République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Médéa
Faculté des Sciences et de la Technologie
Département de Génie des Procédés et Environnement
Laboratoire de Biomatériaux et Phénomènes de Transport (LBMPT)

THESE

Présentée par :

Latifa KHAOUANE

En vue de l'obtention du titre de Doctorat en Génie des Procédés

Thème

ETUDE ET MODELISATION DE LA BIOSYNTHESE DES ANTIBIOTIQUES A PARTIR DE DIFFERENTES SOUCHES PRODUCTRICES - CAS DE PLEUROMUTILINE

Soutenue le 16 novembre 2013, devant le jury composé de :

M. Mustapha OUMOUNA	Professeur, Université de Médéa	Président
Mme Zoubida BENDJAMAA	Professeur, USTHB, Alger	Examinatrice
M. Mustapha DOUANI	Maître de Conférences (A), UHBC, Chlef	Examinateur
M. Salah HANINI	Professeur, Université de Médéa	Directeur de thèse
M. Othmane BENKORTBI	Maître de Conférences (A), Université de Médéa	Invité

الحمد لله كما ينبغي

لجلال وجهه و عظيم سلطانه

Dédicaces

A *mes chers parents,*

A *ma sœur Affaf,*

A *mon frère Djaafer et sa femme Leila,*

A *ma nièce Meriem et mon neveu Mohamed-Amine,*

A *tous ceux qui m'ont soutenu de près ou de loin.*

Latifa.

Remerciements et gratitude

En tout premier lieu, parce que la tradition l'exige certes, mais surtout parce qu'il le mérite, je tiens à remercier mon directeur de thèse, Monsieur **Salah HANINI**, *Professeur à l'UYFM, qui m'a guidé avec grande patience tout au long de l'élaboration de ce travail, pour ses aides précieuses qui ont judicieusement éclairé mon chemin vers l'aboutissement et la réussite de la concrétisation de ma thèse de Doctorat.*

Je tiens à exprimer mes plus vifs remerciements à Monsieur **Othmane BENKORTBI**, *Maître de Conférences (A) à l'UYFM et à Monsieur* **Chérif SI-MOUSSA,** *Maître de Conférences (B) à l'UYFM, pour leurs aides, leurs accompagnements et leurs éclairages techniques. Je leurs dois les remerciements les plus sincères.*

J'adresse mes remerciements à toute l'équipe du Laboratoire de Microbiologie Industrielle de SAIDAL de Médéa.

Je profite de l'occasion de la présentation de ce travail pour exprimer mes vifs remerciements à Monsieur **Mustapha OUMOUNA**, *Professeur à l'UYFM, pour avoir accepté de présider le jury de soutenance.*

Je tiens également à présenter ma profonde gratitude à Madame **Zoubida BENDJAMAA**, *Professeur à l'USTHB et à Monsieur* **Mustapha DOUANI**, *Maître de Conférences (A) à l'UHBC, d'avoir accepté de juger ce modeste travail.*

Merci à tous ceux qui m'ont aidé de près ou de loin durant ces années de recherche et d'études. Je ne peux conclure sans un mot de remerciement à tous mes enseignants et à tous mes collègues.

Résumé

Etude et Modélisation de la Biosynthèse des Antibiotiques à partir de Différentes Souches Productrices – Cas de Pleuromutiline

L'objectif de cette thèse vise à développer des méthodologies basées sur les réseaux de neurones pour la modélisation des bioprocédés et ceux pour un but de prédiction et d'optimisation.

Notre première contribution a porté sur l'utilisation des réseaux de neurones pour la modélisation du processus de fermentation fed-batch de pleuromutiline et ceux dans le but de prédire l'évolution des paramètres difficilement mesurables on line (biomasse, substrat et produit) à partir de variables facilement mesurables. Pour cela, trois modèles neuronaux de type feedforward ont été construits et optimisés. Ces modèles se caractérisent par une structure similaire : cinq neurones dans la couche d'entrée (temps de fermentation, *pH*, indice de réfraction et concentration initiale d'inoculum et concentration initiale du substrat), une couche cachée avec un nombre de neurones optimisé au cours du processus d'apprentissage et un neurone dans la couche de sortie (biomasse, consommation du sucre ou formation du produit). Une bonne concordance entre les valeurs expérimentales et les valeurs prédites pour chaque modèle neuronal a été obtenue (les racines des erreurs quadratiques moyennes pour la base de test étaient respectivement de 0,4624%, de 0,1234 g/L et de 0,0016 mg/g pour le réseau de neurones décrivant la biomasse, la consommation du substrat et la formation du produit). Par ailleurs, la comparaison entre les modèles neuronaux optimisés et les modèles cinétiques non structurés a démontré la supériorité des modèles neuronaux. En plus de l'excellente corrélation des données expérimentales, les trois modèles neuronaux utilisés ont montré une bonne capacité prédictive d'interpolation.

Notre deuxième contribution a porté sur le développement d'une méthodologie d'optimisation, combinant un plan composite centré (PCC), un réseau de neurones artificiel (RN) et un algorithme d'optimisation par essaim de particules (OEP). Cette technique a été appliquée à l'optimisation des conditions de culture de *Pleurotus mutilus* (vitesse d'agitation, température et *pH*) et ceux dans le but d'obtenir une production maximale en pleuromutiline. Le PCC et le RN ont été utilisés respectivement pour planifier et concevoir l'expérience et pour modéliser le processus. L'algorithme d'OEP a été appliqué pour deux fins : remplacement de la rétro-propagation standard dans l'apprentissage du réseau de neurones et l'optimisation du modèle neuronal obtenu. La comparaison entre le modèle neuronal formé avec l'algorithme d'OEP (RNOEP), le réseau de neurones formé avec l'algorithme de régularisation bayésienne (RNRB) et le modèle quadratique de la méthodologie des surfaces de réponse (MSR) a démontré la grande capacité de modélisation des modèles neuronaux. Par la suite, un algorithme générique d'OEP a été utilisé pour optimiser l'espace d'entrées des modèles neuronaux (RNOEP et RNRB). En vertu de l'optimisation du modèle RNOEP, les niveaux optimaux des conditions de culture obtenus sont: une vitesse d'agitation de 242 tr/min, une température de 26,88 °C et un *pH* de 6.06. Par conséquent, une production de 10074 ± 500 µg/g qui est en très bon accord avec la prédiction (10149 µg/g), a été observée dans l'expérience de vérification. L'optimisation du modèle RNOEP (10074 µg/g) a donné un rendement de 27,5 % supérieure à celui obtenue par l'optimisation du modèle RNRB (7900 µg/g) démontrant ainsi la supériorité de la méthodologie développé.

Mots clés : Fermentation, Pleuromutiline, Modélisation, Prédiction, Optimisation, Réseau de Neurones, Algorithme d'Optimisation par Essaim de Particules.

Summary

Study and Modeling of the Biosynthesis of Antibiotics from Different Producing Strains – Case of Pleuromutilin

The objective of this thesis is to develop methodologies based on neural networks for modeling bio processes and for those purposes of prediction and optimization.

Our first contribution is focused on the use of neural networks for modeling the process of fed-batch fermentation of pleuromutilin and those in order to predict the evolution of parameters that are difficult to measure online (biomass, substrate and product) from easily measurable variables. For this purpose, three feed-forward neural models were built and optimized. These models are characterized by a similar structure: five neurons in the input layer (fermentation time, *pH*, refractive index, initial inoculum concentration and initial substrate concentration), a hidden layer with a number of neurons optimized during the learning process, and one neuron in the output layer (biomass, sugar consumption, or product formation). For each model, a good agreement between experimental and predicted values was obtained (root mean square errors for the test set were 0.4624%, 0.1234 g/L and 0.0016 mg/g for the neural network describing biomass, substrate consumption and product formation, respectively). Moreover, the comparison between the optimized neural models and the unstructured kinetic models showed the superiority of neural models. In addition to the excellent correlation of the experimental data, the three neural models used showed good predictive ability of interpolation.

Our second contribution is focused on the development of an optimization methodology, combining central composite design (PCC), artificial neural network (RN) and particle swarm optimization algorithm (OEP). This technique has been applied in optimizing the culture conditions of *Pleurotus mutilus* (stirring speed, temperature and *pH*) and those in order to obtain maximum pleuromutilin production. The PCC and the RN were respectively used to plan and design experience and to model the process. The OEP algorithm was used for two purposes: replacing the standard back-propagation in the learning of neural network and optimization of the obtained neural model. The comparison between the neural model trained with OEP algorithm (RNOEP), the neural network trained with Bayesian regularization algorithm (RNRB) and the quadratic model of response surface methodology has demonstrated the modeling capacity of neural models. Subsequently, a generic OEP algorithm was used to optimize the input space of neural network models (RNOEP and RNRB). Under the optimization of RNOEP model, the optimal levels obtained of culture conditions are: 242 rpm for the agitation speed, a temperature of 26.88 ° C and a *pH* of 6.06. Consequently, a production of 10 074 ± 500 µg/g, which is in good agreement with the prediction (10149 µg/g) was observed in the verification experiment. The optimization of the RNOEP model (10074 µg/g) gave a yield of 27.5% higher than that obtained by optimizing the RNRB model (7900 µg/g), thus demonstrating the superiority of the developed methodology.

Keywords: Fermentation, Pleuromutilin, Modeling, Prediction, Optimization, Neural Network, Particle Swarm Optimization Algorithm.

ملخص

دراسة و نمذجة التركيب الحيوي للمضادات الحيوية من مختلف السلالات المنتجة – حالة البلورموتيلين

الهدف من هذه الرسالة هو وضع منهجيات على أساس الشبكات العصبية لنمذجة العمليات البيولوجية وذلك لغرض التنبؤ والتحسين.

ركزت مساهمتنا الأولى على استخدام الشبكات العصبية لنمذجة عملية تخمير البلورموتيلين وذلك من أجل تنبؤ تطور المعلمات التي يصعب قياسها (الكتلة الحيوية، السكر و المنتج) من المتغيرات القابلة للقياس بسهولة. لهذا الغرض ، تم بناء و تحسين ثلاثة نماذج عصبية نوع التغذية-إلى-الأمام. وتتميز هذه النماذج من بنية مماثله : خمسة عصبونات في طبقة المدخلات (وقت التخمير، درجة الحموضة، معامل الانكسار، اللقاح الأولي والتركيز الأولي من السكر)، طبقة مخفية مع عدد عصبونات تم تحسينها أثناء عملية التعلم، وعصبون واحد في طبقة الإخراج (الكتلة الحيوية، استهلاك السكر أو تشكيل المنتج). تم الحصول على اتفاق جيد بين القيم التجريبية والمتوقعة لكل نموذج من النماذج العصبيه (جذر الاخطاء التربيعية المتوسطة لمجموعة الاختبار كانت على التوالى 0.4624 ٪، 0.1243 غ/ل، 0.0016 مغ/غ للشبكة العصبية التى تصف الكتلة الحيوية، استهلاك السكر و تشكيل المنتج) . وعلاوة على ذلك، فإن المقارنة بين النماذج العصبية الأمثل والنماذج الحركية غير المهيكلة أظهرت تفوق النماذج العصبية. بالإضافة إلى الترابط الممتاز في البيانات التجريبية، أظهرت النماذج العصبية الثلاثة المستخدمة تنبؤات جيدة فى الاستيفاء.

ركزت مساهمتنا الثانية على وضع منهجية تحسين تجمع بين التصميم المركزي المركب (PCC)، الشبكة العصبية الاصطناعية (RN) وخوارزمية التحسين بسرب الجسيمات (OEP). وقد تم تطبيق هذه التقنية في تحسين الظروف التخميرية *للبلوروتيس موتيليس* (سرعة التحريك، درجة الحرارة ودرجة الحموضة) و ذلك من أجل الحصول على أقصى إنتاج من البلورموتيلين. قد تم استخدام PCC وRN من أجل تصميم التجارب و تصميم النموذج. قد تم استخدام خوارزمية OEP لغرضين : استبدال الانتشار القياسي في تعلم الشبكة العصبية وتحسين نموذج الخلايا العصبية المتحصل عليه. المقارنة بين نموذج الخلايا العصبيه المتدرب مع خوارزمية OEP (RNOEP)، نموذج الشبكة العصبية المتدرب مع تنظيم الخوارزمية النظرية الافتراضية (RNRB) ونموذج الدرجة الثانية من منهجية سطوح الاستجابة أثبتت قدرة نمذجة الشبكات العصبية. وبعد ذلك، تم استخدام خوارزمية OEP العامة لتحسين فضاء مدخلات النماذج العصبية (RNOEP وRNRB). تحت تحسين نموذج RNOEP المستويات المثلى للشروط التخميرية المتحصل عليها هى : سرعة التحريك 242 حركة/دقيقة، درجة حرارة 26.88 درجة مئوية و درجة الحموضة 6.06. و عليه إنتاج بقدر 10074 ± 500 ميكروغرام/غرام ، وهو ما يتفق جيدا مع التنبؤ (10149 ميكروغرام/غرام) لوحظ في تجربة التحقق. أعطى تحسين نموذج RNOEP (10074 ميكروغرام/غرام) عائدا بقدر 27.5 ٪ أعلى من المتحصل عليه عن طريق تحسين نموذج RNRB (7900 ميكروغرام/غرام) مما يدل على تفوق المنهجية المتقدمة.

كلمات مفاتيح : التخمير، البلورموتيلين، النمذجة ، التنبؤ، التحسين، الشبكات العصبية، خوارزمية التحسين بسرب الجسيمات.

Table des matières

Table des matières i
Abréviations iv
Nomenclature et symboles vi
Liste des tableaux x
Liste des figures xi
Liste des travaux et activités de recherches xiv

Introduction générale 1

Chapitre I. Les réseaux de neurones

I.1. Introduction 5
I.2. Historique 5
I.3. Notions de base sur les réseaux de neurones 6
I.3.1. Du neurone biologique au modèle mathématique 6
I.3.1.1. Le modèle neurophysiologique 6
I.3.1.2 Le modèle mathématique 8
I.3.2. Architecture des réseaux de neurones 10
I.3.2.1. Réseau de neurones non bouclé (en anglais feedforward) 10
I.3.2.2. Réseau de neurones bouclé (ou récurrent) 11
I.3.3. Types d'apprentissage des réseaux de neurones 12
I.3.3.1. Apprentissage supervisé 12
I.3.3.2. Apprentissage non supervisé 12
I.4. Les réseaux de neurones les plus utilisés 13
I.4.1. Le perceptron monocouche (perceptron simple) 13
I.4.2. Le perceptron multicouche (en anglais Multi Layer Perceptron) 13
I.4.3. Les réseaux à fonction radiale 13
I.4.4. Les cartes auto-organisatrices de Kohonen (SOM) 13
I.4.5. Les réseaux de Hopfield 14
I.4.6. Les réseaux ART 14
I.5. Conception d'un modèle neuronal 14
I.5.1. Détermination des entrées/sorties du réseau de neurones 15
I.5.2. Construction de la base de données et choix des échantillons 15
I.5.3. Définition de la structure du réseau 16
I.5.3.1. Nombre de couches cachées 16
I.5.3.2. Nombre de neurones cachés 16
I.5.4. Apprentissage 17
I.5.5. Evaluation des performances du réseau de neurones 20
I.6. Application des réseaux de neurones en bioprocédés 20
I.6.1. Modélisation et prédiction 21
I.6.2. Capteurs logiciels (en anglais soft sensors) 21
I.6.3. Reconnaissance des formes (en anglais pattern recognition) 22
I.6.4. Optimisation 22
I.6.5. Contrôle 23
I.7. Conclusion 23

Chapitre II. Optimisation par essaim de particules

II.1. Introduction 24
II.2. Développement historique 25
II.2.1. Modèle de Reynolds 25
II.2.2. Modèle de Kennedy et Eberhart 25
II.3. L'algorithme de base 26
II.3.1. Description mathématique 26
II.3.2. Les étapes de l'algorithme 28
II.3.3. Caractéristiques principales 29
II.4. Confinement des particules 30
II.5. Topologie du voisinage 30
II.6. Facteur d'inertie et variantes de l'algorithme d'OEP 31
II.6.1. Trelia de types 1 et 2 (réglage fixe) 32
II.6.2. Approche du poids d'inertie 32
II.6.3. Approche du facteur de constriction 32
II.7. Application de l'algorithme d'OEP en bioprocédés 33
II.7.1. Estimation de paramètres 33
II.7.2. Apprentissage des réseaux de neurones de type perceptron multicouches 34
II.7.3. Optimisation des modèles neuronaux 35
II.8. Conclusion 35

Chapitre III. Modélisation de la fermentation fed-batch de pleuromutiline par les réseaux de neurones

III.1. Introduction 36
III.2. Matériels et méthodes 37
III.2.1. Collecte des données, prétraitement et analyse 37
III.2.2. Procédure de développement des modèles 37
III.2.2.1. Description des modèles 37
III.2.2.2. Développement des modèles 39
III.2.3. Comparaison avec les modèles cinétiques non structurés 42
III.2.4. Analyse de sensibilité 44
III.2.4.1. Méthode de "Poids" 44
III.2.4.1. Méthode de "Profil" 45
III.3. Résultats et discussion 47
III.3.1. Performances des modèles 47
III.3.2. Temps d'exécution 52
III.3.3. Expressions mathématique des modèles neuronaux optimisés 52
III.3.4. Comparaison avec les modèles cinétiques non structurés 57
III.3.5. Performances d'interpolation 62
III.3.6. Analyse de sensibilité 65
III.2.6.1. Méthode de "Poids" 65
III.2.6.1. Méthode de "Profil" 66
III.4. Conclusion 72

Chapitre IV. Optimisation hybride des conditions de culture de *Pleurotus mutilus*

IV.1. Introduction 73
IV.2. Matériels et méthodes 73
IV.2.1. Souche 73
IV.2.2. Les milieux de culture 74
IV.2.2.1. Milieu liquide végétatif 74
IV.2.2.2. Milieu liquide de fermentation 74
IV.2.3. Les conditions de culture 75
IV.2.3.1. Germination 75
IV.2.3.2. Culture en batch 75
IV.2.4. Méthodes analytique 75
IV.2.4.1. Analyse de la biomasse 75
IV.2.4.2. Analyse du produit 76
IV.3. Description de la méthodologie hybride 76
IV.3.1. Plan composite centré (PCC) 76
IV.3.2. Modélisation prédictive : Description du modèle neuronal RNOEP 78
IV.3.2.1.Comparaison avec le réseau de neurones formé avec l'algorithme de régularisation bayésienne (RNRB) 81
IV.3.2.2. Comparaison avec la méthodologie de surface de réponse (MSR) 81
IV.3.2.3. Evaluation des performances de prédiction des modèles 81
IV.3.3. Optimisation par essaim de particules 82
IV.4. Logiciels utilisés 82
IV.5. Résultats et discussion 83
IV.5.1. Modélisation par le RNOEP 83
IV.5.2. Modélisation par le RNRB 87
IV.5.3. Modélisation par la MSR 90
IV.5.4. Comparaison des modèles prédictives 95
IV.5.5. Optimisation des modèles neuronaux 100
IV.6. Comparaison indirect avec d'autres méthodologies d'optimisation à base des RN 104
IV.7. Conclusion 107

Conclusion générale et perspectives 108
Références bibliographiques 112
Annexe 119

