 : Development of Reservoir  
Management Optimal Rules : Case of  
Hammam Boughrara Dam, Wilaya of  
Tlemcen, Algeria**

# Development of Reservoir Management Optimal Rules: Case of Hammam Boughrara Dam, Wilaya of Tlemcen, Algeria

Saad Dahmani, Ahmed Ferhait, Djilali Yebdri, Rabah Bounoua and Hakim Djafer Khodja

## 1 Introduction

Worldwide, more than half of the drinking water is extracted from rivers or from reservoirs built along their banks [14]. The development of dam reservoirs management policies can be a delicate issue because of various reasons: the variation and uncertainty of liquid flow to the reservoir; water releases are not by water only, but also depend on the level of water in the tank; the release period (the season); regularization of the tank mode (annual or inter-annual),... etc. [9] summarized other reasons.

The optimization algorithms aim to determine the values of decision variables that maximize or minimize an objective function under constraints [15]. Several methods have been used to solve the problem of management of dams. Several methods have been used to solve the problem of management of dams. Asmadi [3], Labadie [10], Wurbs [15], Yakowitz [16] and Yeh [17], suggested a review of these methods.

With 59 hm<sup>3</sup> of regularized annual volume, Hammam Boughrara reservoir presents a potential in water resources management in the region. Its role is to satisfy the water needs of Maghnia city (approximately, 154,000 inhabitants in 2012), irrigation of Maghnia agricultural perimeter (about 6000 Ha) and strengthening of the drinking water supply of Oran city (approximately 700,000 inhabitants in 2012).

The main objectives of this study include the following:

Development of management decision plans with a monthly time step, taking into account: the state of water storage in the reservoir at the beginning of the month in question, the random monthly contributions to the previous month and the needs

---

S. Dahmani (✉) · A. Ferhait · D. Yebdri · R. Bounoua · H. Djafer Khodja  
Bouira University, DRISSI Yahia Street, 10000 Bouira, Algeria  
e-mail: s.dahmani@univ-bouira.dz

A. Ferhait  
e-mail: Ferhati28220@Yahoo.fr

of consumers at a considered month. For this purpose, the study duration is from 2012 to 2030.

Amend if necessary. Analysis by multiple regressions of the independent variables (monthly water supplies, the state of the storage at the beginning of each month and the monthly water demands of consumers) on the side, and in other side the dependent variable, represented in this case by the decision to release of water.

## 2 Materials and Methods

### 2.1 Presentation of the Study Area

With 177 hm<sup>3</sup> of capacity, Hammam Boughrara reservoir is the largest of those built in the wilaya of Tlemcen in the west of Algeria (A.N.B.T [1]). It is located on the Mouillah river confluence with Wadi Tafna, at 10 km downstream from the town of Maghnia (W. Tlemcen). The dam belongs to the catchment area of Wadi Mouillah area (2000 km<sup>2</sup>) is shared between the Algerian territory and the Moroccan territory (Fig. 1). This basin is limited by 241 km of perimeter and composed in its majority by the plains of Angad (Oujda, Morocco) and Zrigua (Maghnia, Algeria). The length of his water-main course is of 104.4 km [5]. The water from this dam is done in November 1998, while exploitation began in the month of September 2000 (A.N.B.T, Etude d'impact sur l'environnement du barrage Hammam Boughrara: Rapport final, volume I, [1]). In 2005, the dam