Abréviations

AA	**A**lgorithme d'**A**pprentissage
AG	**A**lgorithme **G**énétique
ART	**A**daptive **R**esonance **T**heory
DS	**D**éviation **S**tandard
EPM	**E**rreur de **P**rédiction du **M**odèle
ESP	**E**rreur **S**tandard de **P**rédiction
FTCC	**F**onction de **T**ransfert de la **C**ouche **C**achée
HPLC	**H**igh **P**erformance **L**iquid **C**hromatography
IN	**I**ntervalle de **N**ormalisation
MC	**M**odèle **C**inétique
MC*P*	**M**odèle **C**inétique décrivant la formation du **P**roduit
MC*S*	**M**odèle **C**inétique décrivant la consommation du **S**ubstrat
MC*X*	**M**odèle **C**inétique décrivant la formation de la biomasse (***X***)
MSR	**M**éthodologie de **S**urface de **R**éponse
OEP	**O**ptimisation par **E**ssaim de **P**articules
PCC	**P**lan **C**omposite **C**entré
PMC	**P**erceptron **M**ulti-**C**ouche
P-value	**P**robabilité
REQM	**R**acine de l'**E**rreur **Q**uadratique **M**oyenne
RFR	**R**éseau à **F**onctions de base **R**adiales
RN	**R**éseaux de **N**eurones
RNFF	**R**éseau de **N**eurones de type **F**eed **F**orward
RNO	**R**éseau de **N**eurones **O**ptimal
RNOEP	**R**éseau de **N**eurones formé avec l'algorithme d'**O**ptimisation par **E**ssaim de **P**articules
RNOEP-OEP	**O**ptimisation par **E**ssaim de **P**articules du **RNOEP**
RN*P*	**R**éseau de **N**eurones décrivant la formation du **P**roduit

RNRB	**R**éseau de **N**eurones formé avec l'algorithme de **R**égularisation **B**ayésienne
RNRB-OEP	**O**ptimisation par **E**ssaim de **P**articules du **RNRB**
RN*S*	**R**éseau de **N**eurones décrivant la consommation du **S**ubstrat
RN*X*	**R**éseau de **N**eurones décrivant la formation de la biomasse (***X***)

Nomenclatures et symboles

Lecture	Désignation	Unité
A_f	Facteur de précision	(-)
B_f	Facteur de biais	(-)
b	Biais d'un neurone artificiel	(-)
c_i	Coefficients d'accélération	(-)
E_i	$i^{ème}$ entrée du modèle	(-)
F	Débit d'alimentation	l/h
IR	Indice de réfraction du bouillon de fermentation	(-)
Ir	Importance relative	%
$Iter$	Nombre d'itération actuel	(-)
$Iter_{max}$	Nombre d'itération maximal	(-)
k_i	Constante d'inhibition	g/l
k_S	Constante de Monod	g/l
L	Nombre de couche cachée	(-)
m_S	Coefficient de maintenance	g de substrat/ (g de biomasse h)
N	Nombre de neurones	(-)
N_1	Nombre de neurones dans la $1^{ère}$ couche cachée	(-)
N_2	Nombre de neurones dans la $2^{ème}$ couche cachée	(-)
n	Nombre d'expériences ou de points	(-)
P	Concentration en pleuromutiline	mg de produit/g de biomasse
pH	Mesure de l'acidité ou la basicité du bouillon de fermentation	(-)
R	Coefficient de corrélation	(-)
S	Concentration en substrat	g/l
S_0	Concentration en glucose initiale	g/l

T	Température	°C
t	Temps de fermentation	h
V	Volume du bioréacteur industriel (culture)	l
V_i	Vitesse actuelle de la i^{ème} particule	(-)
V_{max}	Vitesse maximale seuil	(-)
V_{min}	Vitesse minimale seuil	(-)
w	Poids synaptique des connexions inter-neuronales	(-)
w	Poids d'inertie	(-)
w_{max}	Poids d'inertie initiale	(-)
w_{min}	Poids d'inertie finale	(-)
X	Concentration en biomasse	%
X_0	Concentration d'inoculum initiale	%
X_1	Vitesse d'agitation	tr / min
X_2	Température	°C
X_3	*pH*	(-)
X_i	Position actuelle de la i^{ème} particule	(-)
X_{max}	Position maximale seuil	(-)
X_{min}	Position minimale seuil	(-)
Y	Rendement de production	µg de produit/g de biomasse
$Y_{X/S}$	Rendement de la biomasse par unité de masse de substrat	g de biomasse/g de substrat
$Y_{P/S}$	Rendement du produit par unité de masse de substrat	g de produit/g de substrat
Z_j	j^{ème} sortie de la couche cachée	(-)

Lettres Grecques

Lecture	Désignation	Unité
α	Pente de la droite du meilleur ajustement linéaire	(-)
β	Ordonnée à l'origine de la droite du meilleur ajustement linéaire	(-)
β_o	Constante représentant l'effet moyen général	(-)
β_i	Coefficients de régression linéaire	(-)
β_{ij}	Coefficients de régression associés aux termes d'interaction	(-)
β_{ii}	Coefficients de régression quadratique	(-)
μ	Vitesse spécifique de croissance	h^{-1}
μ_{max}	Valeur maximale de μ	h^{-1}
π	Vitesse spécifique de formation de produit	mg de produit / (g de biomasse. h)
π_{max}	Valeur maximale de π	mg de produit / (g de biomasse. h)

Exposants

Lecture	Désignation
cal	Calculée
exp	Expérimentale
H	Cachée (Hidden)
I	Entrée (Input)
O	Sortie (Output)

Indices

Lecture	Désignation
H	Cachée (Hidden)
i	Elément i d'un vecteur
i,exp	Elément i d'un vecteur de valeurs expérimentales
i,pred	Elément i d'un vecteur de valeurs prédites
ij	Croisement de la ligne i avec la colonne j d'une matrice
j	Elément j d'un vecteur
max	Maximum
min	Minimum
N	Normalisé
O	Sortie (Output)

Liste des tableaux

Chapitre I. Les réseaux de neurones

Tableau I.1	Fonctions de transfert usuelles, $a=f(n)$.	9
Tableau I.2	Résumé des caractéristiques des algorithmes d'apprentissage de rétro-propagation implantés dans MATLAB®	19

Chapitre III. Modélisation de la fermentation fed-batch de pleuromutiline par les réseaux de neurones

Tableau III.1	Matrice de corrélations entres les outputs (*X*, *S* et *P*).	38
Tableau III.2	Analyse statistique des données d'entrée et de sortie.	39
Tableau III.3	Performances des modèles neuronaux et leurs sous-modèles pour l'ensemble de test.	48
Tableau III.4	Structure des modèles neuronaux optimisés.	49
Tableau III.5	Vecteurs de régression linéaire issue de l'équation linéaire : $y^{cal} = \alpha y^{exp} + \beta$, [$\alpha$ (pente), β (ordonnée à l'origine), R (coefficient de corrélation)].	51
Tableau III.6	Temps d'exécution.	52
Tableau III.7	Poids et biais du RN*X* optimal.	54
Tableau III.8	Poids et biais du RN*S* optimal.	55
Tableau III.9	Poids et biais du RN*P* optimal.	56
Tableau III.10	Comparaison, en termes de la *REQM* et l'*ESP*, entre les réseaux de neurones optimisés (RN) et les modèles cinétiques (MC).	61
Tableau III.11	Résultats d'interpolation, en termes de la *REQM* et du coefficient de corrélation (*R*).	62

Chapitre IV. Optimisation hybride des conditions de culture de *Pleurotus mutilus*

Tableau IV.1	Composition du milieu liquide végétatif.	74
Tableau IV.2	Composition du milieu liquide de fermentation.	74
Tableau IV.3	Domaine expérimental et niveaux de chaque facteur.	77
Tableau IV.4	Matrice d'expérimentation et rendement expérimental.	78
Tableau IV.5	Structure des réseaux de neurones optimisés (RNOEP et RNRB).	84
Tableau IV.6	Paramètres de commande utilisés pour l'exécution de la variante du facteur de constriction de l'algorithme d'OEP lors de l'apprentissage du RNOEP.	84
Tableau IV.7	Poids et biais du RNOEP optimal.	84
Tableau IV.8	Poids et biais du RNRB optimal.	88
Tableau IV.9	Coefficients du modèle estimé par analyse de régression linéaire multiple.	91
Tableau IV.10	Temps d'exécution.	95
Tableau IV.11	Rendement expérimental et prédit par les modèles.	97
Tableau IV.12	Vecteurs de régression linéaire issue de l'équation linéaire $y^{cal} = \alpha y^{exp} + \beta$, [$\alpha$ (pente), β (ordonnée à l'origine), R (coefficient de corrélation)].	98
Tableau IV.13	Résultats de l'optimisation.	102
Tableau IV.14	Description de différentes méthodologies d'optimisation à base des RN dans la littérature.	105
Tableau IV.15	Comparaison des résultats des différentes méthodologies d'optimisation à base des RN dans la littérature.	106

Liste des figures

Chapitre I. Les réseaux de neurones

Figure I.1	Schéma d'un neurone.	7
Figure I.2	Schéma d'une synapse.	7
Figure I.3	Neurone formel.	8
Figure I.4	Réseaux de neurones les plus utilisés.	10
Figure I.5	Structure d'un réseau de neurones non bouclé.	11
Figure I.6	Un réseau de neurones bouclé à deux entrées.	11

Chapitre II. Optimisation par essaim de particules

Figure II.1	Schéma du principe de déplacement d'une particule.	28
Figure II.2	Organigramme de la méthode d'OEP.	29
Figure II.3	Trois topologies du voisinage.	31
Figure II.4	Phases du processus de modélisation en biotechnologie.	34

Chapitre III. Modélisation de la fermentation fed-batch de pleuromutiline par les réseaux de neurones

Figure III.1	Organigramme pour le développement et l'optimisation de l'architecture d'un RNFF.	41
Figure III.2	Représentation des % de division de la base de données.	42
Figure III.3	Organigramme de la méthode de "poids".	46
Figure III.4	Schéma explicatif de la méthode de "Profil".	47
Figure III.5	Performances des modèles neuronaux (RN*X*, RN*S* et RN*P*) et leurs sous-modèles (sm1, sm2, sm3 et sm4) pour la base de test.	49
Figure III.6	Comparaison entre les valeurs expérimentales et calculées par les modèles neuronaux optimisés pour la totalité de l'ensemble de données : **(a)** biomasse, **(b)** glucose, **(c)** produit.	51
Figure III.7	Représentation schématique du RN*S* optimisé.	53
Figure III.8	Résultats de modélisation: **(a)** lot N°2, **(b)** lot N°4, **(c)** lot N°9, **(d)** lot N°17. (Les données expérimentales (symboles) pour la biomasse (●), glucose (■) et pleuromutiline (▲); Les résultats de la modélisation par RNFF sont représentés par des traits pleins et les résultats de la modélisation cinétique sont présentés par des lignes en pointillés).	59
Figure III.9	Comparaison entre les réseaux de neurones optimisés (RN) et les modèles cinétiques (MC) pour la totalité de la base de données : **(a)** en termes de la *REQM*, **(b)** en termes de l'*ESP*.	60
Figure III.10	Représentation des résultats de l'interpolation : **(a)** en termes de la *REQM*, **(b)** en termes du coefficient de corrélation (*R*).	63
Figure III.11	Résultats de simulation de la première base de données d'interpolation. (Les données interpolées expérimentales (symboles) pour la biomasse (○), glucose (□) et pleuromutiline (△); Les résultats de la modélisation par RNFF sont représentés par des traits pleins).	64

Figure III.12 Résultats de simulation de la seconde base interpolation: X_0=3% et S_0=23.5g/L. (Les données expérimentales Interpolé (symboles) pour la biomasse (○), glucose (□) et pleuromutiline (△); Les résultats de modélisation par RNFF sont représentés par des traits pleins). 64

Figure III.13 Présentation des % de l'importance relative (*Ir*) de chaque variable d'entée sur la biomasse (RN*X*), sur la consommation du sucre (RN*S*) et sur le produit (RN*P*). 65

Figure III.14 Graphes du profil de contributions des variables simulées (*t, pH, IR,* X_0 et S_0) sur la biomasse prédite par RN*X* : **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de X_0, **(e)** effet de S_0. 68

Figure III.15 Graphes du profil de contribution des variables simulées (*t, pH, IR,* X_0 et S_0) sur le substrat prédit par RN*S*: **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de X_0, **(e)** effet de S_0. 69

Figure III.16 Graphes du profil de contribution des variables simulées (*t, pH, IR,* X_0 et S_0) sur le produit prédit par RN*P*: **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de X_0, **(e)** effet de S_0. 70

Figure III.17 Présentation des % de contribution relative de chaque variable d'entrée sur la biomasse (RN*X*), sur la consommation du sucre (RN*S*) et sur le produit (RN*P*). 71

Chapitre IV. Optimisation hybride des conditions de culture de *Pleurotus mutilus*

Figure IV.1 Tubes coniques pour sédimentation. 76

Figure IV.2 Représentation des hauteurs L_1, L_2 et L_{max}. 76

Figure IV.3 Procédure de développement du RNOEP et du RNRB. 80

Figure IV.4 Représentation schématique du modèle neuronal (RNOEP ou RNRB). 85

Figure IV.5 Présentation des % de l'importance relative (*Ir*) (obtenus par la méthode de "poids") et des % de contribution relative (obtenus par la méthode de "profil") de chaque variable d'entée du RNOEP sur le rendement. 86

Figure IV.6 Graphes du profil de contributions des variables simulées (Vitesse d'agitation, température, *pH* initial) sur le rendement prédit par le RNOEP : **(a)** effet de la vitesse d'agitation, **(b)** effet de la température, **(c)** effet du *pH* initial. 87

Figure IV.7 Présentation des % de l'importance relative (*Ir*) (obtenus par la méthode de "poids") et des % de contribution relative (obtenus par la méthode de "profil") de chaque variable d'entée du RNRB sur le rendement. 89

Figure IV.8 Graphes du profil de contributions des variables simulées (Vitesse d'agitation, température, *pH* initial) sur le rendement prédit par le RNRB: **(a)** effet de la vitesse d'agitation, **(b)** effet de la température, **(c)** effet du *pH* initial. 90

Figure IV.9 Tracés de surfaces de réponses et de contours de la MSR : **(a)** effet de la température et de la vitesse d'agitation, **(b)** effet du *pH* initial et de la vitesse d'agitation, **(c)** effet du *pH* initial et de la température. 94

Figure IV.10 Comparaison entre les valeurs expérimentales et calculées par les modèles (MSR, RNRB et RNOEP) pour la totalité de l'ensemble de données. 96

Figure IV.11 Comparaison entre les modèles : **(a)** en termes de la *REQM*, **(b)** en termes des % d'erreurs (*ESP* et *EPM*), **(c)** en termes des facteurs A_f et B_f. 99

Figure IV.12 Evolutions de la fonction de fitness meilleure et moyenne au cours de la méthodologie "RNRB-OEP". 101

Figure IV.13 Evolutions de la fonction de fitness meilleure et moyenne au cours de la méthodologie hybride "RNOEP-OEP". 102

Figure IV.14 Comparaison entre les deux méthodologies (RNOEP-OEP et RNRB-OEP) : **(a)** en termes de rendement prédit et expérimental, **(b)** en termes de l'*EPM*. 103

Figure IV.15 Comparaison, en termes des % d'erreurs (*ESP* et *EPM*) de la phase de modélisation, de différentes méthodologies d'optimisation à base des RN dans la littérature. 106

Figure IV.16 Comparaison, en termes de l'*EPM* de la phase d'optimisation, des différentes méthodologies d'optimisation à base des RN dans la littérature. 107

Annexe

Figure 1 Résultats de modélisation. (Les données expérimentales (symboles) pour la biomasse (●), glucose (■) et pleuromutiline (▲); Les résultats de la modélisation par RNFF sont représentés par des traits pleins et les résultats de la modélisation cinétique sont présentés par des lignes en pointillés). 126

Liste des travaux et activités de recherches

Publications

02. Khaouane L., Benkortbi O., Hanini S. et Si-Moussa C. **(2013)**. "Modeling of an industrial process of pleuromutilin fermentation using feed-forward neural networks." *Brazilian Journal of Chemical Engineering* **30**(1): 105-116. ISSN: 0104-6632.

01. Khaouane L., Si-Moussa C., Hanini S. et Benkortbi O. **(2012)**. "Optimization of culture conditions for the production of pleuromutilin from *Pleurotus mutilus* using a hybrid method based on central composite design, neural network, and particle swarm optimization." *Biotechnology and Bioprocess Engineering* **17**(5): 1048-1054. ISSN: 1226-8372.

Communications nationales avec comité de lecture

02. Khaouane L., Si-Moussa C., Hanini S. et Benkortbi O. **(2012)**. "Optimization of culture conditions for the production of pleuromutilin from *Pleurotus mutilus* using neural network and particle swarm optimization." *3ème Journée Scientifique sur le Génie des Procédés (JSGP3-2012*. Le 19 novembre 2012, Blida, Algérie.

01. Khaouane L., Hanini S., Si-Moussa C. et Benkortbi O. **(2009)**. " Modélisation paramétriques de la fermentation de pleuromutiline par réseau de neurones." *4ème Congrée Algérien de Génie des Procédés (CAGEP4), « Génie des Procédés, Innovation Industrielles et Développement Durable »*. Le 04-06 mai 2009, Alger, Algérie.

Communications internationales avec comité de lecture

01. Benkortbi O., **Khaouane L.,** Hanini S. et Bentahar F. **(2011)**. "Use of neural networks to predict the temperature profile in the biosynthesis of pleuromutilin antibiotic produced by *Pleurotus mutilus*." *4th International Conference on Approximation Methods and Numerical Modeling in Environment and Natural Resources (MAMERN11)*. Le 23-26 mai 2011, Saidia, Morocco.

Projet de recherche

01. Membre du projet CNEPRU intitulé "Application de l'intelligence computationnelle aux phénomènes de transports: Etudes, compréhension et application" (MESRS réf. J0102620110007 à partir du **01/01/2012**), Université de Médéa **(Chargé de recherche)**.

Introduction générale

Dans un marché mondial d'antibiotique de plus en plus concurrentiel et face aux contraintes économiques, environnementales et de sécurité, les industriels doivent augmenter leur production d'antibiotiques quantitativement et qualitativement et réduire leurs coûts et leur consommation d'énergie. Par conséquent, une modélisation du processus qui reste toujours un problème difficile en raison de la complexité et de la non-linéarité du phénomène s'avèrent nécessaire (Potocnik et Grabec 1999).

Les modèles peuvent être utilisés pour soutenir l'amélioration des bioprocédés par de nombreuses façons différentes : prédiction, contrôle, optimisation, capteurs logiciels ou reconnaissance de formes. Dans le présent travail l'intérêt est consacré aux aspects liés à la prédiction et à l'optimisation du processus de fermentation de la pleuromutiline, qui est un produit naturel, obtenu par la fermentation de *Pleurotus mutilus*, un champignon comestible. Ce composé est actif contre une variété de bactéries gram-positives et de mycoplasmes et est un inhibiteur de la synthèse protéique bactérienne (Kavanagh *et al.* 1951; Egger et Reinshagen 1976). D'autre part, la pleuromutiline a servi de structure principale dans le développement de plusieurs antibiotiques commerciaux comme tiamulin et valnemulin utilisé dans la médecine vétérinaire (Hannan *et al.* 1997) et retapamulin approuvé pour le traitement de problèmes de peau humaine (Hirokawa *et al.* 2008).

Dans l'approche conventionnelle de modélisation en bioprocédés, on trouve les modèles cinétiques. Ces modèles, connus comme une stratégie de boîte blanche, fournissent une expression analytique reliant les caractéristiques clés du système physique à son comportement dynamique. Dans la modélisation cinétique, les valeurs des paramètres du modèle ont une signification physique, contiennent toujours les conséquences des suppositions introduites et des simplifications du fait de la complexité du cycle de fermentation. Ces modèles ont été et sont encore largement utilisés dans les procédés de fermentation (Paul et Thomas 1996; Zangirolami *et al.* 1997; Cruz *et al.* 1999; Benkortbi *et al.* 2007). Toutefois, dans certains cas, ces modèles ne s'appliquent pas, en raison de la non-linéarité inhérente du système, le manque de données expérimentales, l'inexactitude expérimentale, ou les écarts aux conditions idéales (Feyo de Azevedo *et al.* 1997; Silva *et al.* 2008; Saraceno *et al.* 2009).