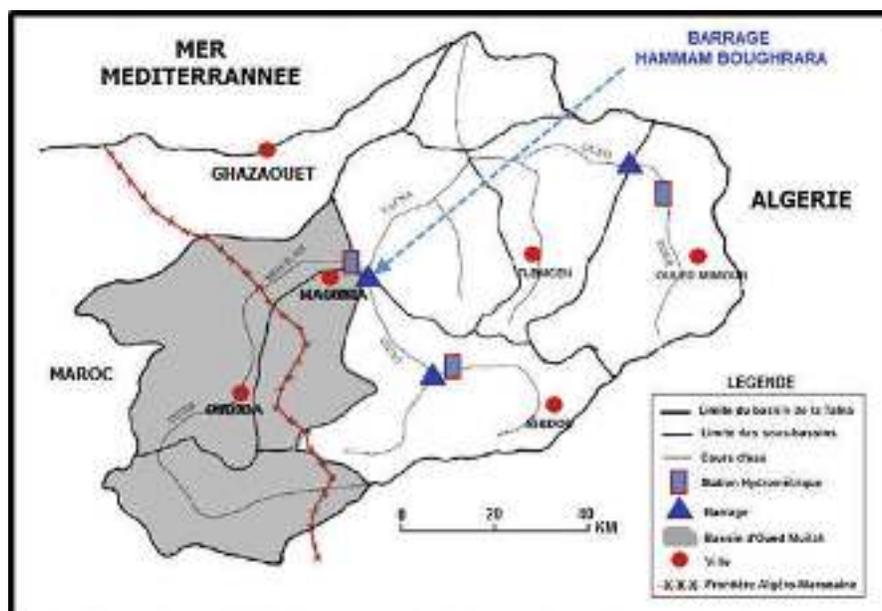


Fig. 1 Situation of sub basin and Wadi Mouillah dam Hammam Boughrara

underwent an operation of dredging although the silting rate has not exceeded 0.88% in 2010 (l'Eau, Atelier Régional 13 et 14 décembre [11]).

The wind conditions have been studied from observational data at Tlemcen and Oran weather stations that have the most representative data. These are the westerly winds and northwest prevailing in the territory studied. The maximum average wind speed 24 m/s but it can reach 44 m/s (Ajouter la Référence).

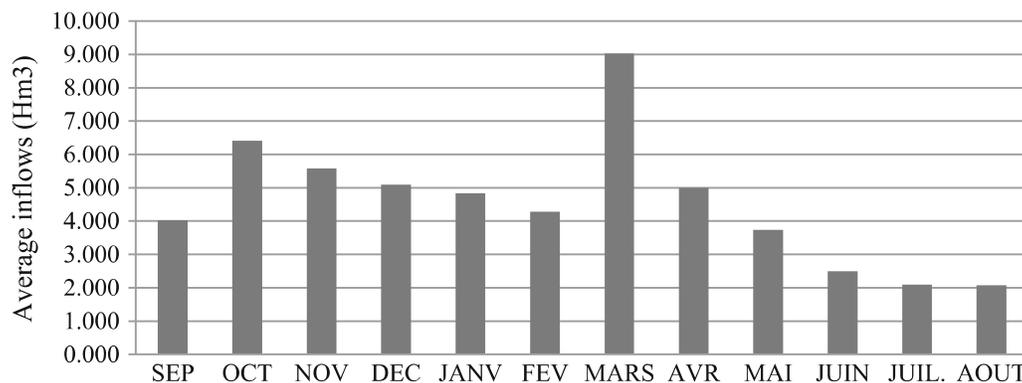
Several studies such as [5, 8, 13] showed that the dam is reached an advanced degree of eutrophication.

## 2.2 Reservoir Liquid Inflows

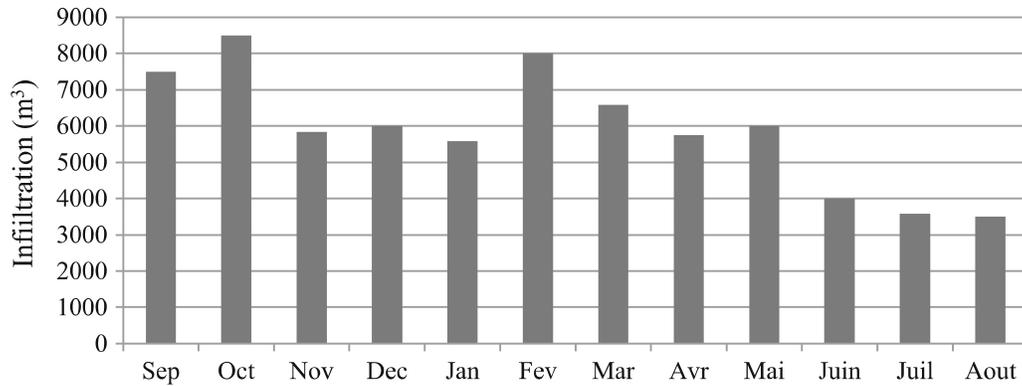
Inflows to Hammam Boughrara dam are arriving through the Wadi Tafna and Mouillah contributions. The measures of the contributions to the level of the watershed of Wadi Mouillah are taken using «Sidi Belkheir» station near the dam (Fig. 1). The following figure (Fig. 2) shows the variation of monthly intakes conveyed by the Mouillah Wadi and Wadi Tafna.

## 2.3 Reservoir Water Loss by Evaporation

The monthly average volumes of water leaks that have been saved during the operation of the dam (September 2000–June 2012) vary between a volume of 3500 and 8500 m<sup>3</sup> (Fig. 4) (Manque référence). Similarly, for the volumes lost by evaporation, the volumes lost by infiltration are taken the month of calculation function (Fig. 3).



**Fig. 2** Monthly average inflows of Hammam Boughrara reservoir during the period (1972–2012)



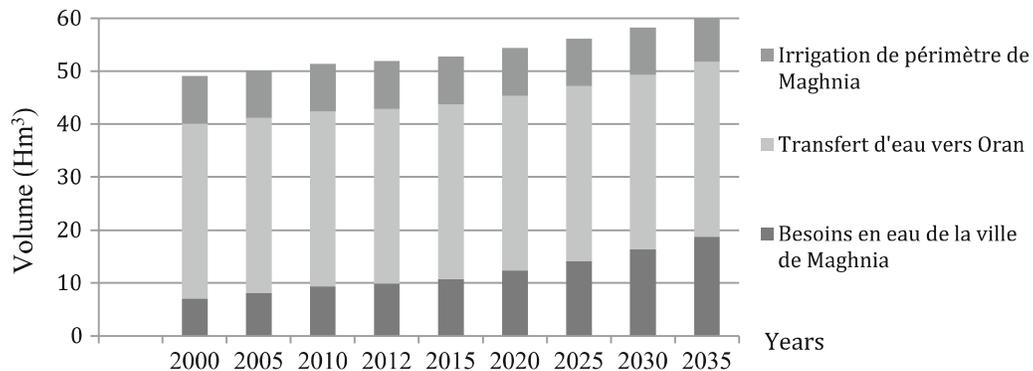
**Fig. 3** Monthly average infiltration volumes (September 2000–June 2012) [2]

## 2.4 Use of Water from the Dam

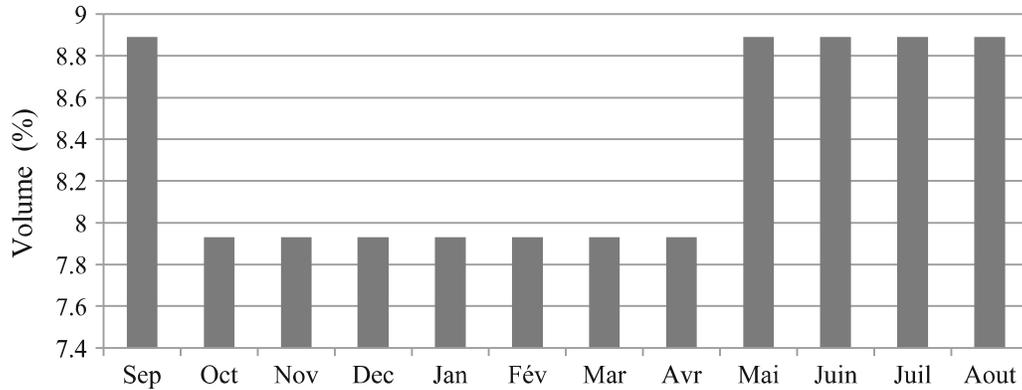
With the regularized volume of  $59 \text{ hm}^3/\text{year}$ , the dam is intended initially to supply drinking water to Maghnia and Oran towns of with annual volumes of ( $17$  and  $33 \text{ hm}^3$ ) respectively, thus irrigation perimeter of Tafna with an annual volume of  $9 \text{ hm}^3$  [1]. In this study, water (drinking and industrial) of Maghnia town needs are projected to 2030 horizon. A rate of increase of population of the order of  $3.1\%$  (ONS 2014). The transfer of water from Hammam Boughrara reservoir to Oran is directed to carry a flow equal  $33 \text{ hm}^3/\text{year}$ . In this study, this flow is assumed to be constant up to 2030. Hammam Boughrara dam is for irrigation of the Maghnia perimeter (area of  $6000 \text{ ha}$ ) with an annual volume of  $9 \text{ hm}^3/\text{year}$ . The latter is assumed to be constant up to 2030 (Figs. 4 and 5).

Below are the monthly plans of water allocation for the three consumers (towns of Maghnia and Oran).