Dans l'approche courante d'optimisation en bioprocédés et à afin d'obtenir la meilleure performance des procédés de fermentation, des stratégies d'optimisations diverses de ces processus, allant de la méthode conventionnelle de manipulation séquentielle d'un facteur à la méthodologie des surfaces de réponse (MSR), ont été générées par les scientifiques. La méthode conventionnelle

d'optimisation par manipulation séquentielle d'un seul paramètre est non seulement laborieuse et prend du temps (Kim *et al.* 2005; Desai *et al.* 2006; Desai *et al.* 2008; Xiong *et al.* 2008), mais ignore également les interactions combinées de chaque facteur (Kalil *et al.* 2000; Burkert *et al.* 2004; Gao *et al.* 2009). Sous la méthodologie des surfaces de réponse, un modèle polynomial quadratique, représentant l'effet d'interaction entre les variables, est généralement construit pour les processus / variables définies. Bien que la MSR présente de nombreux avantages et elle a été appliquée avec succès pour étudier et optimiser plusieurs processus de fermentation (Banik *et al.* 2007; Gao *et al.* 2009), il est difficile de dire qu'elle est généralisable à toutes les études de modélisation et d'optimisation (Baş et Boyacı 2007).

Ces deux problématiques nous ont amené à envisager une modélisation non conventionnelle en faisant appel aux réseaux de neurones artificiels. À l'heure actuelle, les réseaux de neurones (RN) constituent une alternative intéressante dans la modélisation des processus non linéaires et très complexes. La littérature dédiée fait largement état d'applications ou les RN offrent des solutions efficaces pour une large gamme de problèmes de fermentation, dont certains sont difficiles à traiter par les approches classiques (Tompos *et al.* 2007; Desai *et al.* 2008; Singh *et al.* 2008; Zhang *et al.* 2010; Dragoi *et al.* 2013; Nasr *et al.* 2013).

En plus, l'espace d'entrée du modèle quadratique de la MSR peut être facilement optimisé en utilisant des méthodes basées sur les gradients. Les modèles neuronaux étant exclusivement basés sur des données, ne peuvent pas être garantie comme étant lisse. Par conséquent, les méthodes classiques d'optimisation basées sur les gradients, qui nécessitent que la fonction objective soit continue, dérivable et surtout lisse, ne peuvent pas être utilisés efficacement pour optimiser l'espace d'entrée d'un modèle neuronal. Récemment, l'optimisation par essaim de particules (OEP) a évolué comme une branche alternative importante des techniques stochastiques pour explorer l'espace de recherche dans les problèmes d'optimisation.

En effet, l'optimisation par essaim de particules (OEP) est une méta-heuristique récente développée en 1995 par l'ingénieur électrique Russell Eberhart et le sociologue James Kennedy (Eberhart et Kennedy 1995). Cet algorithme a été appliqué avec succès dans l'apprentissage des réseaux de neurones (Shen *et al.* 2006; Zhang *et al.* 2007; Lazzús 2009) et dans la résolution de divers problèmes d'optimisation mono-objectif, multi-objectif, à variables discrètes ou à variables continues (Cockshott et Hartman 2001; Huang *et al.* 2007; Liu *et al.* 2009)**.** Comparé à d'autres algorithmes évolutionnaires comme les algorithmes génétiques, l'algorithme d'OEP présente des propriétés intéressantes telles que : mise en ouvre facile, efficacité de calcul, de mémoire et une

coopération constructive entre les individus, il est donc plus probable et plus rapide d'affluer dans le domaine des meilleures solutions et de trouver rapidement des résultats optimaux (Eberhart et Kennedy 1995; Cockshott et Hartman 2001).

Le premier objectif de ce travail est l'application des RN de type feedforward pour modéliser la production industrielle de pleuromutiline en mode fed-batch et ceux dans le but de prédire et d'évaluer les variables difficilement mesurables on line, à savoir la biomasse, la consommation du substrat et la formation du produit, à partir des variables facilement mesurables. Notre deuxième objectif porte sur la présentation d'un couplage entre un réseau de neurones de type feedforward et l'algorithme d'OEP. Ce couplage a été appliqué à l'optimisation des conditions de cultures de *Pleurotus mutilus* en mode batch et ceux dans le but d'obtenir une production maximale en pleuromutiline.

La présente thèse s'organise en quatre chapitres :

Le premier chapitre est dédié à la présentation de la théorie des réseaux de neurones. Après avoir exposé un aperçu historique sur les RN, nous détaillons les différents éléments de base de cet outil mathématique. Nous limitons notre présentation à des aspects qui sont utiles pour la suite de ce travail. Une synthèse sur les principaux travaux utilisant les RN en bioprocédés est présentée à la fin de ce chapitre.

Dans le deuxième chapitre, consacré à la méthodologie d'optimisation par essaim de particules (OEP), nous donnons dans un premier temps un aperçu sur les modèles historiques relatifs à l'OEP. Nous poursuivons en exposant l'algorithme de base de l'OEP (description mathématique, étapes de l'algorithme et les caractéristiques principales). Ensuite nous passons en revue le confinement des particules, la topologie du voisinage ainsi que les différentes variantes de l'algorithme. Nous terminons ce chapitre par la présentation des divers travaux relatant de l'application de l'algorithme en bioprocédés.

Nous présentons dans le troisième chapitre l'application des RN pour la modélisation de la fermentation fed-batch de pleuromutiline. Pour cela, nous abordons en première partie les matériels et méthodes utilisés : Nous examinons d'abord la base de données collectée, puis nous présentons la procédure de développement et d'optimisation de l'architecture des trois modèles neuronaux de type feedforward employés respectivement pour décrire la croissance microbienne, la consommation du substrat et la formation du produit. Par la suite, une comparaison des trois modèles neuronaux optimisés avec les modèles cinétiques est illustrée. Enfin, une analyse de

sensibilité par deux méthodes différentes est appliquée aux trois réseaux. Dans la deuxième partie de ce chapitre, les différents résultats prometteurs obtenus sont discutés : Dans un premier temps, nous présentons les architectures optimales des modèles neuronaux, leurs performances de prédiction ainsi que leurs temps d'exécution. Ensuite, après l'extraction de formules mathématiques à partir de ces modèles et la présentation des résultats de comparaison, nous étudions leurs performances d'interpolation. Les résultats de l'analyse de sensibilité sont présentés à la fin de cette deuxième partie.

Le dernier chapitre est consacré à la méthodologie hybride d'optimisation. La première partie de ce chapitre aborde les matériels et les différentes techniques utilisées lors du suivi des expériences de fermentation ; Dans un deuxième temps, nous présentons les différents modèles prédictifs adoptés, la mise en œuvre de l'algorithme d'OEP et les différents logiciels utilisés ; la partie qui suit est consacrée à la discussion des résultats significatifs obtenus. La dernière partie illustre une comparaison de fourchette entre la méthodologie hybride proposée et d'autres méthodologies d'optimisation à base des RN existant dans la littérature.

Les conclusions tirées de ce travail ainsi que les études qui peuvent être envisagées à l'avenir sont présentées à la fin de cette thèse.

Chapitre I

Les réseaux de neurones

I.1. Introduction

Les réseaux de neurones connaissent depuis quelques années un succès croissant dans divers domaines des sciences de l'ingénieur, celui du génie des procédés n'échappe pas à cette règle. Dans ce chapitre, nous nous intéressons à l'application des réseaux de neurones aux bioprocédés.

Dans un premier temps, nous rappelons les définitions et notations de bases relatives aux réseaux de neurones. Nous poursuivons en exposant les types et la méthodologie d'apprentissage.

Nous présentons aussi, d'une manière générale, les étapes de conception d'un réseau de neurones : le choix des entrées et des sorties, l'élaboration de la base de données et de la structure du réseau…etc.

Nous terminons ce chapitre par la présentation de diverses applications qui sont destinées à montrer que dans le domaine des bioprocédés, les réseaux de neurones sont susceptibles d'apporter des solutions efficaces et élégantes.

I.2. Historique

En 1943, Mac Culloch et Pitts ont proposé le premier modèle de neurone biologique (Davalo et Naïm 1989). Ce dernier, appelé aussi neurone à seuil, a été inspiré des récentes découvertes en biologie. Ce sont des neurones logiques (0 ou 1). Ces deux physiciens ont montré que les neurones formels peuvent réaliser des fonctions logiques.

En 1949, le psychologue Donald Hebb a introduit, dans son ouvrage "The Organization of Behavior", le terme connexionisme pour parler des modèles massivement parallèles et connectés. Il a proposé de nombreuses règles de mise à jour des poids dont la plus connue à cette époque est la "règle de Hebb" (Hebb 1949; Davalo et Naïm 1989).

En 1958, le psychologue Frank Rosenblatt a développé le modèle du perceptron (Rosenblatt 1958). Il s'agit d'un réseau de neurones, capable d'apprendre à différencier des formes simples et à calculer certaines fonctions logiques. Il est inspiré du système visuel. Il a réussi à l'appliquer dans la reconnaissance des formes. Au début des années 60, les travaux de Rosenblatt ont vécu un grand intérêt dans le milieu scientifique. Mais en 1969, deux scientifiques américains appelés Minsky et Papert (Minsky et Papert 1969; Davalo et Naïm 1989) ont publié un livre dans lequel ont démontré

les limites du perceptron proposé par Rosenblatt. En particulier, son incapacité réside à la résolution des problèmes non linéaires.

Les travaux se sont ralentis considérablement jusqu'aux années 80. En 1982, Hopfield a fini par démontrer l'intérêt des réseaux entièrement connectés (Hopfield 1982; Davalo et Naïm 1989). Parallèlement, Werbos a conçu un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron : la rétro-propagation (en anglais back-propagation). Cet algorithme, qui permet de propager l'erreur vers les couches cachées, sera popularisé en 1986 dans un livre "Parallel Distributed Processing" par Rumelhart et McClelland (Rumelhart et McClelland 1986).

En 1989 Moody et Darken (Moody et Darken 1989) exploitent quelques résultats de l'interpolation multi variables pour proposer le Réseau à Fonctions de base Radiales (RFR), connu sous l'appellation anglophone "Radial Basis Function network" (RBF). Ce type de réseau se distingue des autres types de réseaux de neurones par sa représentation locale (Talbi 2011).

Depuis ces travaux, les applications des réseaux de neurones n'ont cessé de croître. Les applications en sont multiples et variées.

I.3. Notions de base sur les réseaux de neurones

I.3.1. Du neurone biologique au modèle mathématique

I.3.1.1. Le modèle neurophysiologique

Le cerveau humain est composé de cellules distinctes appelées neurones formant un ensemble dense d'environ 10 à 100 milliards d'unités interconnectées. Tel qu'illustré sur la figure I.1, le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma (corps du neurone).

L'information est traitée alors par le corps cellulaire. Si le potentiel d'action dépasse un certain seuil, le corps cellulaire répond par un stimulus. Le signal transmis par le neurone est cheminé ensuite le long de l'axone (unique) pour être transmis aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angströms entre l'axone du neurone afférent et les dendrites du neurone efférent. La jonction entre deux neurones, comme schématisé sur la figure I.2, s'appelle synapse (Tzanakou 2000).

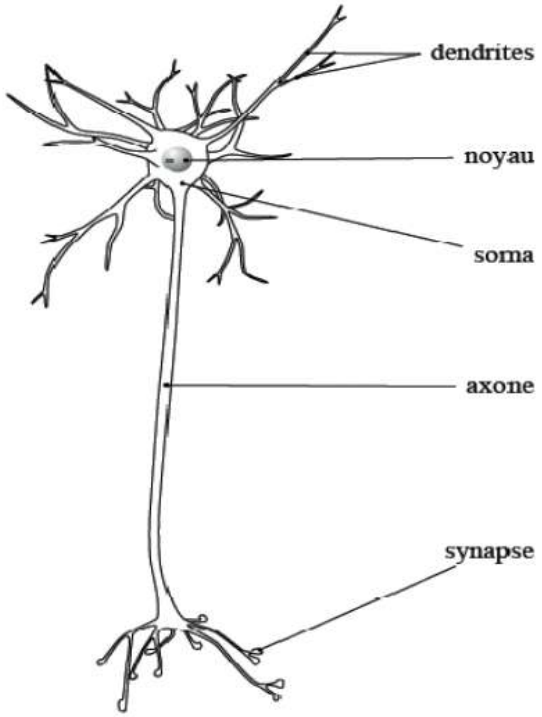


Figure I.1. Schéma d'un neurone (Talbi 2011).

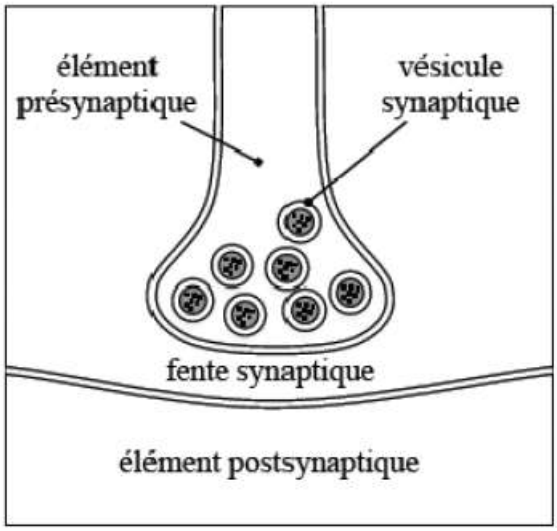


Figure I.2. Schéma d'une synapse (Talbi 2011).

I.3.1.2 Le modèle mathématique

Par analogie avec le neurone biologique, chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones en amont. À chacune de ces entrées est associé un poids *w*, abréviation du mot poids (weight en anglais), représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique qui se ramifie ensuite pour alimenter un nombre variable de neurones en aval tel qu'illustré sur la figure I.3 (Haykin 1999). À chaque connexion est associé un poids.

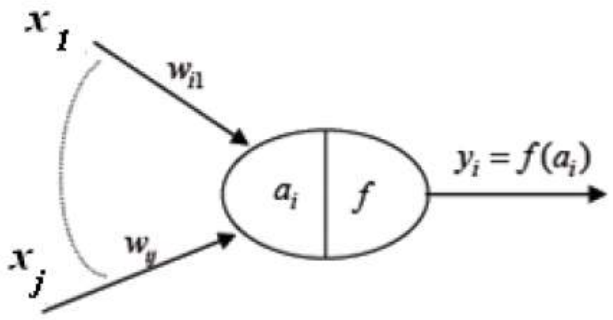


Figure I.3. Neurone formel (Talbi 2011).

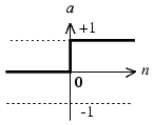
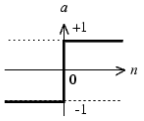
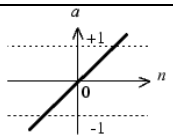
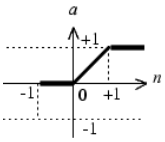
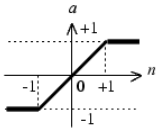
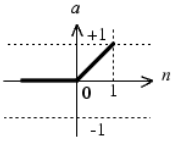
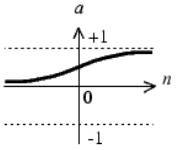
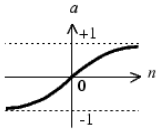
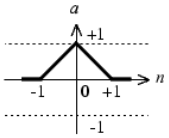
Le comportement du neurone artificiel se compose de deux phases (Haykin 1999) :

1. La première phase représente le prétraitement des données reçues en calculant la somme pondérée des entrées selon l'expression suivante :

$$a_i = \sum_j w_{ij} x_j \qquad \textbf{(I.1)}$$

2. Dans la deuxième phase, une fonction de transfert calcule la valeur de l'état interne du neurone. C'est cette valeur qui sera transmise aux neurones en aval. Plusieurs possibilités existent pour spécifier la nature de la fonction d'activation du modèle. Les fonctions mathématiques les plus souvent utilisées sont regroupées dans le tableau I.1. À la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle [0, +1] ou [-1, +1]. Ainsi, le neurone peut être défini mathématiquement comme étant une fonction algébrique, a priori non linéaire (suivant la fonction d'activation *f*) et bornée, avec des entrées *x* et paramétrée par les poids w_{ij}.

Tableau I.1. Fonctions de transfert usuelles, *a=f(n)* (Si-Moussa 2009).

Nom de la fonction	**Relation entrée / sortie**	**Graphe de la fonction**	**Fonction MATLAB® correspondante**
Seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlim
Seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlims
Linéaire	$a = n$		purelin
Linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$		satlin
Linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$		satlins
Linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$		poslin
Logistique (Sigmoïde logarithmique)	$a = \frac{1}{1+e^{-n}}$		logsig
Tangente hyperbolique	$a = \frac{e^{+n} - e^{-n}}{e^{+n} + e^{-n}}$		tansig
Triangulaire de base	$a = 1 - \lvert n \rvert,\ -1 \leq n \leq +1$ $a = 0,$ *autrement*		tribas

I.3.2. Architecture des réseaux de neurones

On distingue deux grandes familles de réseaux de neurones : les réseaux bouclés et les réseaux non bouclés (Figure I.4).

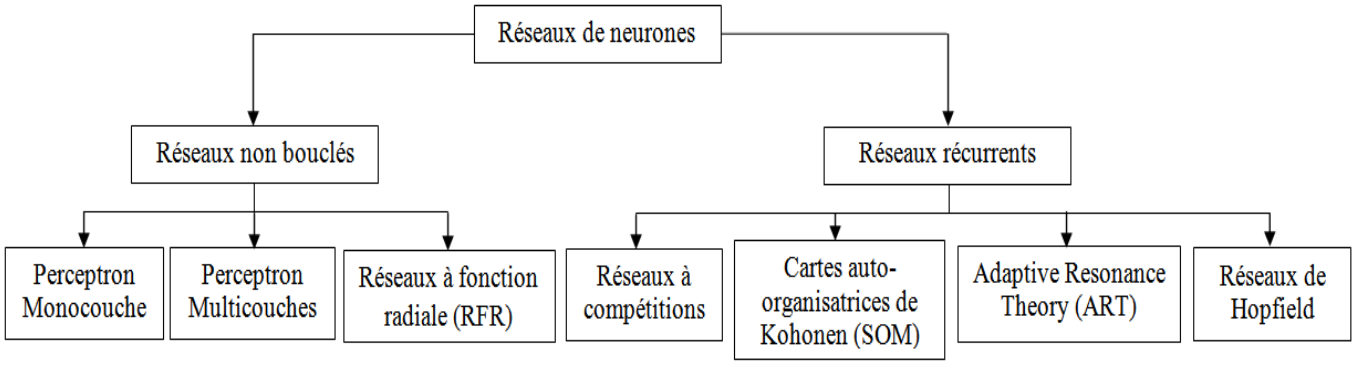


Figure I.4. Réseaux de neurones les plus utilisés (Talbi 2011).

I.3.2.1. Réseau de neurones non bouclé (en anglais feedforward)

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière" (Bishop 1995). On constate que le graphe d'un réseau non bouclé est acyclique. En effet, si on se déplace dans ce type de réseau à partir d'un neurone quelconque en suivant les connexions, on ne peut pas revenir au neurone de départ. Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie. Ceux qui effectuent des calculs intermédiaires sont les neurones cachés (Figure I.5).

Les réseaux non bouclés à couche sont structurés tel que les neurones qui appartiennent à une même couche ne soient pas connectés entre eux, chacune des couches recevant des signaux de la couche précédente et transmettant le résultat de ses traitements à la couche suivante. Les deux couches extrêmes correspondent à la couche d'entrée qui reçoit ses entrées du milieu extérieur d'une part et à la couche de sortie qui fournit le résultat des traitements effectués d'autre part. Les couches intermédiaires sont appelées couches cachées, leur nombre est variable (Chamekh 2007).

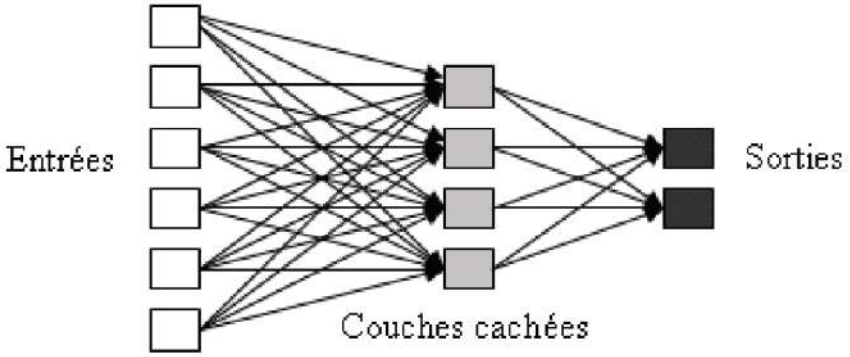


Figure I.5. Structure d'un réseau de neurones non bouclé (Chamekh 2007).

I.3.2.2. Réseau de neurones bouclé (ou récurrent)

Un réseau de neurones bouclé est schématisé par un graphe de connexions qui est cyclique. Lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de "cycle"). Ainsi, un retard entier multiple de l'unité de temps choisie est attaché à chaque connexion d'un réseau de neurones bouclé (ou à chaque arrête de son graphe). Une grandeur à un instant donné ne peut pas être fonction de sa propre valeur au même instant. Tout cycle du graphe du réseau doit avoir un retard non nul (Chamekh 2007).

La figure I.6 représente un exemple de réseau de neurones bouclé. Les chiffres dans les carrés indiquent le retard attaché à chaque connexion, exprimé en multiple de l'unité de temps. Ce réseau contient un cycle, qui part du neurone 3 et revient à celui-ci en passant par le neurone 4. La connexion de 4 vers 3 ayant un retard d'une unité de temps.

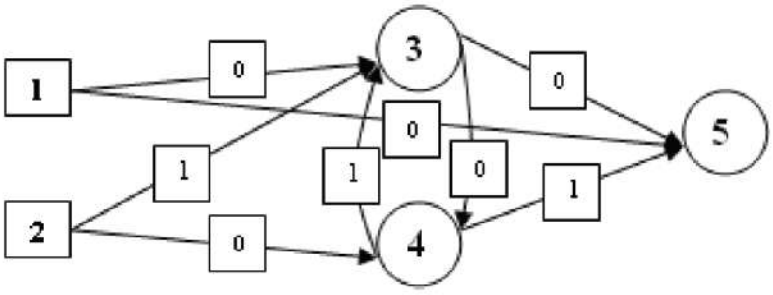


Figure I.6. Un réseau de neurones bouclé à deux entrées (Chamekh 2007).

I.3.3. Types d'apprentissage des réseaux de neurones

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux de neurones. L'apprentissage est une phase du développement du réseau de neurones durant laquelle on calcule les poids des neurones de telle manière que les sorties du réseau soient aussi proches que possible des sorties désirées. Les procédures d'apprentissage sont divisées, elles aussi, en deux grandes catégories : apprentissage supervisé et apprentissage non supervisé (Hassoun 1995).

I.3.3.1. Apprentissage supervisé

L'apprentissage supervisé implique l'existence d'un professeur qui a pour rôle d'évaluer le succès (ou l'échec) du réseau quand il lui est présenté un stimulus connu (on dit que ce stimulus est un exemple appartenant à la base d'apprentissage). Cette supervision consiste à renvoyer au réseau une information lui permettant de faire évoluer ses connexions (parfois aussi sa propre architecture) afin de faire diminuer son taux d'échec. L'information peut être explicite, sous la forme d'une mesure de l'erreur commise, par exemple, ou globalement sur l'ensemble des exemples de la base. C'est ce qui se passe dans le cas du perceptron. Mais elle peut être plus implicite (apprentissage par renforcement), sous forme d'une simple appréciation (bon ou mauvais, punition ou récompense), sans mesure d'erreur et même être globale, sur l'ensemble des tâches que le réseau doit exécuter. Dans ce dernier cas, il est facile de comprendre que l'apprentissage est le plus difficile, la difficulté majeure consistant pour le réseau à identifier les étapes du processus qui sont responsables de l'échec ou du succès (Hassoun 1995).

I.3.3.2. Apprentissage non supervisé

L'apprentissage non supervisé implique la fourniture à un réseau autonome d'une quantité suffisante d'exemples contenant des répétitions (autrement dit de la redondance), telles que celui-ci en dégage les régularités automatiquement. Ces réseaux sont souvent appelés auto-organisateurs, ou encore à apprentissage compétitif. Dans l'apprentissage non supervisé, les données ne contiennent pas d'informations sur une sortie désirée, il n'y a pas de superviseur ou expert humain. Il s'agit de déterminer les paramètres du réseau de neurones suivant un critère à définir. Dans ce cas, les exemples présentés à l'entrée provoquent une auto adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse pour des valeurs d'entrées similaires. Ce type d'apprentissage possède souvent moins de complexité dans le calcul par rapport à l'apprentissage supervisé. Bien entendu, l'architecture du réseau, préalablement définie par son utilisateur, est une forme de supervision (Hassoun 1995).

I.4. Les réseaux de neurones les plus utilisés

La Figure I.4 représente les réseaux de neurones les plus utilisés dans la littérature. Dans ce qui suit, nous allons faire une description de chacun d'eux.

I.4.1. Le perceptron monocouche (perceptron simple)

Le perceptron est le premier modèle des réseaux de neurones inventé en 1957 par Frank Rosenblatt (Rosenblatt 1958). C'est un réseau simple, puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie. Il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples, il suit généralement un apprentissage supervisé (Talbi 2011).

I.4.2. Le perceptron multicouche (en anglais Multi Layer Perceptron)

C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie (Keedwell et Narayanan 2005). Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Il peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués (Talbi 2011). Il suit un apprentissage supervisé, ce réseau a été choisi dans la suite de nos travaux.

I.4.3. Les réseaux à fonction radiale

Les réseaux à fonction de bases radiales (RFR) ou plus simplement réseaux à bases radiales ont été proposés par Moody et Darken (Moody et Darken 1989). Les RFR seront donc employés dans les mêmes types de problèmes que les perceptrons multicouches à savoir, en classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RFR est le mode hybride (Talbi 2011).

I.4.4. Les cartes auto-organisatrices de Kohonen (SOM)

Les cartes topologiques ou cartes auto organisatrices ont été introduites pour la première fois par Kohonen en 1982 (Kohonen 1982). Ce sont des réseaux à apprentissage non-supervisé qui établissent une carte discrète, ordonnée topo-logiquement, en fonction de patterns d'entrée. Le réseau forme ainsi une sorte de treillis dont chaque nœud est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation (Talbi 2011).

I.4.5. Les réseaux de Hopfield

Les réseaux de Hopfield (Hopfield 1982) sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différence entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non-linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances, mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non-supervisé (Talbi 2011).

I.4.6. Les réseaux ART

Les réseaux ART (en anglais Adaptive Resonance Theory) sont des réseaux à apprentissage par compétition (Carpenter et Grossberg 1987; Carpenter et Grossberg 1988). Dans ce genre de réseau, les vecteurs de poids ne seront adaptés que si l'entrée fournie est suffisamment proche d'un prototype déjà connu par le réseau. Il est à noter qu'il existe deux principaux types de réseaux ART : les ART-1 pour des entrées binaires et les ART-2 pour des entrées continues. Le mode d'apprentissage des ART peut être supervisé ou non (Talbi 2011).

I.5. Conception d'un modèle neuronal

Les réseaux de neurones réalisent des fonctions non linéaires paramétrées. Leurs mises en œuvre nécessitent :

- La détermination des entrées et des sorties pertinentes, c'est-à-dire les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser ;
- La construction de la base de données et le choix des échantillons ;
- La détermination du nombre de neurones cachés nécessaires pour obtenir une approximation satisfaisante ;
- La réalisation de l'apprentissage ;
- L'évaluation des performances du réseau de neurones à l'issue de l'apprentissage.

I.5.1. Détermination des entrées/sorties du réseau de neurones

Pour toute conception de modèle, la sélection des entrées doit prendre en compte deux points essentiels (Ammar 2007) :

- Premièrement, la dimension intrinsèque du vecteur des entrées doit être aussi petite que possible, en d'autres termes, la représentation des entrées doit être la plus compacte possible, tout en conservant pour l'essentiel la même quantité d'information et en gardant à l'esprit que les différentes entrées doivent être indépendantes.

- En second lieu, toutes les informations présentées dans les entrées doivent être pertinentes pour la grandeur que l'on cherche à modéliser : elles doivent donc avoir une influence réelle sur la valeur de la sortie.

I.5.2. Construction de la base de données et choix des échantillons

Le processus d'élaboration d'un réseau de neurones commence par la construction d'une base de données et le choix des échantillons. Il faut donc déterminer les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser.

Cette étape est cruciale et aide à déterminer le type de réseau le plus approprié pour résoudre un problème donné. En particulier, la façon dont se présente l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée et de sortie et la façon dont il faudra mener l'apprentissage, le test et la validation (Bishop 1995).

Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données, une pour effectuer l'apprentissage et l'autre pour tester le réseau obtenu et déterminer ses performances.

Pour mieux contrôler la phase d'apprentissage, il est souvent préférable de posséder une troisième base de données appelée "base de validation croisée" (Stone 1977). Les avantages liés à l'utilisation de cette troisième base de données seront exposés dans les paragraphes suivants. Notons qu'il n'y pas de règle pour déterminer ce partage d'une manière quantitative, néanmoins chaque base doit satisfaire aux contraintes de représentativité de chaque classe de données et doit généralement refléter la distribution réelle, c'est-à-dire la probabilité d'occurrence des diverses classes (Nascimento *et al.* 2000).

Généralement, les bases de données subissent un prétraitement qui consiste à effectuer une normalisation appropriée tenant compte de l'amplitude des valeurs acceptées par le réseau. Les valeurs d'entrées et de sortie sont normalisées dans un intervalle spécifique afin de donner à chaque paramètre la même influence statistique. De plus, un problème additionnel peut survenir à cause d'une possible saturation de la sortie, si la somme pondérée des entrées du perceptron multicouches ($\sum(w_i x_i)$) est dans la partie asymptotique de la fonction d'activation (Dames 2005).

I.5.3. Définition de la structure du réseau

La détermination de l'architecture optimale est restée pendant longtemps un problème ouvert. Il existe actuellement diverses méthodes empiriques mettant notamment en jeu des tests statistiques pour une vaste classe de réseaux (Urbani *et al.* 1994).

Comme nous l'avons mentionné, nous avons retenu le perceptron multicouche comme base du modèle. Nous structurons ce réseau en précisant le nombre de couches et de neurones cachés pour que le réseau soit en mesure de reproduire ce qui est déterministe dans les données.

I.5.3.1. Nombre de couches cachées

Mis à part les couches d'entrée et de sortie, il faut décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation. Néanmoins, il a été démontré qu'un perceptron multicouches avec une seule couche cachée pourvue d'un nombre suffisant de neurones, peut approximer n'importe quelle fonction avec la précision souhaitée (Hornik 1991).

I.5.3.2. Nombre de neurones cachés

Chaque neurone peut prendre en compte des profils spécifiques de neurones d'entrée. Un nombre plus important permet donc de mieux "coller" aux données présentées, mais diminue la capacité de généralisation du réseau. Il faut alors trouver le nombre adéquat de neurones cachés nécessaire pour obtenir une approximation satisfaisante. Le nombre de neurones cachés dépend du type de données dont on dispose. En général, ce nombre est déterminé par des règles empiriques. Suivant les auteurs, la taille de la couche cachée doit être (Dames 2005) :

- Soit égale à celle de la couche d'entrée ;
- Soit égale à 75 % de celle-ci ;

- Soit égale à la racine carrée du produit du nombre de neurones dans la couche d'entrée et de sortie.

Pour des cas plus sophistiqués, la recherche consisterait soit à :

- Estimer un réseau comportant de nombreux neurones puis à le simplifier par l'analyse des multi-colinéarités ou par une règle d'apprentissage éliminant les neurones inutiles (Dames 2005) ;

- Définir une architecture tenant compte de la structure des variables identifiées au préalable par une analyse en composantes principales (Bermejo et Cabestany 1999; Bermejo et Cabestany 2001) ;

- Augmenter progressivement la complexité du modèle à chaque itération, dite "Cascading", et réévaluer l'efficacité de généralisation à chaque addition d'un nouveau neurone (Dames 2005).

I.5.4. Apprentissage

Une fois l'architecture choisie, elle doit subir une phase d'apprentissage. Cette dernière consiste à calculer les coefficients synaptiques de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties "désirées". Celles-ci correspondent à la valeur de la fonction que l'on veut approcher ou à la sortie du processus que l'on veut modéliser (Dreyfus 2008).

L'apprentissage correspond donc à la phase du développement du réseau durant laquelle les coefficients synaptiques sont modifiés jusqu'à l'obtention du comportement désiré.

Dans notre étude, comme nous disposons des valeurs de sortie, nous avons choisi l'apprentissage supervisé, mettant en jeu un algorithme d'optimisation. Ce dernier cherche à minimiser, par des méthodes d'optimisation non linéaire, une fonction de coût qui constitue une mesure de l'écart entre les réponses réelles du réseau et ses réponses désirées (Dreyfus 2008).

Cette optimisation se fait de manière itérative, en modifiant les poids en fonction du gradient de la fonction de coût : le gradient est estimé par une méthode spécifique, dite la règle du gradient de l'erreur (en anglais delta rule). C'est une des règles les plus utilisées pour l'apprentissage supervisé de réseaux de neurones. Initialement développée pour résoudre des problèmes de traitements adaptatifs du signal, elle a conduit ensuite à un algorithme connu sous le nom de

l'algorithme de rétro-propagation du gradient de l'erreur (en anglais Back-propagation) (Dames 2005). On distingue deux étapes dans la phase d'apprentissage (Dames 2005) :

- Une phase de propagation, qui consiste à présenter une configuration d'entrée au réseau, puis à la propager à celle de sortie en passant par les couches cachées.

- Une phase de rétro-propagation, qui consiste, après le processus de propagation, à minimiser l'erreur commise sur l'ensemble des exemples présentés, erreur considérée comme une fonction des poids synaptiques. Cette erreur représente la somme des différences au carré entre les réponses calculées et celles désirées pour tous les exemples contenus dans l'ensemble de l'apprentissage.

Les algorithmes de rétro-propagation implantés actuellement dans les logiciels de RNA appartiennent à une des cinq classes suivantes (Si-Moussa 2009):

- Descente du gradient ;
- Descente des gradients conjugués ;
- Quasi-Newton ;
- Levenberg-Marquardt ;
- Régularisation

Le tableau I.2 résume les algorithmes d'apprentissage par rétro-propagation implanté dans la boite à outils des RN du programme MATLAB® (Demuth et Beale 2000).

Tableau I.2. Résumé des caractéristiques des algorithmes d'apprentissage de rétro-propagation implantés dans MATLAB® (Demuth et Beale 2000).

Algorithme	Caractéristiques
Descente du gradient	
Descente du gradient simple (***traingd)***	Descente du gradient de base. Réponse lente; peut être utilisée pour apprentissage en mode séquentiel
Descente du gradient avec moment (***traingdm)***	Descente du gradient avec moment. Le moment permet à un réseau de répondre non seulement au gradient local, mais aussi aux tendances récentes dans la surface d'erreur. Sans moment un réseau peut être coincé dans un minimum local peu profond. Plus rapide que ***traingd***, ne peut être utilisée que pour apprentissage en mode séquentiel
Descente du gradient avec taux d'apprentissage variable (***traingda)***	Taux d'apprentissage variable. Plus rapide que ***traingd***, mais ne peut être utilisée que pour apprentissage en mode batch. Utilise des règles heuristiques pour la variation du taux d'apprentissage
Descente du gradient avec moment et taux d'apprentissage variable (***traingdx)***	Descente du gradient avec inertie (momentum) et taux d'apprentissage variable
Descente des gradients conjugués	
Descente des gradients conjugués avec la méthode de mise à jour de Fletcher-Reeves ***(traincgf)***	Il possède la plus petite mémoire de stockage parmi les algorithmes de gradient conjugués.
Descente des gradients conjugués avec la méthode de mise à jour de Polak-Ribiére ***(traincgp)***	Nécessite une mémoire de stockage légèrement plus grande que ***traincgf***. Convergence plus rapide sur quelques problèmes.
Descente des gradients conjugués avec la méthode de reprises Powell-Beale ***(traincgb)***	Nécessite une mémoire de stockage légèrement plus grande que ***traincgp***. Convergence Généralement plus rapide.
Scaled conjugate gradient ***(trainscg)***	Le seul algorithme de gradients conjugués qui n'exige aucune recherche de ligne. Très bon algorithme d'apprentissage en général.
Quasi-Newton	
BFGS quasi-Newton ***(trainbfg)***	Il exige le stockage d'une approximation de la matrice Hessiènne et nécessite plus de calcul à chaque itération que les algorithmes de gradients conjugués, mais possède habituellement une convergence plus rapide.

Sécante à un pas ***(trainoss)***	Cet algorithme exige moins de stockage et calcul par époque que l'algorithme BFGS. On peut considérer cela un compromis entre des algorithmes de quasi-Newton complets et des algorithmes de gradients conjugués.
Levenberg-Marquardt	
Levenberg-Marquardt ***(trainlm)***	C'est l'algorithme d'apprentissage le plus rapide pour les réseaux de taille modérée. Il a la caractéristique de réduction de mémoire pour l'utilisation quand l'échantillon d'apprentissage est grand.
Régularisation	
Régularisation Bayésienne ***(trainbr)***	Mise à jour des poids et des biais selon l'optimisation de Levenberg-Marquardt. Il réduit au minimum une combinaison d'erreurs quadratiques et des poids et détermine ensuite la combinaison correcte afin de produire un réseau avec une capacité de généralisation meilleure. Ce processus est appelé la régularisation bayésienne. Il réduit la difficulté de déterminer l'architecture de réseau optimum.

I.5.5. Evaluation des performances du réseau de neurones

Enfin, la dernière étape doit permettre d'estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage. Elle permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui posent problème. Si les performances ne sont pas satisfaisantes, il faudra, soit modifié l'architecture du réseau, soit modifié la base d'apprentissage (Dames 2005).

I.6. Application des réseaux de neurones en bioprocédés

Durant les dernières années, la littérature scientifique relative aux réseaux de neurones (RN) s'est enrichie considérablement par la publication des travaux relatant des applications diverses telles que la modélisation, la reconnaissance des formes, l'optimisation et le contrôle des bioprocédés, tous sujet d'intérêt en biotechnologie. Les ouvrages de Baughman et Liu (Baughman et Liu 1995), Montague et Morris (Montague et Morris 1994), Basheer et Hajmeer (Basheer et Hajmeer 2000) et Nicoletti (Nicoletti 2009) donnent des aperçus sur les applications des RN en bioprocédés. Plusieurs applications sont résumées ci-dessous :

I.6.1. Modélisation et prédiction

Plusieurs exemples, sur la modélisation et la prédiction par les RN, sont cités dans la littérature. À titre d'exemple, l'estimation en ligne des concentrations de la biomasse, du glucose et de l'éthanol par un réseau récurrent ayant pour entrées : les mesures antérieures, les conditions du processus et les entrées de commande du processus. Zhu et ces collaborateurs (Zhu *et al.* 1996) ont utilisé un RN pour décrire la production à grande échelle de lysine et ont prédit avec succès la formation du produit à partir des mesures en ligne et la simulation récursive de la consommation du sucre. Galván et ces co-auteurs (Galván *et al.* 1996) ont utilisé les réseaux de neurones dans l'ajustement de données cinétiques complexes des réactions chimiques, ils ont démontré que cette modélisation s'avère très utile lorsque les mécanismes cinétiques ne sont pas bien saisis et que suffisamment de données sur le processus sont disponibles. Les RN peuvent remplacer les modèles (semi-) mécanistiques, parce que, une fois ajusté, ils sont mathématiquement robuste, facile à utiliser et rapide (Fontaine et Germain 2001).