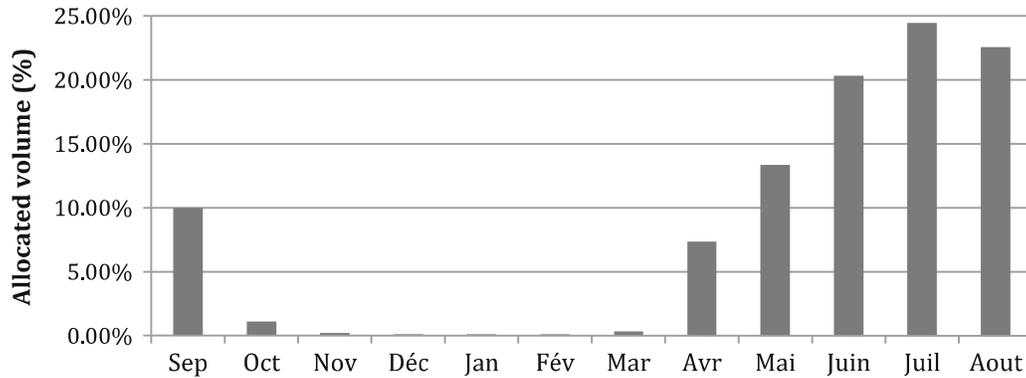
The annual distribution of allocated volume ( $9 \text{ hm}^3$ ) is supposed to be constant during the study period (2012–2030). The monthly distribution of the volume follows the following distribution (Fig. 6):



**Fig. 4** Evolution of allocated volumes (2000–2030)



**Fig. 5** Monthly distribution of the volumes allocated to the town of Maghnia and the city of Oran [6]



**Fig. 6** Monthly distribution of were desirous of water allocated to irrigation perimeter of Maghnia [7]

## 2.5 Reservoir Management Optimization

### 2.5.1 Choice of the Reservoir Objective Function

The objective function (or criterion of optimization) is the heart of an optimization model. Wurbs [15] listed examples of this feature mentioned in the literature in three categories: costs and economic benefits, hydroelectric power production, and reliability and availability of water. In this study, the chosen objective function is the minimization of the absolute difference between the release and monthly water demands, by another way, the releases must follow (the maximum possible) the trajectory of water demands. Each monthly demand is the sum of the volumes allocated to the towns of Maghnia and Oran and irrigation of Maghnia perimeter. The optimization criterion is expressed as follows:

$$\text{Optimisation criterion} = \text{Min}|R - D| \quad (1)$$

Where  $R$  is the monthly water allocation (decision) in  $\text{hm}^3$ .  $D$  is the monthly water demand in  $\text{hm}^3$ .  $R$  and  $D$  must be superior or equal to zero.

### 2.5.2 Stochastic Dynamic Programming

The adopted optimization approach is explicit stochastic optimization. The latter takes into account uncertainty on the natural contributions by integrating their probability distribution directly in the model [18]. Stochastic dynamic programming (SDP) is one of the methods based on the stochastic optimization approach. The method was developed by Richard Bellman in 1957 [4]. It has been used by several authors to solve the problem of optimization of management of dams. Labadie [10] presented a review of these algorithms. Stochastic dynamic programming (SDP) is looking for an optimal solution to the following mathematical program:

$$F(S_t) = \text{Max ou Min}\{f(Q_t, S_t, U_t) + E[F_{t+1}(S_{t+1})]\} \\ U_t \in U \quad (2)$$

where  $Q_t$  is liquid supply to the reservoir at the time  $t$ ,  $S_t$  is the water storage level in the reservoir at the time  $t$ ,  $U_t$  is the command (decision) of water at time  $t$ ,  $U$  is all possible decisions values,  $E$  is the mathematical expectation,  $f(Q_t, S_t, U_t)$  is immediate optimization revenue, calculated by the objective function and  $t$  is time (monthly step).

With this method, the planning period is divided into ( $T$ ) steps. The State of the system at each stage is described by a state vector. One of the components of the state vector is the level of water storage in the reservoir. The tank is discretized in ( $K$ ) levels equal [12]. Reservoir water inflows are divided into ( $I$ ) equal classes. The calculation process is to start with the final step ( $t = T$ ) up to the initial ( $t = T_0$ ). At each step, the optimized income of current step is evaluated. Thereafter, it will be added to revenues already calculated during the previous steps. The expected value is calculated using the probabilities of occurrence of contributions during recorded history. The independence between the contributions is assessed by one of the tests of independence, such as: Wald-Wolfowitz (1943) or Khi 2 tests.

In this study, in order to well formulate the problem, Eq. (2) is simplified as follows:

$$F(S_t) = \text{Min}\left\{f(Q_t, S_t, R_t) + \sum_{j=1}^I P_j x F_{t+1}(S_{t+1})\right\} \\ R_t \in U \quad (3)$$

Equation (3), the transition between the states of the reservoir is limited by the continuity equation (Eq. 4):

$$S_{t+1} = S_t + Q_t - V_{ev_t} - V_{inf_t} - R_t - Dev_t \quad (4)$$

Under the following condition:

$$S_{\min} \leq S_{i+1} \leq S_{\max}$$

where  $S_{t+1}$  is the reservoir water storage at time  $(t + 1)$ ,  $S_t$  is the reservoir water storage at time  $(t)$ ,  $Q_t$  is liquid supply to the tank at the time  $(t)$ ,  $V_{ev_t}$  water evaporated volume at time  $(t)$ ,  $V_{inf_t}$  is infiltration water lost volume at time  $(t)$ ,  $R_t$  is the reservoir water release at time  $(t)$ , and  $D_{ev_t}$  is reservoir water diversion at time  $(t)$ ,  $S_{\min}$  is the reservoir minimum storage and  $S_{\max}$  is the reservoir maximum storage.

### 2.5.3 Application on Hammam Bouhrara Reservoir

The duration of planning is 2012–2030. Optimization is applied on a monthly scale. For this purpose, the calculation has held level 336 months of optimization. Hammam Bouhrara tank is discretized into 153 classes (or units) volume. The Discretization is applied on the volume between the level corresponds to the lower intake ( $H = 272$  m) corresponds to the dead volume ( $V_u = 0$  hm<sup>3</sup>) and the maximum level of the tank ( $H = 305$  m) corresponds to the maximum volume ( $V_u = 153$  hm<sup>3</sup>). Each discretization class matches (01 hm<sup>3</sup>).

The inflows to the reservoir are discretized into 37 equal classes. Each interval is equal to (01 hm<sup>3</sup>) monthly volume. 37 intervals are justified by the volume of monthly contribution appeared in the month of October 2008, which is quantified in (36.348 hm<sup>3</sup>) [2]. ( $P_j$ ) the probability for each class of contribution is calculated by the frequency of the input in the series of available measure.

### 2.5.4 Regression Analysis

After the calculated results of reservoir optimal release value for each month for the chosen objective function, a multiple regression analysis is performed. This analysis on the one hand, connects independent variables represented in this case by contributions of liquid in the reservoir ( $Q_i$ ), states of storage (reservoir levels) ( $S_i$ ), and consumers monthly water demand ( $D_i$ ) and on the other hand, the dependent variable in this case represented by the optimal water release. Volume discretized from the reservoir (between the lower intake level and the maximum intake level) is divided into three (03) intervals in order to develop regression relationships between dependents and independents variables. These regression relationships are expressed in the following equation:

$$R_t = A_n S_t + B_n Q_t + C_n D_t + F_n \quad (5)$$

where  $R_t$  is the optimal release on ( $\text{hm}^3$ ),  $S_t$  is storage of water at the beginning of months en ( $\text{hm}^3$ ),  $Q_t$  is the reservoir inflow during the month on ( $\text{hm}^3$ ),  $D_t$  is the water demand of different consumers during the month on ( $\text{hm}^3$ ),  $A$ ,  $B$ ,  $C$ , and  $D$  are the regression coefficients.

### 3 Results

The following Fig. 7 show examples of the monthly calculated optimal releases (for January and June of 2017). The optimal release does not concern volumes dumped by the works of protection as the spillway.

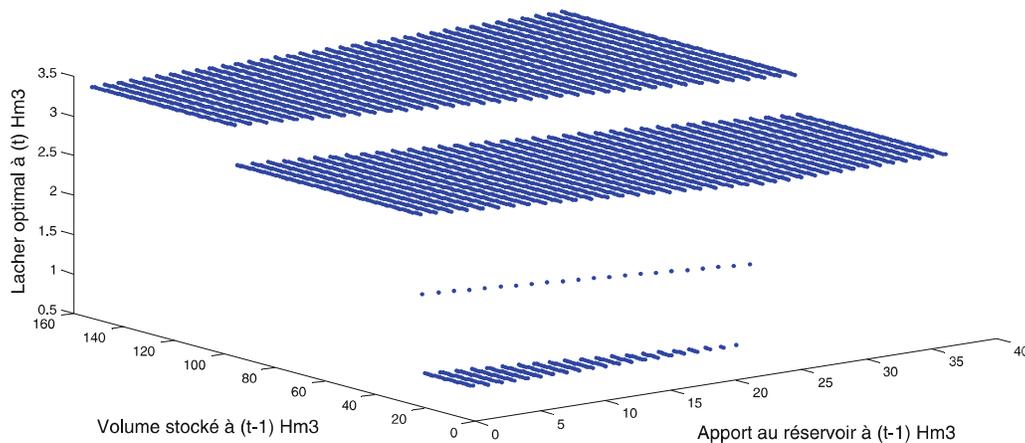
Below is the calculated optimal releases aggregation (Fig. 9).