Dans le travail de Shene et co-auteurs (Shene *et al.* 1999)**,** deux réseaux de neurones différents (un réseau neuronal de type feedforward et un réseau neuronal hybride) ont été conçus pour prédire les variables d'état dans la fermentation batch de l'éthanol en tenant compte de l'effet de la composition du milieu et de la température. Becker et ces collaborateurs (Becker *et al.* 2002) ont appliqué un RN dynamique pour la modélisation de la fermentation de la bière, suivie d'une optimisation des profils de température afin de réduire le temps de fermentation. Saraceno et co-autheurs (Saraceno *et al.* 2009) ont simulé la fermentation de *lactosérum de fromage ricotta* pour la production d'éthanol au moyen d'un modèle neuronal hybride multiple, obtenu par couplage d'un réseau neuronal de type perceptron multicouches avec les équations de bilan de masse pour le lactose, l'éthanol et la biomasse. Nasr et ces collaborateurs (Nasr *et al.* 2013) ont développés un modèle neuronal de type rétro-propagation pour estimer le profil de production d'hydrogène en fonction du temps dans les études batch, les entrées du RN sont : le *pH* initial, les concentrations initiales du substrat et de la biomasse, la température et le temps. Pour ce qui est de notre application (Khaouane *et al.* 2013), modélisation de la fermentation fed-batch de la pleuromutiline, elle sera détaillée dans le 3ème chapitre.

I.6.2. Capteurs logiciels (en anglais soft sensors)

Les RN sont aussi appliqués pour le développement de capteurs logiciels à des fins d'estimation et de contrôle. Généralement les mesures on line sont reliées aux mesures off line des états du processus (Di Massimo *et al.* 1992; Karim *et al.* 1997; Ye *et al.* 1998; Linko *et al.* 1999).

Par exemple, les mesures on line tel que le *pH*, l'ammoniac et la vitesse d'agitation ont été appliqués pour prédire les niveaux de biomasse et de produit dans le travail de Jin et co-auteurs (Jin *et al.* 1996) et Linko et co-auteurs (Linko *et al.* 1997). Yuan et Vanrolleghem (Yuan et Vanrolleghem 1999) ont appliqué les RN pour la prédiction on line des profils de concentration de pénicilline.

En outre, les informations spécifiques du processus comme la phase du processus et le temps de fermentation peuvent être utilisées comme des inputs additionnels (Di Massimo *et al.* 1992). Les capteurs logiciels peuvent aussi être appliqués pour remplacer les procédures analytiques couteuses et difficiles. Kang et ces collaborateurs (Kang *et al.* 1998) ont développé un capteur logiciel pour la mesure de la concentration de l'acide clavulanique. Une autre nouveauté est le nez électronique (Bachinger et Mandenius 2000).

I.6.3. Reconnaissance des formes (en anglais pattern recognition)

Les RN sont appliqués pour la détection des irrégularités du processus, pour réduire la variabilité du processus, pour la prédiction on line de la performance du processus et pour détecter les défauts des capteurs (Glassey *et al.* 1997). Un autre exemple est l'application d'un RN à auto-organisation (Kohonen) pour établir un rapport entre les données de la qualité des semences pour prédire la performance de la phase finale de la fermentation (Ignova *et al.* 1999) ; Par ailleurs, la phase finale atypique de la fermentation peut être détectée dès le début. L'application des RN auto-associative pour la détection d'anomalie on line a été proposée par Shimizu et co-auteurs (Shimizu *et al.* 1998) et aussi appliquée dans la production de virginiamycin par Huang et ces collaborateurs (Huang *et al.* 2002).

I.6.4. Optimisation

Les RN ont été appliqués pour l'optimisation à base de modèle des milieux de culture en mode batch par kennedy et co-auteurs (Kennedy *et al.* 1992; Kennedy et Spooner 1996), qui a abouti à une grande réduction dans les expériences par comparaison à l'approche classique basée sur les plans d'expérience.

La combinaison des RN avec les algorithmes évolutionnaires, tel que les algorithmes génétiques, a donné naissance à une nouvelle approche qui a prouvé et ne cesse de prouver son efficacité et sa robustesse dans l'optimisation des milieux de culture de *Lactobacillus plantarum* (Desai *et al.* 2006), optimisation des constituants du milieu pour la production de lipase (Haider *et*

al. 2008), optimisation de la production de scléroglucane (Desai *et al.* 2008), optimisation de la production de L-asparagine (Gurunathan et Sahadevan 2011) et optimisation de la production de lipase (Nelofer *et al.* 2012).

I.6.5. Contrôle

Les RN sont souvent appliqués au sein des schémas de contrôle en tant que modèles, estimateurs d'état ou optimiseur. Schenker et Agarwal (Schenker et Agarwal 1998) ont étudié différent type de structures neuronales pour le contrôle prédictif des réacteurs chimiques. Kovarova et ces collaborateurs (Kovarova-Kovar *et al.* 2000) ont utilisé un modèle neuronal en combinaison avec une optimisation on line pour l'adaptation de la trajectoire d'alimentation durant la production de riboflavine.

I.7. Conclusion

Ce chapitre introduit les notions de base sur les réseaux de neurones artificiels (RN). Après un historique sur l'apparition des RN et la présentation du modèle biologique et mathématique du neurone, les différentes architectures des RN ainsi que leurs types d'apprentissage sont présentés. Dans un deuxième temps, nous avons exposé les RN les plus utilisés ainsi que les différentes étapes de conception d'un réseau de neurones. Nous avons terminé ce chapitre par la présentation de différent travaux relatant des applications diverses des RN en bioprocédés notamment dans la modélisation, la prédiction, la reconnaissance des formes, l'optimisation et le contrôle.

Chapitre II

Optimisation par essaim de particules

II.1. Introduction

L'optimisation des problèmes dits "difficiles", qui surgissent dans des secteurs très divers notamment en biotechnologie, a conduit les chercheurs à proposer des méthodes de résolutions performantes et conceptuellement différentes des techniques classiques de programmation mathématique. Parmi ces méthodes modernes d'optimisation, on trouve les méta-heuristiques.

Les méta-heuristiques sont souvent inspirées par les systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou l'optimisation par essaim de particules). L'avantage des méta-heuristiques est de trouver un minimum global à un problème de minimisation et de ne pas rester bloqué sur un minimum local (Dorigo et Caro 1999).

Ce chapitre est consacré uniquement à l'optimisation par essaim de particules OEP (en anglais Particle Swarm Optimization PSO), l'algorithme d'OEP est l'un des algorithmes heuristiques modernes, qui peut être utilisé pour optimiser des problèmes non-linéaires, non-continus et à plusieurs variables (Eberhart et Kennedy 1995). Il est inspiré des essaims d'insectes (abeilles, guêpes,…), des écoles de poissons, ou des troupeaux d'oiseaux et de leurs mouvements coordonnés. En effet, comme ces animaux se déplacent en groupe pour trouver la source de nourriture ou éviter les prédateurs, l'algorithme à essaim de particules recherche des solutions à un problème d'optimisation.

Comparé à d'autres algorithmes évolutionnaires comme les algorithmes génétiques, l'algorithme d'OEP présente des propriétés intéressantes telles que : mise en œuvre facile, efficacité de calcul, de mémoire et une coopération constructive entre les individus, il est donc plus probable et plus rapide d'affluer dans le domaine des meilleures solutions et de trouver rapidement des résultats optimaux (Eberhart et Kennedy 1995; Cockshott et Hartman 2001).

Dans un premier temps, nous donnons un aperçu sur les modèles historiques relatifs à l'OEP. Nous poursuivons en exposant l'algorithme de base de l'OEP (description mathématique, étapes de l'algorithme et les caractéristiques principales). Nous présentons aussi, d'une manière générale, le confinement des particules, la topologie du voisinage ainsi que les différentes variantes de l'algorithme. Nous terminons ce chapitre par la présentation des diverses applications réussites de l'algorithme en biotechnologie.

II.2. Développement historique

L'optimisation à essaim de particules est une méta-heuristique développée en 1995 par Eberhart (ingénieur en électricité) et Kennedy (socio-psychologue) (Eberhart et Kennedy 1995). Cet algorithme s'inspire de l'intelligence en essaim et s'appuie notamment sur un modèle développé par le biologiste Reynolds à la fin des années 80, permettant de simuler le déplacement organisé d'un groupe d'oiseaux (Reynolds 1987).

II.2.1. Modèle de Reynolds

En 1987, Reynolds (Reynolds 1987) créa un modèle appelé "boid", afin de simuler les formations des oiseaux pour mieux comprendre et expliquer ces phénomènes et également la coordination des mouvements (parfois très brusques) de ces populations. Un boid est une entité sensée modélisée un individu afin d'étudier les propriétés d'assemblage de plusieurs de ces individus. Chaque boid est implémenté individuellement et navigue selon sa propre perception de l'environnement dynamique. Un boid obéit aux trois règles simples suivantes (Khedam 2008) :

1. Règle de séparation qui stipule qu'un boid doit s'éloigner des autres boids trop proches de lui ;
2. Règle de mimétisme ou d'alignement qui stipule qu'un boid doit se déplacer dans la même direction que l'ensemble en moyennant les vitesses et les directions des autres boids ;
3. Règle de cohésion qui stipule qu'un boid doit éviter de se trouver sur l'extérieur de la formation et ainsi se déplacer vers le centre perçu de la formation.

Ces trois règles de base, permettent l'attraction et la répulsion de chacun des individus et permettent la stabilité de l'ensemble (Khedam 2008).

II.2.2. Modèle de Kennedy et Eberhart

Kennedy et Eberhart (Eberhart et Kennedy 1995) ont cherché à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale. Le modèle qu'ils ont proposé a ensuite été étendu en un algorithme simple et efficace d'optimisation. Les particules sont les individus et elles se déplacent dans l'hyperespace de recherche. Le processus de recherche est basé sur les deux règles suivantes (Khedam 2008) :

1. Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.

2. Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

Dans ce modèle, l'intelligence est donc collective et non individuelle. L'environnement et les interactions y jouent un rôle fondamental. Ainsi, dans un essaim volant ou nageant, chaque individu réagit aux différentes règles établies par Reynolds, Kennedy et Eberhart. Ces règles sont identiques à toute la population et ne prennent en compte que le voisinage immédiat de l'individu qui ne peut pas percevoir l'ensemble du nuage. À partir du déplacement local de l'individu, des formes complexes pour le nuage d'animaux vont pouvoir apparaître, formes qui dépassent le cadre de la règle locale. Cette propriété d'émergence est recherchée dans de nombreux problèmes d'informatique, puisqu'elle va notamment permettre à la fois un parallélisme décentralisé et massive de l'algorithme, avec des comportements élémentaires simples tout en obtenant globalement un résultat complexe (Khedam 2008).

II.3. L'algorithme de base

II.3.1. Description mathématique

Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, "survole" l'espace de recherche, en quête de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes (Cooren 2008) :

i. Une composante physique : la particule tend à suivre sa direction courante de déplacement ;

ii. Une composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;

iii. Une composante sociale : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins.

Dans le cas d'un problème d'optimisation, la qualité d'un site de l'espace de recherche est déterminée par la valeur de la fonction objet en ce point. La figure II.1 illustre la stratégie de déplacement d'une particule.

Dans la version originale de l'OEP, l'essaim se compose de N particules en mouvement autour d'un espace de recherche D-dimensionnelle de la fonction objet (en anglais fitness or objective function). Chaque particule évalue la fonction objet à son emplacement actuel, puis

détermine son mouvement à travers l'espace de recherche en combinant certains aspects de l'histoire de ces emplacements actuels et meilleurs avec ceux d'un ou de plusieurs membre de l'essaim, avec quelques perturbations aléatoires.

La valeur $Pbest_i$ est présentée comme le meilleur emplacement déjà visité de la $i^{ème}$ particule, elle est notée $P_i = (P_{i1}, P_{i2}, ..., P_{id})$. La valeur *gbest* est le meilleur emplacement globale dans les emplacements visités de toutes les particules jusqu'à présent, elle est notée $g = (g_1, g_2, ..., g_d)$. La position instantanée de la $i^{ème}$ particule est présentée par $x_i = (x_{i1}, x_{i2}, ..., x_{id})$, $x \in [x_{min}, x_{max}]$ et sa vitesse est présentée par $v_i = (v_{i1}, v_{i2}, ..., v_{id})$, $v \in [v_{min}, v_{max}]$.

v_{max} est une vitesse maximale seuil qui est une constante prédéterminée selon la fonction d'optimisation objectif. Si une vitesse dans une dimension dépasse le maximum, elle sera réglée à v_{max}. ce paramètre contrôle la vitesse de convergence de l'algorithme d'OEP et peut l'empêcher de se développer trop vite (Panigrahi *et al*. 2011).

À chaque itération, la position et la vitesse de la $i^{ème}$ particule sont mis à jour avec $Pbest_i$ et *gbest* de l'essaim de population. Les équations de mise à jour peuvent être formulées ainsi :

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (Pbest_{id}(t) - x_{id}(t)) + c_2 r_2 (gbest_d(t) - x_{id}(t)) \quad \textbf{(II.1)}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad \textbf{(II.2)}$$

Où r_1 et r_2 sont deux nombres aléatoires avec une répartition uniforme entre 0 et 1 ; c_1 et c_2 sont deux constantes, appelées coefficients d'accélération ; w est en général une constante répartie uniformément dans l'intervalle [0, 1], appelée facteur d'inertie ou coefficient d'inertie ou encore poids d'inertie (Panigrahi et al. 2011).

- $wv_{id}(t)$ correspond à la composante *physique* du déplacement. Le paramètre w contrôle l'influence de la direction de déplacement sur le déplacement futur. Il est à noter que, dans certaines applications, le paramètre w peut être variable (Cooren 2008).

- $c_1 r_1 (Pbest_{id}(t) - x_{id}(t))$ correspond à la composante cognitive du déplacement. c_1 contrôle le comportement cognitif de la particule.

- $c_2 r_2 (gbest_d(t) - x_{id}(t))$ correspond à la composante sociale du déplacement. c_2 contrôle l'aptitude sociale de la particule.

La combinaison des paramètres w, c_1 *et* c_2 permet d'établir un équilibre entre les phases diversification et intensification du processus de recherche (Porto *et al.* 1998; Eberhart *et al.* 2001).

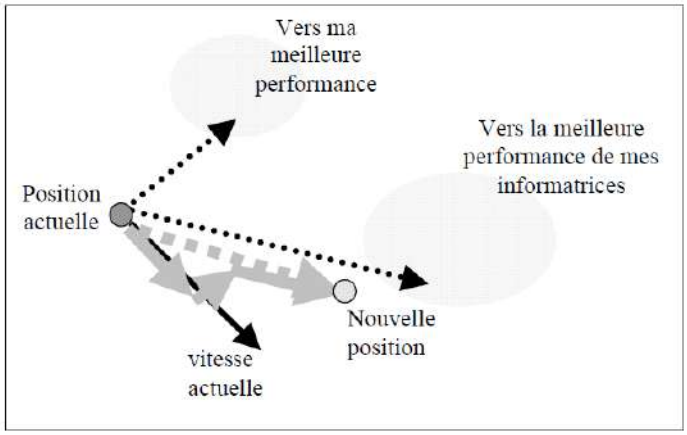


Figure II.1. Schéma du principe de déplacement d'une particule (Slimani 2009).

II.3.2. Les étapes de l'algorithme

L'algorithme de cette méthode peut être décrit comme suit (Slimani 2009) :

- **1ère étape** : Initialisation des coefficients c_1, c_2 et w.
- **2ème étape** : Création de la population initiale aléatoirement et calcul de la fitness de chaque particule ($Pbest_i$) : la meilleure position de la particule i dans la population actuelle ; (*gbest*): la meilleure position dans toutes les populations (la meilleure des meilleures).
- **3ème étape** : Calcul de la nouvelle vitesse et nouvelle position de chaque particule par l'utilisation des formules (II.1) et (II.2).
- **4ème étape** : Calcul de la meilleure fitness de la population initiale et comparer à la précédente pour trouver la meilleure de toutes les populations (*gbest*).
- **5ème étape** : Incrémentassions du nombre d'itération t = t+1.
- **6ème étape** : Passer à la 7ème étape si un critère d'arrêt est satisfait. Autrement, aller à la 3ème étape.
- **7ème étape** : La position enregistrée dans (*gbest*) est la solution optimale.

La figure II.2 présente l'organigramme de la méthode d'OEP.

Le critère d'arrêt peut être différent suivant le problème posé. Si l'optimum global est connu a priori, on peut définir une "erreur acceptable" comme critère d'arrêt. Sinon, il est commun de fixer un nombre maximum d'évaluations de la fonction objet ou un nombre maximum d'itérations comme

critère d'arrêt. Cependant, au regard du problème posé et des exigences de l'utilisateur, d'autres critères d'arrêt peuvent être utilisés (Cooren 2008).

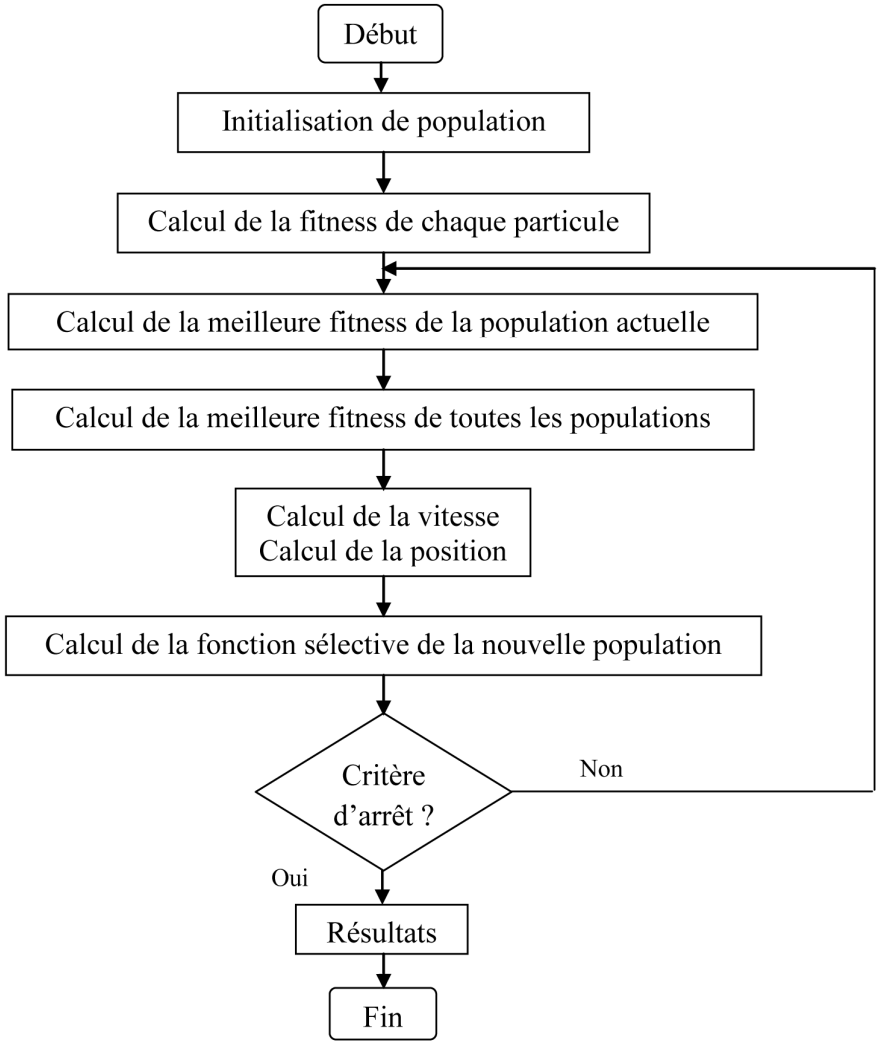


Figure II.2. Organigramme de la méthode d'OEP (Slimani 2009).

II.3.3. Caractéristiques principales

Ce modèle présente quelques propriétés intéressantes, qui en font un bon outil pour de nombreux problèmes d'optimisation, particulièrement les problèmes fortement non linéaires, continus ou mixtes (certaines variables étant réelles et d'autres entières) (Slimani 2009) :

- il est facile à programmer, quelques lignes de code suffisent dans n'importe quel langage évolué.
- il est robuste (de mauvais choix de paramètres dégradent les performances, mais n'empêchent pas d'obtenir une solution).

Pour réaliser son prochain mouvement, chaque particule combine trois tendances : suivre sa vitesse propre, revenir vers sa meilleure performance, aller vers la meilleure performance de ses informatrices.