The following Table 1 shows the multiple regression of the dependent and independent variables analysis:

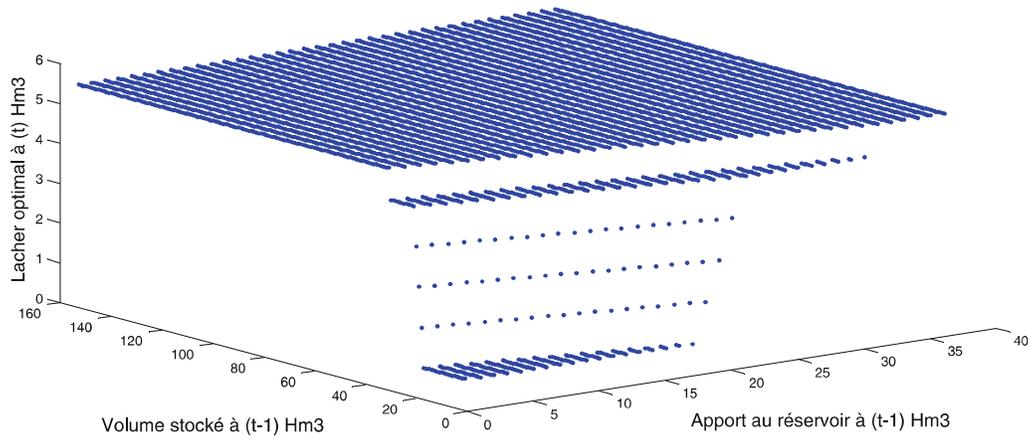
For all of regularizable total volume ( $153 \text{ hm}^3$ ), the multiple regression Eq. (5) can be written as follows:

$$R_t = 0.0075 S_t + 0.0154 Q_t + 0 D_t + 3.41$$

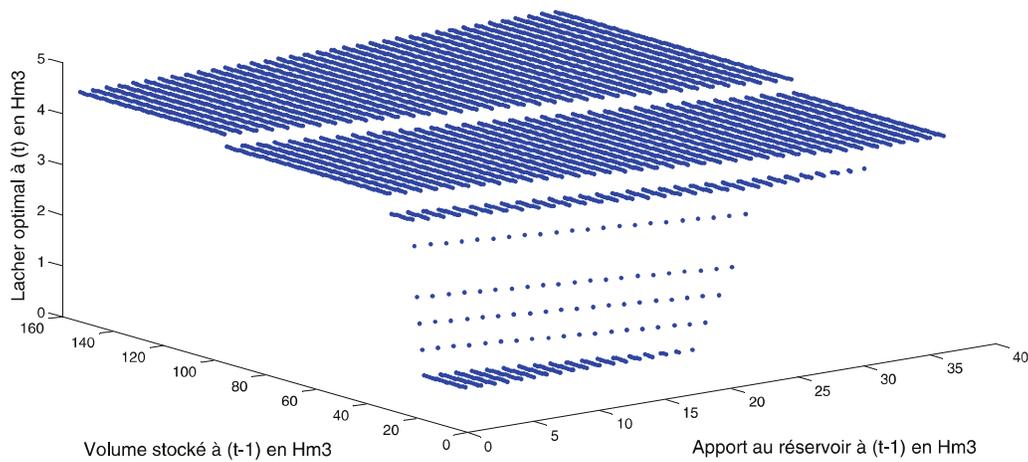
With a coefficient of correlation ( $R^2 = 0.31$ ).



**Fig. 7** Calculated optimal releases for January, 2017



**Fig. 8** Calculated optimal releases for June, 2017



**Fig. 9** The optimal release based on reservoir water storage levels and inflows

**Table 1** Calculation results of the multiple regression coefficients

Interval index ( <i>n</i> )	Reservoir interval storage from the lower intake (hm <sup>3</sup> ) level			Regression coefficients				<i>R</i> <sup>2</sup>
				<i>A<sub>n</sub></i>	<i>B<sub>n</sub></i>	<i>C<sub>n</sub></i>	<i>F<sub>n</sub></i>	
1	0	–	17	0.071	0.069	0	1.253	0.61
2	18	–	30	0.161	0.161	0	–0.471	0.52
3	31	–	153	0.0026	0.0021	0	4.162	0.74

## 4 Discussion

Analysis of the optimization results in Fig. 8, shows eight (08) plans of optimal release according to the sum of the volume of water storage in the reservoir (at time  $t-1$ ) and probable liquid flows arriving at the tank (at time  $t-1$ ). The first (1st) leading to optimal release corresponds to a regularized total counted from the decision-making of lower water level and including the volume of water stored at ( $t-1$ ) and the contribution to ( $t-1$ ). This regularized total volume varies between 1 and 17  $\text{hm}^3$ . The optimal release (time  $t$ ) is not (0.89  $\text{hm}^3/\text{month}$ ). The second (2nd) plan a regularized total volume (stock + contribution to the  $t-1$ ) is equal to 18  $\text{hm}^3$ , optimal release, in this case, equals (1.39  $\text{hm}^3/\text{month}$ ). The third (3rd) plan corresponds to a regularized total volume (stock + contribution) equal to 19  $\text{hm}^3$ . The drop optimal equals (1.89  $\text{hm}^3/\text{month}$ ) for this volume. al to 20  $\text{hm}^3$ . The optimal release do not exceed 2.39  $\text{hm}^3/\text{month}$ ). The same for the fifth (5th) plan, giving optimal equal to (3.39  $\text{hm}^3/\text{month}$ ), for a volume regularized ( $t-1$ ) equal to 21  $\text{hm}^3$ . The optimal releases reach a volume monthly equal to (3.89  $\text{hm}^3/\text{month}$ ) for regularized volumes (stock + contribution) at ( $t-1$ ) vary between (22 and 30  $\text{hm}^3$ ), these results are represented by the 6th plan. The seventh (7th) plan is an interval regularized volumes vary between 31 and 95  $\text{hm}^3$ , releases in case reach (4.28  $\text{hm}^3/\text{month}$ ). The latest plan is an interval of regularized (stock + contribution to  $t-1$ ) volumes ranging from 95 to 155  $\text{hm}^3$ . The optimal volume of release reached a volume of 4.53  $\text{hm}^3/\text{month}$ . This latest plan is a total satisfaction of the water needs of consumers.

The cutting of the tank in three (03) levels of storage volume (0–17  $\text{hm}^3$ ), (18–30  $\text{hm}^3$ ) and (31–153  $\text{hm}^3$ ) is performed at the end of the optimal release plans. The regression shows independence from monthly applications. For this purpose, the optimal release can be described by the combination of contributions and the State of the storage only. The correlation coefficients are relatively low for the first (1st) and the second (2nd) interval of storage with values 0.61 and 0.52, respectively. While the correlation coefficient is relatively satisfying for the third interval of volume (31–153  $\text{hm}^3$ ) with a value of 0.74.

Multiple regression of the independent variables (monthly liquid intake, the State of the storage in the reservoir and monthly water demand) for the whole of regularizable volume (153  $\text{hm}^3$ ) shows a low regression as a ( $R^2 = 0.31$ ).

## 5 Conclusions

In this study, a test of elaboration of the optimal management rules of Hammam Boughrara dam is made. In the first part, the system (inputs–tank–consumer) is exposed. The study area is presented with some brief details of the watershed and reservoir characteristics. An analysis of monthly liquid flows to the reservoir with the loss of water by evaporation is by infiltration are described. End to simplify the

integration of these losses, the variation of these last is considered monthly. The monthly water use by consumers three (03) of the water in the tank (AEP of the cities of Maghnia and Oran thus irrigation perimeter of Maghnia) is detailed.

In the second part, the objective function (optimization criterion) is chosen. It is to minimize the absolute difference between releases and applications. This feature allows to satisfy the demands in the first place and streamline the management of the reservoir and maximize storage. Stochastic dynamic programming (SDP) is selected to develop the rules of management of the tank. Stochasticity of the contributions is introduced through the distribution of probability (frequency) of the contributions.

These are discretized into 37 equal classes. Each class corresponds to  $(1 \text{ hm}^3)$ . Similarly, the tank is discretized in 153 equal units of volume  $(1 \text{ hm}^3)$ . Optimization is performed over the period (2012–2030) with a monthly no. Optimal management plans are obtained for each month of the period of optimization. An analysis by multiple regression shows that a cutting volume of storage in three (03) areas to describe linear equations connecting the inputs, the State of the storage and water demand. The coefficients of the regression equations show that the optimal release can be described by the combination of liquid flows and the State of the storage in the tank only.