II.4. Confinement des particules

Il est possible que le déplacement d'une particule la conduise à sortir de l'espace de recherche. Dans ce cas, on peut assister à une amplification des rétroactions positives, qui conduit à une divergence de système. Pour s'affranchir de ce problème, on peut introduire un nouveau paramètre v_{max}, qui va permettre de contrôler l'explosion du système (Eberhart *et al.* 1996). Une étude sur le comportement de l'OEP suivant les valeurs de v_{max} est disponible dans (Fan et Shi 2001).

De plus, une stratégie de confinement des particules peut être introduite. Une telle stratégie permet de ramener une particule sortie de l'espace de recherche à l'intérieur de celui-ci. Plusieurs méthodes peuvent être alors employées (Cooren 2008) :

- La particule est laissée à l'extérieur de l'espace de recherche, mais on n'évalue pas sa fonction objet. Ainsi, elle ne pourra pas attirer les autres particules en dehors de l'espace de recherche ;
- La particule est stoppée à la frontière et les composantes correspondantes de la vitesse sont annulées ;
- La particule rebondit sur la frontière. La particule est stoppée sur la frontière, mais les composantes correspondantes de la vitesse sont multipliées par un coefficient à tirer aléatoirement dans l'intervalle [-1, 0].

Cependant, suivant les besoins, de nombreuses autres méthodes peuvent être utilisées.

II.5. Topologie du voisinage

Le voisinage constitue la structure du réseau social. Les particules à l'intérieur d'un voisinage communiquent entre elles. Différents voisinages ont été étudiés (Kennedy 1999) et sont considérés en fonction des identificateurs des particules et non des informations topologiques comme les distances euclidiennes dans l'espace de recherche (Kennedy 1999; Wang *et al.* 2005) :

- Topologie en étoile (Figure II.3 (a)) : le réseau social est complet, chaque particule est attirée vers la meilleure particule notée *gbest* et communique avec les autres.
- Topologie en anneau (Figure II.3 (b)) : chaque particule communique avec *n* voisines immédiates (*n* = 3). Chaque particule tend à se déplacer vers la meilleure dans son voisinage local noté *Pbest.*
- Topologie en rayon (Figure II.3 (c)) : une particule "centrale" est connectée à toutes les autres, seule cette particule centrale ajuste sa position vers la meilleure. Si cela provoque une amélioration, l'information est propagée aux autres.

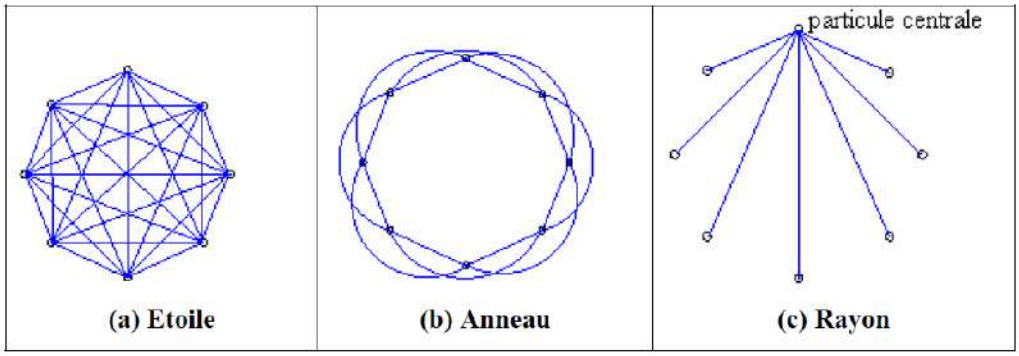


Figure II.3. Trois topologies du voisinage (Slimani 2009).

II.6. Facteur d'inertie et variantes de l'algorithme d'OEP

Le facteur d'inertie joue un rôle important dans le comportement de convergence de l'algorithme d'OEP. Le facteur d'inertie est utilisé pour contrôler l'impact de l'historique des vitesses précédentes sur la vitesse instantanée (Pintér 2006). De cette façon, le paramètre *w* régule le compromis entre les aptitudes d'exploration (recherche globale) et d'exploitation (recherche locale). Une grande valeur du poids d'inertie facilite la recherche globale (recherche de nouveaux domaines), tandis qu'une valeur faible tend à faciliter la recherche locale (affiner la zone de recherche en cours) (Pintér 2006).

Une valeur appropriée du facteur d'inertie fournit habituellement un équilibre entre les capacités de recherche globales et locales et par conséquent une réduction du nombre d'itérations nécessaire pour trouver la solution optimale (Pintér 2006).

Une règle générale suggère qu'il est préférable initialement de définir une grande valeur du coefficient d'inertie, afin de favoriser l'exploration dans l'espace de recherche et progressivement la diminuer afin d'augmenter la performance de l'algorithme d'OEP (Pintér 2006).

Plusieurs chercheurs ont essayé d'établir cet équilibre entre l'exploitation et l'exploration, ce qui a généré et ne cesse de générer des variantes de l'algorithme d'OEP. On se limite dans ce qui suit à quatre variantes.

II.6.1. Trelea de types 1 et 2 (réglage fixe)

Dans l'article de Trelea (Trelea 2003) et après un grand nombre d'expériences de simulation**,** il a été démontré qu'une bonne convergence peut être assurée par deux séries de paramètres (série 1 : $w = 0.6$, $c_1 = c_2 = 1.7$; série 2 : $w = 0.729$, $c_1 = c_2 = 1.494$).

$$v_{id}(t+1) = 0.6v_{id}(t) + 1.7r_1(Pbest_{id}(t) - x_{id}(t)) + 1.7r_2(gbest_d(t) - x_{id}(t)) \quad \textbf{(II.3)}$$

$$v_{id}(t+1) = 0.729v_{id}(t) + 1.494r_1(Pbest_{id}(t) - x_{id}(t)) + 1.494r_2(gbest_d(t) - x_{id}(t)) \quad \textbf{(II.4)}$$

II.6.2. Approche du poids d'inertie

Cette approche est basée sur la diminution du poids d'inertie selon l'équation suivante (Eberhart *et al.* 2001) :

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})}{Iter_{\max}} Iter \quad \textbf{(II.5)}$$

Avec : w_{max}, w_{min} : poids d'inertie initiale et finale ;

$Iter_{max}$: nombre d'itération maximal ;

Iter : nombre d'itération actuel.

II.6.3. Approche du facteur de constriction

Clerc et Kennedy (Clerc et Kennedy 2002) ont développé un facteur de constriction ayant pour but d'assurer la convergence de l'algorithme d'OEP. Ce facteur χ est introduit dans la formule de calcul de vitesse et est définie comme suit :

$$v_{id}(t+1) = \chi \otimes [v_{id}(t) + c_1 r_1(Pbest_{id}(t) - x_{id}(t)) + c_2 r_2(gbest_d(t) - x_{id}(t))] \quad \textbf{(II.6)}$$

$$\chi = \begin{cases} \dfrac{2.k}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|} & Si\ \phi \succ 4 \\ k & Si\ 0 \le \phi \le 4 \end{cases} \quad \textbf{(II.7)}$$

Avec : $\phi = c_1+c_2$;

k est un nombre entre 0 et 1, qui est généralement pris égale à 1.

II.7. Application de l'algorithme d'OEP en bioprocédés

II.7.1. Estimation de paramètres

L'estimation des paramètres d'un modèle (nommé aussi calibration de modèle) constitue une phase importante dans le processus de construction d'un modèle mathématique (Figure II.4). La qualité des paramètres estimés et par conséquent la méthode d'estimation utilisée à cette fin ont une grande influence sur les performances du modèle (efficacité de prédiction et utilité futur). L'algorithme d'OEP et vue ces qualités citées précédemment s'est ajouté ces dernières années aux méthodes d'estimation existantes dans la littérature comme un estimateur précis. En effet, l'algorithme d'OEP a démontré une très grande efficacité dans la calibration des modèles mathématiques décrivant :

- les procédés d'adsorption pour la séparation de l'enzyme dans une colonne à lit expansé (Moraes *et al.* 2009) ;
- la bio-sorption des colorants synthétiques à partir de l'eau (Fiorentin *et al.* 2010) ;
- séparation des sucres à partir du mélange réactionnel pour la production de fructo-oligosaccharides en utilisant la zéolite X comme adsorbant dans un réacteur à cuve agitée (Kuhn 2010) ;
- le processus batch de biodégradation des composés toxiques BTEX (**B**enzène, **T**oluène, **É**thyle benzène et les isomères de **X**ylène) (Trigueros *et al.* 2010) ;
- l'adsorption de céphalosporine C dans une colonne à lit fixe (Burkert *et al.* 2011).

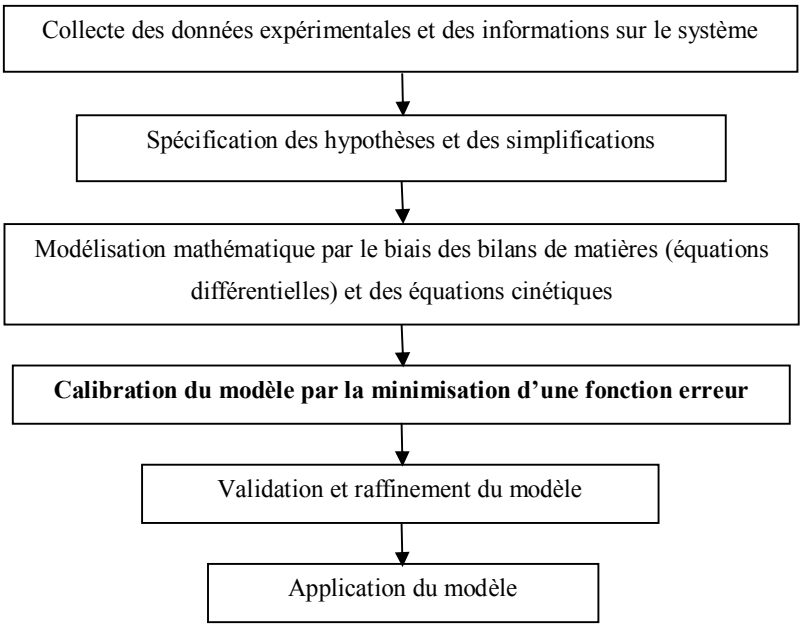


Figure II.4. Phases du processus de modélisation en biotechnologie.

II.7.2. Apprentissage des réseaux de neurones de type perceptron multicouches

L'apprentissage des réseaux de neurones, comme on a vu dans le 1er chapitre, constitue une phase cruciale dans le développement d'un modèle neuronal durant laquelle on calcul les poids des neurones de telle manière que les sorties du réseau soient aussi proches que possibles des sorties désirées. Lors des revues de l'application des RN en biotechnologie au fil des années, 90 % des recherches utilisaient l'algorithme de rétro-propagation du gradient dans l'apprentissage des RN multicouches. Bien que cet algorithme s'est avéré efficace dans certaines applications, il présente cependant deux inconvénient majeurs (Haridas et Devi.M 2010) :

i. Taux de convergence est relativement lent ;

ii. Risque d'être bloqué ou piégés dans un minimum local.

Pour surmonter le problème de taux de convergence lent, les chercheurs ont proposé plusieurs solutions, la plus récente d'entre elles et qui a prouvé son efficacité (bonne précision et bonne capacité de généralisation) consiste à coupler le RN avec l'algorithme d'OEP (RNOEP) (Haridas et Devi.M 2010).

Cette approche fructueuse (RNOEP) trouve de dénombrables applications dans divers domaines, cependant en biotechnologie on en a trouvé jusqu'à présent que deux applications :

- Shee et ces collaborateurs (Shee *et al.* 2011) ont développé un classificateur efficace à l'aide d'un RN multicouches pour la classification des séquences de protéines. Pour cela ils ont combiné l'algorithme d'OEP et l'algorithme de rétro-propagation, donnant naissance ainsi à une méthode hybride d'apprentissage des RN qui a donné de bons meilleurs résultats par comparaison à l'algorithme de rétro-propagation et donc un classificateur efficace.
- Pour ce qui de notre application Khaouane et ces collaborateurs (Khaouane *et al.* 2012a) elle sera détaillée dans le 4ème chapitre.

II.7.3. Optimisation des modèles neuronaux

L'algorithme d'OEP est utilisé pour rechercher l'espace d'entrée optimal d'un modèle neuronal et trouver ainsi la sortie correspondante optimale du modèle. Huang et ces collaborateurs ont couplé un modèle neuronal avec l'algorithme d'OEP pour optimiser la production de l'acide gamma-amino-butyrique (Huang *et al.* 2007). Liu et co-auteurs ont appliqué un RN à fonction radiale avec une variante de l'algorithme d'OEP pour optimiser les conditions de culture de *Streptococcus zooepidemicus* et ceux afin de maximiser la production d'acide hyaluronique (Liu *et al.* 2009).

II.8. Conclusion

Dans ce chapitre, nous avons donné des notions de base nécessaire dans la suite de nos travaux sur l'algorithme d'OEP. Après un bref historique sur l'origine de l'algorithme, nous avons présenté sa description mathématique, ces étapes de conception et ces caractéristiques principales.

Dans un deuxième temps, nous avons exposé d'une façon générale le confinement des particules, la topologie du voisinage, ainsi que quelques variantes de l'algorithme. Nous avons terminé ce chapitre par les différentes applications de l'algorithme d'OEP en bioprocédés notamment pour l'estimation des paramètres physiques de modèles mathématiques, l'apprentissage des RN et l'optimisation des modèles neuronaux.

Chapitre III

Modélisation de la fermentation fed-batch de pleuromutiline par les réseaux de neurones

III.1. Introduction

Les réseaux de neurones artificiels (RN) ont acquis une grande popularité dans la dernière décennie en raison de leur parallélisme élevé, robustesse (Basheer et Hajmeer 2000) et, plus intéressant encore, leur capacité inhérente à extraire à partir des données expérimentales, les relations fortement non-linéaires et complexes entre les variables du problème (Si-Moussa *et al*. 2008) sans aucune connaissance détaillée du système (Bryjak *et al*. 2004).

Dans ce chapitre, nous avons développé trois modèles neuronaux de type feedforward, afin d'évaluer les quantités difficilement mesurables on line, à savoir les concentrations de la biomasse, du substrat et du produit, à partir de variables facilement mesurables dans la fermentation fed-batch de pleuromutiline. Ces modèles neuronaux peuvent trouver des applications très intéressantes dans le contrôle numérique du processus en question et donc permettront l'amélioration de la reproductibilité du processus et l'augmentation de la productivité. Les résultats de la modélisation neuronale de cette biosynthèse de pleuromutiline ont été publiés dans le "Brazilian Journal of Chemical Engineering" (Khaouane *et al.* 2013).

Le chapitre est organisé en deux parties :

- Dans la première partie nous présentons les matériels et méthodes utilisés : Nous examinons d'abord la base de données collectées. Ensuite, nous présentons la procédure de développement et d'optimisation de l'architecture des trois modèles neuronaux de type feedforward employés respectivement pour décrire la croissance microbienne, la consommation du substrat et la formation du produit. Par la suite, une comparaison des trois modèles neuronaux optimisés avec les modèles cinétiques est réalisée afin de mieux illustrer la méthode. Enfin, une analyse de sensibilité, par deux méthodes différentes, est appliquée aux trois RN optimisés.

- Dans la deuxième partie, les différents résultats prometteurs obtenus sont discutés : Dans un premier temps, nous présentons les architectures optimales des modèles neuronaux ainsi que leurs performances de prédiction. Ensuite, après l'extraction de formules mathématiques à partir de ces modèles et la présentation des résultats de comparaison, nous étudions leurs performances d'interpolation. Cette partie ce termine par les résultats de l'analyse de sensibilité.

III.2. Matériels et méthodes

III.2.1. Collecte des données, prétraitement et analyse

Dix-neuf ensembles de données (lots) ont été fournis par Antibiotical, une filiale du groupe pharmaceutique algérien Saidal, contenant des données sur : la concentration en biomasse (*X*), la consommation de glucose (*S*), la concentration en pleuromutiline (*P*), l'évolution du *pH* et l'indice de réfraction du bouillon de fermentation (*IR*).

Les données expérimentales rassemblées ont été interpolées en utilisant une fonction MATLAB "spline cubique" quand nécessaire et lissées avec un filtre numérique à moyenne mobile (en anglais "Digital Moving Average Filter"). Sans lissage, le RN a tendance à capturer le bruit dans le système plutôt que les mécanismes fondamentaux du processus sous-jacent (Laursen *et al.* 2007), ce qui entraîne une efficacité inférieure et un temps plus long dans l'apprentissage du RN.

III.2.2. Procédure de développement des modèles

III.2.2.1. Description des modèles

Un réseau de neurones tel que décrit dans le premier chapitre est un cadre de calcul qui est inspiré par les systèmes neuronaux biologiques. Il se compose d'un certain nombre d'unités de traitement simple interconnectés appelés neurones artificiels. L'un des paradigmes les plus populaires des réseaux de neurones appliqués à la modélisation d'un large éventail de systèmes non linéaires, en particulier des processus d'ingénierie chimique et biologique, est le réseau de neurones de type feedforward (RNFF) à algorithme de rétro-propagation du gradient (Lee et Park 1999; Silva *et al.* 2000), qui est utilisé dans le présent chapitre avec un apprentissage supervisé et pour un horizon de prévision.

Trois modèles de RNFF caractérisés par la même structure générale (une couche d'entrée, une ou deux couches cachées et une couche de sortie) ont été conçus pour prédire respectivement les profils de concentration de la biomasse (RN*X*), du glucose (RN*S*) et du produit (RN*P*) au cours de la fermentation fed-batch de la pleuromutiline. Le choix de trois RN au lieu d'un seul regroupant les trois sorties simultanément se justifie par l'intensité de la liaison qui existe entre ces paramètres. Cette intensité de liaison à été obtenue par le calcul des coefficients de corrélation de Pearson (*R*) :

$$R(Z,Y)=\frac{\sum_i\left(z_i-\bar{z}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_i\left(z_i-\bar{z}\right)^2}\sqrt{\sum_i\left(y_i-\bar{y}\right)^2}} \qquad \textbf{(III.1)}$$

Où z_i est la i^{ème} valeur du vecteur Z (X, S ou P) ; y_i est la i^{ème} valeur du vecteur Y (X, S ou P) ; $\bar{z}$ et $\bar{y}$ sont respectivement les moyennes arithmétiques des vecteurs Z et Y.

Le tableau III.1 regroupe les coefficients de corrélation entre X, S et P. On remarque que, le degré de corrélation est élevé entre X et S et entre S et P ($\pm 0.75 \leq R \leq \pm 1$), tandis qu'entre X et P le degré de corrélation est modéré ($\pm 0.25 \leq R < \pm 0.75$) ; montrant ainsi l'existence de liaison linéaire entre X et S, entre S et P et entre X et P.

Tableau III.1. Matrice de corrélations entres les outputs (X, S et P).

	X	***S***	***P***
X	**1**	-0.9	0.73
S	-0.9	**1**	-0.75
P	0.73	-0.75	**1**

Les entrés des trois modèles sont : le temps de fermentation (t), le pH, l'indice de réfraction (IR), la concentration initiale d'inoculum (X_0) et la concentration initiale en glucose (S_0). Une analyse préliminaire fondée sur la méthode "stepwise" utilisée pour tester les contributions des entrées du RN a révélé que ces variables présentaient la plus forte influence sur la performance des processus ; plus de détail sur cette technique peut être trouvé dans (Gevrey *et al.* 2003). Les sorties sont la biomasse pour le premier modèle (RNX), la consommation du substrat pour le deuxième modèle (RNS) et le produit pour le troisième modèle (RNP). L'analyse statistique des données d'entrée et de sortie, en termes de la moyenne arithmétique $\bar{x}$ (Equation III.2), de la déviation standard (Equation III.3), du minimum et du maximum, est indiquée dans le tableau III.2.

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i \qquad \textbf{(III.2)}$$

$$DS=\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^{1/2} \qquad \textbf{(III.3)}$$

Tableau III.2. Analyse statistique des données d'entrée et de sortie.