## References

1. A.N.B.T. (2012a). *Etude d'impact sur l'environnement du barrage hammam boughrara: RAPPORT FINAL, VOLUME I. ALGERIE: AGENCE NATIONALE DES BARRAGES ET DES TRANSFERTS.*
2. A.N.B.T. (2012b). *Rapport d'exploitation du barrage Hammam Boughrara—Document interne.* Algérie: Agence Nationale des Barrages et des Transferts (ANBT).
3. Asmadi, A., El-Shafie, A., Razali, S. F., & Mohamad, Z. S. (2014). Reservoir optimization in water resources: A review. *Water Resources Management*, 28, 3391. <https://doi.org/10.1007/s11269-014-0700-5>.
4. Bellman, R. (1957). *Dynamic programming.* New Jersey (USA): Defense Technical Information Center, princeton University Press édition.
5. Bouzid-Lagha, S., Djelita, B. (2012). Etude du phénomène d'eutrophisation dans le Barrage de Hammam Boughrara (Wilaya de Tlemcen, Algérie). *51(1)*, 186–201. doi: <https://doi.org/10.1080/02626667.2011.634417>.
6. Tlemcen, DRE-W. (2012a). *Données de régime de consommation d'eau potable de la ville de Maghnia (W. Tlemcen).* Tlemcen: Direction des Ressources en Eau de la Wilaya de Tlemcen.
7. Tlemcen, DRE-W. (2012b). *Données de régime de d'irrigation de périmètre de Maghnia (W. Tlemcen).* Tlemcen: Direction des Ressources en Eau de la Wilaya de Tlemcen.
8. EL Haouati, G. M. (November 2013). Effet de l'eutrophisation sur les milieux aquatiques: Barrage de Hammam Boughrara, Tlemcen. *USTHB-FBS-4th International Congress of the Populations & Animal Communities "Dynamics & Biodiversity of the terrestrial & aquatic Ecosystems (CIPCA4)"*, (pp. 223–231). TAGHIT (Bechar), ALGERIA.
9. Kelman, J., Stedinger, J. R., Cooper, L. A., Hsu, E., & Yuan, S.-Q. (1990). Sampling stochastic dynamic programming applied to reservoir operation. *Water resources Research*, 26(3), 447–454. doi:<https://doi.org/10.1029/WR026i003p00447>.

10. Labadie, J. W. (2004). Optimal operation of multireservoir systems: State-of-the-art review. *Journal of Water Resources Planning and Management*, 93–111.
11. l'Eau, I. M. (Atelier Régional 13 et 14 décembre 2010). RAPPORT REGIONAL SUR LA GESTION DES SEDIMENTS DANS LES BARRAGES EN MEDITERRANEE. *CONCEPTION ET GESTION DURABLE DES BARRAGES EN MEDITERRANEE* (pp. 1–245). Marseille (France): Institut Méditerranéen de l'Eau (IME).
12. Moran, P. (1954). *The theorie of storage*. New York (USA): J Wiley.
13. Selka, G. D. Z. M. (2007). Evaluation des effets environnementaux du oued Mouilah sur la Pérennité du Barrage Hammam Boughrara. *25e rencontres de l'AUGC, 23–25 mai 2007*. Bordeaux.
14. Barnett, T. P., Adam, J. C. (November 2005). Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature*, 438(17), 303–309.
15. Wurbs, R. A. (1993). Reservoir-system simulation and optimization models. *Water Resources Planning and Management Journal*, 119(4), 455–472, [https://doi.org/10.1061/\(ASCE\)0733-9496](https://doi.org/10.1061/(ASCE)0733-9496).
16. Yakowitz, S. (1982). Dynamic programming applications in water resources. *Water Resources Research*, 18(4), 673–696. <https://doi.org/10.1029/WR018i004p00673>.
17. Yeh, W. W.-G. (1985). Reservoir management and operations models: A state-of-the-art review. *Water Resources Reaserch*.
18. Zéphyr, L. (2015). *Optimisation stochastique des systèmes multi-réservoirs par l'agrégation de scénarios et la programmation dynamique approximative*. Québec, Canada: Doctorat en sciences de l'administration—Opérations et systèmes de décision—Université de Laval.