Variables	Moyenne	*DS*	Min	Max
***t* (h)**	140	83,7	0	280
pH	7	0,3	5,69	7,67
IR	2,9	0,9	1,80	6,67
***X_0* (%)**	4	1,7	1	7
***S_0* (g/l)**	24,8	2,8	19,69	30,36
***X* (%)**	45,1	15,7	1	63,17
***S* (g/l)**	15,1	4,3	9,30	30,36
***P* (mg/g)**	1,7	1,3	0	7,25

III.2.2.2. Développement des modèles

Dans ce chapitre, une procédure basée sur le développement et l'optimisation de l'architecture d'un RNFF est avancée. Elle est basée, comme décrit sur la figure III.1, sur la conception de quatre sous-modèles de RNFF qui se distinguent par le type de fonction de transfert et le type d'algorithme d'apprentissage couramment utilisé en biotechnologie.

Afin de rendre l'apprentissage des quatre sous-modèles plus efficace en empêchant la fonction de transfert de devenir saturé et rendre l'apprentissage des réseaux très rapide, les entrées et les sorties ont été normalisées dans l'intervalle (IN) 0 à 1 et -1 à 1 suivant la fonction de transfert attribuée à la couche cachée (FTCC) (log-sigmoïde ou tangente hyperbolique sigmoïde, respectivement), en utilisant l'équation de normalisation suivante :

$$x_N = (y_{\max} - y_{\min})\left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) + y_{\min} \qquad \textbf{(III.4)}$$

Où x_N est la valeur normalisée du paramètre x (*t, pH, IR, X_0, S_0, X, S* ou *P*) ; x_{max} et x_{min} sont respectivement les valeurs maximales et minimales de x ; y_{min} *et* y_{max} peuvent prendre les valeurs 0 et 1 ou -1 et 1.

Pour tous les sous-modèles, la fonction de transfert linéaire a été attribuée à la couche de sortie.

Deux techniques de généralisation sont souvent employées avec les algorithmes d'apprentissage de rétro-propagation du gradient les plus efficaces et les plus couramment utilisés en biotechnologie. Le sur-apprentissage ou la faible capacité de généralisation survient quand un réseau neuronal sur-apprend pendant la phase d'apprentissage. En conséquence, un tel modèle trop

bien formé ne peut pas fonctionner correctement sur un ensemble de données non vue dans la phase d'apprentissage à cause de son manque de capacité de généralisation. Pour surmonter ce problème, les méthodes d'arrêt prématuré (en anglais "early stopping") et de régularisation bayésienne ont été appliquées respectivement avec les algorithmes d'apprentissage de Levenberg-Marquardt (trainlm en code MATLAB) et de régularisation bayésienne (trainbr en code MATLAB).

La méthode d'arrêt prématuré est une pratique très courante dans l'apprentissage du réseau de neurones et produit souvent des réseaux possédant de grandes capacités de généralisation. Cette technique est basée sur la répartition des ensembles de données en trois parties : apprentissage, validation et test. Le réseau est formé avec l'ensemble d'apprentissage pour minimiser l'erreur et vérifier avec l'ensemble de validation après chaque itération pour éviter le sur-apprentissage et la perte de la capacité de généralisation (Bishop 1995; Pigram et MacDonald 2000). Comme vérification finale, l'ensemble de test est utilisé pour vérifier le bon fonctionnement et la capacité de généralisation du réseau formé (Hagan *et al.* 1996; Hirschen et Schafer 2006; Asensio-Cuesta *et al.* 2010).

La régularisation bayésienne est une approche plus sophistiquée pour améliorer la généralisation. Elle a d'abord été utilisée par MacKay (1991). Cette méthode non seulement minimise une combinaison linéaire des erreurs quadratiques et des poids, plutôt que la minimisation des erreurs quadratiques, mais modifie également la combinaison linéaire de sorte qu'à la fin de l'apprentissage, le réseau résultant possède de bonnes qualités de généralisation (MacKay 1991; Danaher *et al.* 2004).

Dans ce travail, les données disponibles (551 échantillons) ont été réparties aléatoirement en trois sous-ensembles distincts (Figure III.2), en réservant 60 % de données (385 échantillons) pour la phase d'apprentissage, 20 % (83 échantillons) pour la phase de validation et les 20 % restantes (83 échantillons) pour la phase de test. Lorsque la technique d'arrêt prématuré est appliquée, les ensembles d'apprentissage et de validation sont utilisés aux fins décrites précédemment, alors qu'avec la régularisation bayésienne l'ensemble de validation est ajouté à l'ensemble d'apprentissage pour former les modèles. L'ensemble de test est utilisé dans les deux techniques pour tester la généralisation des sous-modèles formés.

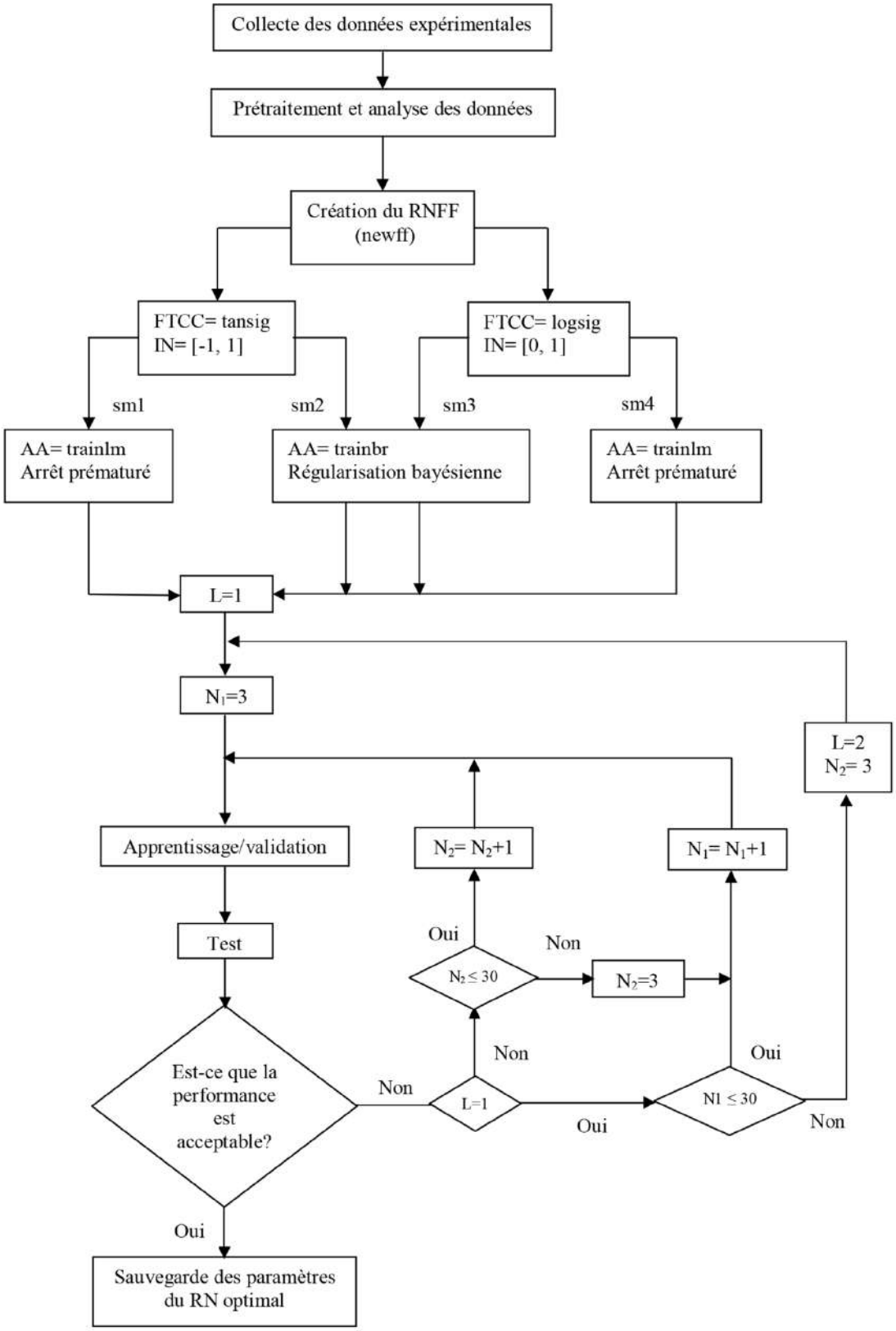


Figure III.1. Organigramme pour le développement et l'optimisation de l'architecture d'un RNFF.

Afin d'optimiser l'architecture de chaque sous-modèle et déterminer ainsi le nombre de couches cachées et le nombre de nœuds dans chaque couche cachée, une procédure d'essai et erreur est exécutée (Figure III.1). Le nombre de couches cachées (L) varie de 1 à 2 couches et le nombre de neurones (N) dans chaque couche cachée varie de 3 à 30 neurones (Heaton 2005).

Les performances des sous-modèles sont évaluées, en termes de la racine de l'erreur quadratique moyenne (*REQM*), en utilisant l'équation suivante :

$$REQM = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_{i,\exp} - Y_{i,pred}\right)^2}{n}} \qquad \textbf{(III.5)}$$

Où n est le nombre total de données ; $Y_{i,pred}$ représente la sortie prédite par le modèle neuronal ; $Y_{i,exp}$ est la valeur expérimentale.

La procédure d'élaboration du RNFF décrit ci-dessus a été réalisée par l'élaboration d'un programme MATLAB sous MATLAB Neural Network Toolbox ver.7.10. Ce programme est utilisé pour optimiser l'architecture des trois modèles neuronaux employés respectivement pour prédire les profils de concentration de la biomasse (RN*X*), du glucose (RN*S*) et du produit (RN*P*).

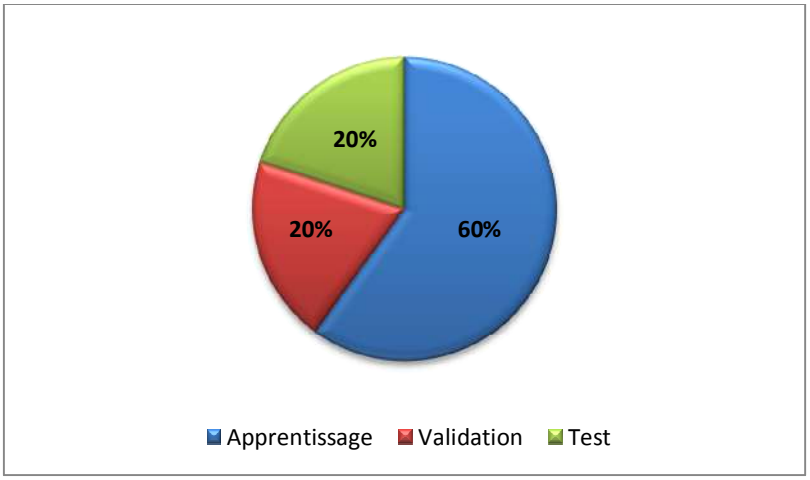


Figure III.2. Représentation des % de division de la base de données.

III.2.3. Comparaison avec les modèles cinétiques non structurés

La culture fed-batch de pleuromutiline est mise en œuvre initialement dans un mode batch, puis alimentée par une solution concentrée de substrat limitant (forme non diluée), résultant en une augmentation non significative du volume. La culture fed-batch à volume fixe peut être décrite par les équations suivantes de bilan de masse:

$$\frac{dX}{dt} = \mu X - \frac{F}{V} X \qquad \textbf{(III.6)}$$

$$\frac{dS}{dt} = \frac{F}{V}(S_0 - S) - \frac{1}{Y_{X/S}} \frac{dX}{dt} - \frac{1}{Y_{P/S}} \frac{dP}{dt} - m_S X$$

(III.7)

$$\frac{dP}{dt} = \pi X - \frac{F}{V} P \qquad \textbf{(III.8)}$$

Où *X, S* et *P* sont respectivement les concentrations de la biomasse, du substrat et de la pleuromutiline ; *t* est le temps de fermentation ; μ est la vitesse spécifique de croissance ; π est la vitesse spécifique de formation de produit ; *V* est le volume du bioréacteur industriel (culture) ; S_0 et *F* sont respectivement la concentration initiale du substrat et le débit d'alimentation du milieu ajouté au bioréacteur ; m_S est le coefficient de maintenance ; $Y_{X/S}$ est le rendement de la biomasse par unité de masse de substrat et $Y_{P/S}$ est le rendement du produit par unité de masse de substrat.

Comme le volume *V* du bioréacteur industriel est très grand par rapport à l'alimentation *F*, le terme *F/V* est négligeable et par conséquent les équations de bilans deviennent (Pirt 1979; Stanbury *et al.* 1999) :

$$\frac{dX}{dt} = \mu X \qquad \textbf{(III.9)}$$

$$\frac{dS}{dt} = -\frac{1}{Y_{X/S}} \frac{dX}{dt} - \frac{1}{Y_{P/S}} \frac{dP}{dt} - m_S X \qquad \textbf{(III.10)}$$

$$\frac{dP}{dt} = \pi X \qquad \textbf{(III.11)}$$

Les vitesses spécifiques de croissance et de formation de la pleuromutiline sont modélisés par les équations suivantes (Patnaik 2001) :

$$\mu = \frac{\mu_{\max} S}{k_S + S} \qquad \textbf{(III.12)}$$

$$\pi = \frac{\pi_{\max} S}{k_S + S + S^2/k_i} \qquad \textbf{(III.13)}$$

Où k_S est la constante de Monod ; k_i est la constante d'inhibition ; μ_{max} et π_{max} sont respectivement les valeurs maximales de μ et π.

Les modélisation cinétiques ont été effectuées par élaboration d'un programme MATLAB basé sur les quatre étapes suivante (Cutlip et Shacham 2008):

1. Initialisation des paramètres cinétiques et intégration des équations différentielles par l'utilisation de la fonction MATLAB "ode45" et ceux dans le but de calculer les valeurs des variables dépendantes ;
2. Calcule de la somme des carrés de la différence entre les valeurs calculées et expérimentales des variables dépendantes ;
3. Application d'un programme d'optimisation basé sur la fonction MATLAB "fminsearch", qui modifient les valeurs des paramètres cinétiques de manière à obtenir le minimum de la somme des carrés ;
4. Réintégration des équations différentielles en utilisant les valeurs optimales des paramètres cinétiques.

Les valeurs des variables dépendantes ainsi obtenues pour chaque lot sont ensuite comparées à celles obtenues par les modèles neuronaux optimisés.

III.2.4. Analyse de sensibilité

Pour voir la contribution ainsi que le profil de variation de chaque variable d'entrée (*t, pH, IR,* X_0 et S_0) sur la sortie de chaque réseau (RN*X*, RN*S* et RN*P*), On a procédé à une analyse de sensibilité par deux méthodes différentes : Méthode de "Poids" (en anglais 'Weights' method) et méthode de "profil" (en anglais 'Profile' method).

III.2.4.1. Méthode de "Poids"

Cette méthode, proposée d'abord par Garson (Garson 1991) et reprise par Goh (Goh 1995), donne une quantification de l'importance relative (*Ir*) des différentes entrées sur la sortie du RN (Gevrey *et al*. 2003). Elle se base sur le partitionnement des poids de connexions en :

- Poids de connexions entrée –cachée (w_{ji}^{I}) ;
- Poids de connexions cachée- sortie (w_{1j}^{H}).

Le processus de calcul, qui se compose de quatre étapes, est schématisé sur l'organigramme de la figure III.3.

III.2.4.1. Méthode de "Profil"

Cette méthode a été proposée par Lek (Lek *et al.* 1995; Lek *et al.* 1996a; Lek *et al.* 1996b). L'idée générale est d'étudier successivement chaque variable d'entrée quand les autres sont alors bloquées à des valeurs fixes. Le principe de cet algorithme est de construire une matrice fictive relative à la plage de toutes les variables d'entrée (Gevrey *et al.* 2003). La figure III.4 présente un schéma explicatif général de la méthode de profil. Lorsque la variable d'entrée X_1 est distribuée sur 12 niveaux de variation entre son minimum et son maximum, les autres variables d'entrées sont maintenues fixes, successivement au minimum, le premier quartile, la médiane, le troisième quartile et maximum. Pour chaque valeur de X_1, cinq réponses sont obtenues et c'est la valeur médiane des cinq réponses prédites par RN qui est prise en compte. Alors le profil de la grandeur de sortie (réponse) peut être tracé pour des valeurs de l'échelle de la variable X_1. Cette procédure est répétée pour les autres variables.

Nous avons appliqué cette méthode pour déterminer le profil de contribution de chaque variable d'entrées simulées (*t, pH, IR,* X_0 et S_0) sur la biomasse prédite par le RN*X*, sur la consommation du sucre calculée par le RN*S* et sur la formation du produit prédite par le RN*P*.

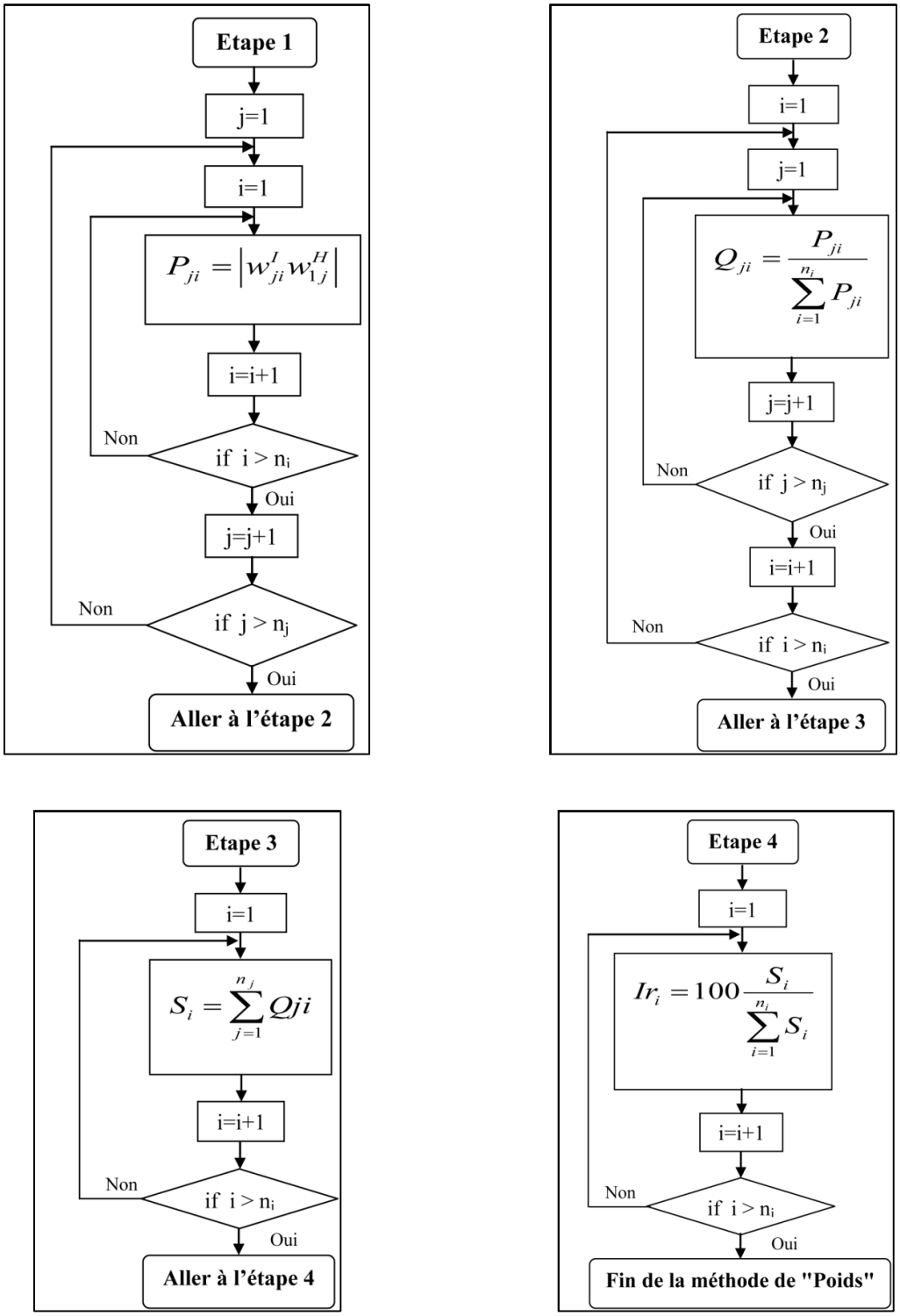


Figure III.3. Organigramme de la méthode de "poids".

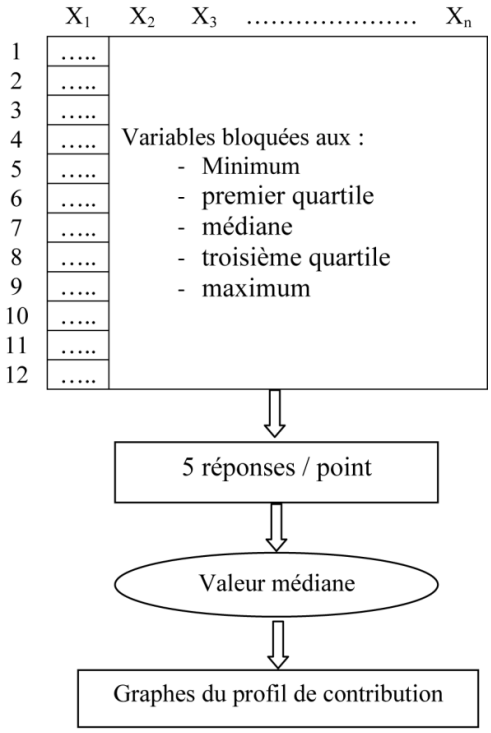


Figure III.4. Schéma explicatif de la méthode de "Profil" (Gevrey *et al.* 2003).

III.3. Résultats et discussion

III.3.1. Performances des modèles

Selon la discussion précédente, trois modèles neuronaux ont été développés dans le but de prédire les profils de concentration de la biomasse (RN*X*), du glucose (RN*S*) et de la pleuromutiline (RN*P*) pendant la fermentation fed-batch de *Pleurotus mutilus*. Afin d'optimiser leur structure, quatre sous-modèles suivant les algorithmes d'apprentissage et les fonctions de transfert utilisées, ont été développés pour chaque modèle neuronal.

Le tableau III.3 et la figure III.5 résument les performances des sous-modèles, en termes de la *REQM*, pour chaque modèle neuronal. Pour le RN*X* et le RN*P*, le sous-modèle sm3 donne l'erreur minimale par rapport aux autres sous-modèles. Pour le RN*S*, le sous-modèle sm2 est le plus performant avec une erreur de 0.1234 g/L. Les structures obtenues des modèles neuronaux optimisés sont représentées dans le tableau III.4. Une couche cachée était suffisante pour prévoir avec assez de précision les profils de concentration de la biomasse, du substrat et du produit. Il a été prouvé que l'algorithme d'apprentissage de rétro-propagation de type régularisation bayésienne couplée avec sa technique de généralisation correspondante augmente la capacité d'apprentissage.

La figure III.6 montre la comparaison entre les valeurs expérimentales et les valeurs calculées par les modèles neuronaux optimisés pour la totalité de la base de données. Le tableau III.5 rassemble les vecteurs de régression linéaire pour les trois modèles neuronaux (RN*X*, RN*S* et RN*P*), ce tableau montre que les paramètres de régression les moins favorables sont ceux obtenus pour la prédiction du sucre. Il est prouvé que l'approche neuronale proposée donne des résultats satisfaisants avec des valeurs des vecteurs de régression approchant l'idéal [i.e. α=1 (pente), β=0 (ordonnée à l'origine), R=1 (coefficient de corrélation)] dans l'ajustement des profils de la biomasse, du glucose et de la pleuromutiline.

Tableau III.3. Performances des modèles neuronaux et leurs sous-modèles pour l'ensemble de test.

Sous-modèles	**RN*X***		**RN*S***		**RN*P***	
	Nombre de neurones	***REQM* (%)**	**Nombre de neurones**	***REQM* (g/L)**	**Nombre de neurones**	***REQM* (10^{-2} mg/g)**
sm1	24	1.1046	21	0.2351	23	0.54
sm2	29	0.4826	**25**	**0.1234**	21	0.18
sm3	**29**	**0.4624**	18	0.1305	**27**	**0.16**
sm4	26	1.1572	27	0.2757	20	0.6

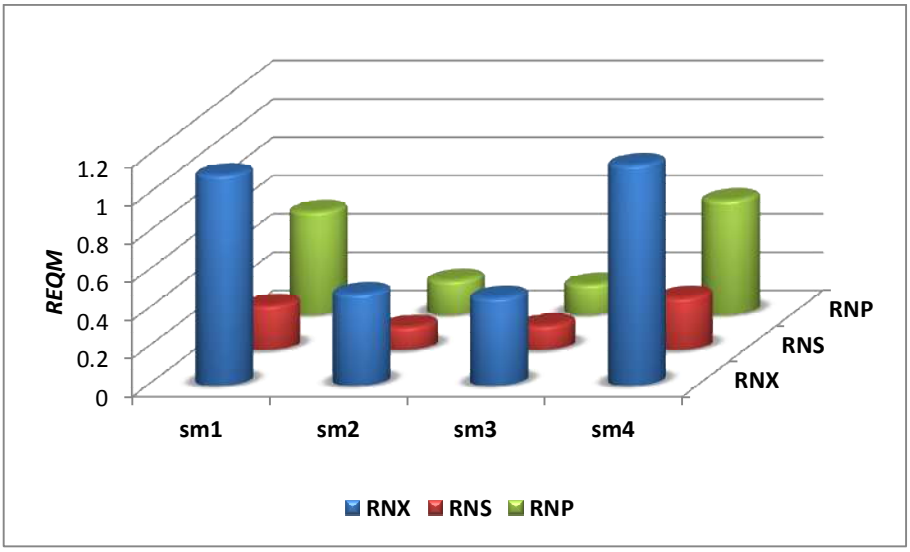
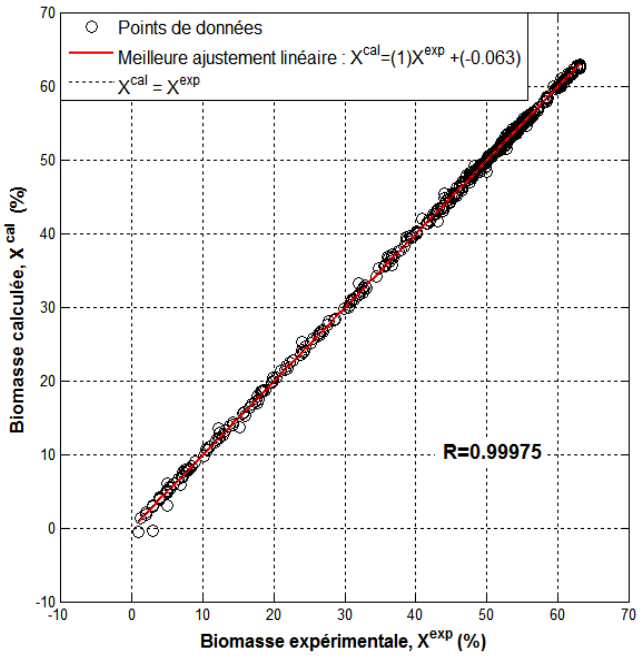


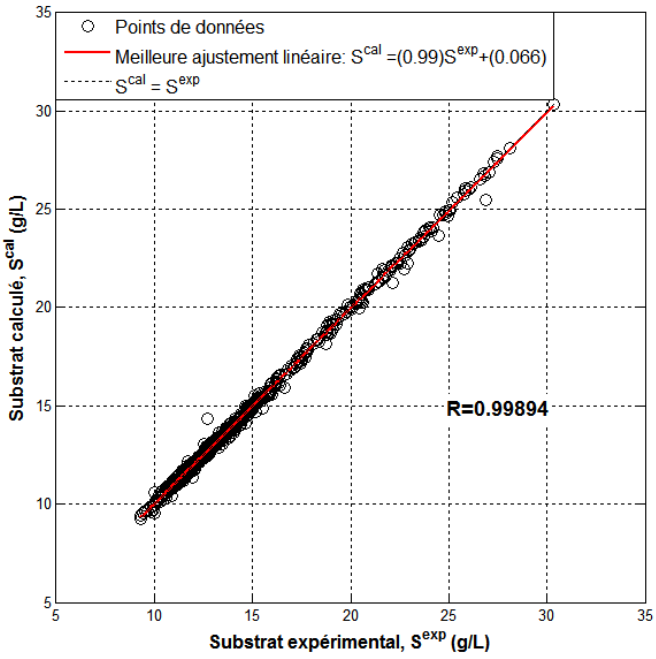
Figure III.5. Performances des modèles neuronaux (RN*X*, RN*S* et RN*P*) et leurs sous-modèles (sm1, sm2, sm3 et sm4) pour la base de test.

Tableau III.4. Structure des modèles neuronaux optimisés.

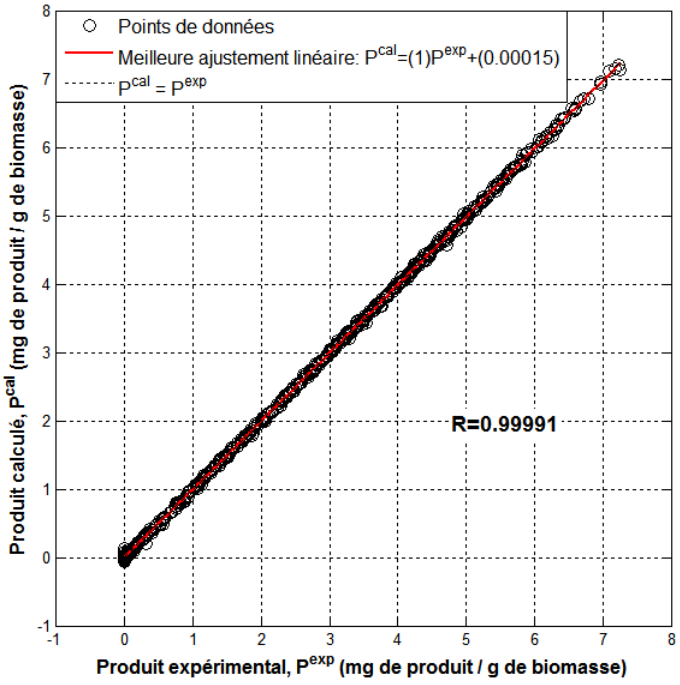
Modèles neuronaux	Algorithme d'apprentissage	Couche d'entrée	Couche cachée		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
RN*X*	trainbr	5	29	logsig	1	purelin
RN*S*	trainbr	5	25	tansig	1	purelin
RN*P*	trainbr	5	27	logsig	1	purelin


Points de données
Meilleure ajustement linéaire : X^cal=(1)X^exp +(-0.063)
X^cal = X^exp
R=0.99975
Biomasse calculée, X cal (%)
Biomasse expérimentale, X^exp (%)

(a)


Points de données
Meilleure ajustement linéaire: S^cal =(0.99)S^exp+(0.066)
S^cal = S^exp
R=0.99894
Substrat calculé, S^cal (g/L)
Substrat expérimental, S^exp (g/L)

(b)



(c)

Figure III.6. Comparaison entre les valeurs expérimentales et calculées par les modèles neuronaux optimisés pour la totalité de l'ensemble de données : **(a)** biomasse, **(b)** glucose, **(c)** produit.

Tableau III.5. Vecteurs de régression linéaire issue de l'équation linéaire : $y^{cal} = \alpha y^{exp} + \beta$, [$\alpha$ (pente), β (ordonnée à l'origine), R (coefficient de corrélation)].

		Base d'apprentissage	**Base de test**	**Totalité de la base**
α	**RN*X***	1	1	1
	RN*S*	1	0.98	0.99
	RN*P*	1	1	1
β	**RN*X***	0.0074	-0.35	-0.063
	RN*S*	0.017	0.29	0.066
	RN*P*	0.0005	0.0011	0.0002
R	**RN*X***	0.9999	0.9991	0.9998
	RN*S*	0.9995	0.9967	0.9989
	RN*P*	0.9999	0.9998	0.9999

III.3.2. Temps d'exécution

Il nous a fallu 4 mois pour la collecte des données, le prétraitement et la programmation des RN en MATLAB. L'exécution des programmes est réalisée dans l'interface Windows 7 avec 32 bits et une RAM de 2 GB, le temps d'exécution des programmes est donné dans le tableau III.6.

Tableau III.6. Temps d'exécution.

RN	Sous-modèle	Temps d'exécution (s) du programme du RN	Temps d'exécution (s) de simulation du RNO
RN*X*	**sm1**	773.8758	0.0326
	sm2	$1.0855\ 10^4$	0.0540
	sm3	$1.5743\ 10^4$	0.0720
	sm4	771.5629	0.0340
RN*S*	**sm1**	849.6533	0.0828
	sm2	$9.8922\ 10^3$	0.0720
	sm3	$1.3982\ 10^4$	0.0625
	sm4	698.7762	0.0880
RN*P*	**sm1**	908.0220	0.1268
	sm2	$1.1125\ 10^4$	0.0990
	sm3	$1.5393\ 10^4$	0.1250
	sm4	$1.1888\ 10^3$	0.1150

III.3.3. Expressions mathématique des modèles neuronaux optimisés

À partir du RN*S* optimisé et qui est représenté sur la figure III.7, on peut exprimer l'assimilation du substrat par un modèle mathématique qui y incorpore toutes les entrées E_i (t, pH, IR, X_0, S_0) comme suit:

Les sorties Z_j de la couche cachée:

$$Z_j = f_H\left[\sum_{i=1}^{5} w_{ji}^I E_i + b_j^H\right] = \frac{2}{1+\exp(-2\sum_{i=1}^{5} w_{ji}^I E_i + b_j^H)} - 1 \qquad \textbf{(III.14)}$$

$j=1, 2, \ldots, 25$

La sortie S:

$$S = f_0\left[\sum_{j=1}^{25} w_{1j}^{H} Z_j + b_1^{O}\right] = \sum_{j=1}^{25} w_{1j}^{H} Z_j + b_1^{O} \qquad \textbf{(III.15)}$$

La combinaison des équations (IV.14) et (IV.15) mène à la formule mathématique suivante qui décrit l'assimilation de substrat en prenant en compte toutes les entrées E_i (t, pH, IR, X_0, S_0):

$$S = \sum_{j=1}^{25} w_{1j}^{H}\left[\frac{2}{1+\exp(-2\sum_{i=1}^{5} w_{ji}^{I} E_i + b_j^{H})} - 1\right] + b_1^{O} \qquad \textbf{(III.16)}$$

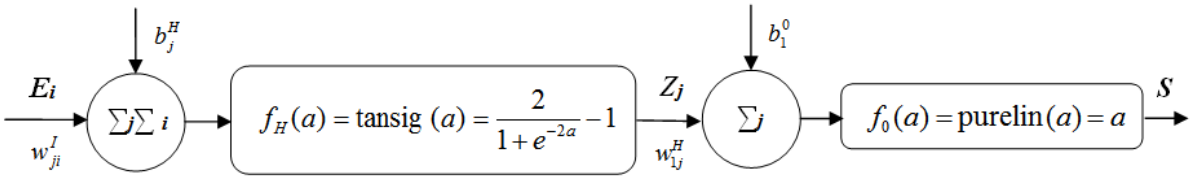


Figure III.7. Représentation schématique du RN*S* optimisé.

De même, les concentrations de la biomasse et de la pleuromutiline peuvent être exprimées par des équations mathématiques extraites des RN*X* et RN*P* optimisés comme suit:

$$X = \sum_{j=1}^{29} w_{1j}^{H}\left[\frac{1}{1+\exp(-\sum_{i=1}^{5} w_{ji}^{I} E_i + b_j^{H})}\right] + b_1^{O} \qquad \textbf{(III.17)}$$

$$P = \sum_{j=1}^{27} w_{1j}^{H}\left[\frac{1}{1+\exp(-\sum_{i=1}^{5} w_{ji}^{I} E_i + b_j^{H})}\right] + b_1^{O} \qquad \textbf{(III.18)}$$

Ou w_{ji}^{I} représente le poids synaptique de la connexion du neurone i de la couche d'entrée au neurone j de la couche cachée ; w_{1j}^{H} représente le poids synaptique de la connexion du neurone j de la couche cachée au neurone de la couche de sortie ; b_j^{H} et b_1^{O} représentent respectivement les biais du neurone j de la couche cachée et du neurone de la couche de sortie. Les matrices des poids et biais, des trois RN optimisés, sont donnés dans les tableaux III.7, III.8 et III.9.

Comme ces équations mathématiques, dérivant des modèles neuronaux optimisés, contiennent juste le degré exigé de complexité et incluent dans leurs formules les caractéristiques

pertinentes du bio-processus à savoir les conditions de fonctionnement et les conditions initiales, ils peuvent donc facilement être appliqués dans des boucles de contrôle du bio-processus.

Tableau III.7. Poids et biais du RN*X* optimal.

Connexions Couche d'entrée-Couche cachée						**Connexions Couche cachée-Couche de sortie**	
Poids					**Biais**	**Poids**	**Biais**
w_{j1}^{I}	w_{j2}^{I}	w_{j3}^{I}	w_{j4}^{I}	w_{j5}^{I}	b_{j}^{H}	w_{1j}^{H}	b_{1}^{0}
-0.8354	-1.2637	0.1349	2.0221	-3.2050	2.0109	0.7177	
-0.1707	0.9634	-0.7157	0.3131	0.1681	-0.4961	-2.7030	
0.2411	-1.5332	-0.1923	-2.4550	-0.7520	0.6709	1.3862	
-2.8765	-1.2900	-0.6103	0.7585	-0.6989	-0.0322	-0.6418	
0.2558	-0.0494	-0.8620	-0.9874	-1.9360	-0.3356	-1.8420	
-0.1145	1.7908	0.0359	1.5959	0.2494	-0.7446	1.4893	
0.1477	-1.5980	-0.9304	-3.2265	-1.0579	-1.9098	1.1202	
-0.0596	1.3392	0.0504	1.4031	0.6438	0.8670	2.2706	
-2.2511	-0.6470	0.0841	1.0809	-0.1548	-0.1746	1.0095	
-1.5418	0.4175	0.0599	0.7850	-1.3099	0.7502	-0.4819	
-0.4349	0.4342	0.1578	-0.4660	-2.3435	-0.5555	0.4432	
-0.9984	-0.7252	-1.5541	0.6684	1.2499	-0.5206	-1.7870	
-1.9264	-0.3633	0.2436	2.1841	-2.4865	-1.1252	1.2719	
0.0922	-0.8548	0.3546	1.5943	-2.8940	-0.2157	-2.3448	
0.1605	-1.1330	0.8118	0.6638	0.3894	0.4944	-1.7971	-0.7409
-0.0227	-1.0143	0.1729	2.6873	-4.9177	-0.5492	1.7748	
0.0560	0.8807	1.5489	-0.2283	0.5415	-0.6904	1.3860	
-0.0154	0.2886	-1.0549	-1.1250	0.0472	-0.2124	-2.0431	
-1.1562	-0.2979	-1.6549	-0.3172	-0.7602	-0.5490	1.2609	
1.7440	-0.3495	-0.6322	0.0527	-0.2098	1.7447	2.5359	
-2.4835	0.2614	-1.0163	1.1579	1.2635	-0.3993	0.2946	
2.8277	0.8605	-0.6175	-0.8095	0.8307	0.5505	0.4555	
1.9439	-0.2521	-0.5233	-2.1297	1.7090	0.9802	1.3903	
0.9562	-0.7169	2.2102	-1.1279	-2.0820	1.6698	-0.6627	
-1.9522	-0.6736	0.8961	0.8283	4.3211	-2.4213	0.5216	
-0.8258	-0.6182	-1.2898	0.1798	1.2874	-0.1099	2.3968	
1.1010	-0.4139	0.8712	-0.4649	-2.3572	-0.1412	0.9812	
2.0325	0.3207	0.8679	0.5312	0.6171	-0.5889	0.6610	
0.8268	-0.2994	-1.2249	-1.0204	-1.7741	-0.1208	1.1387	

Tableau III.8. Poids et biais du RN*S* optimal.

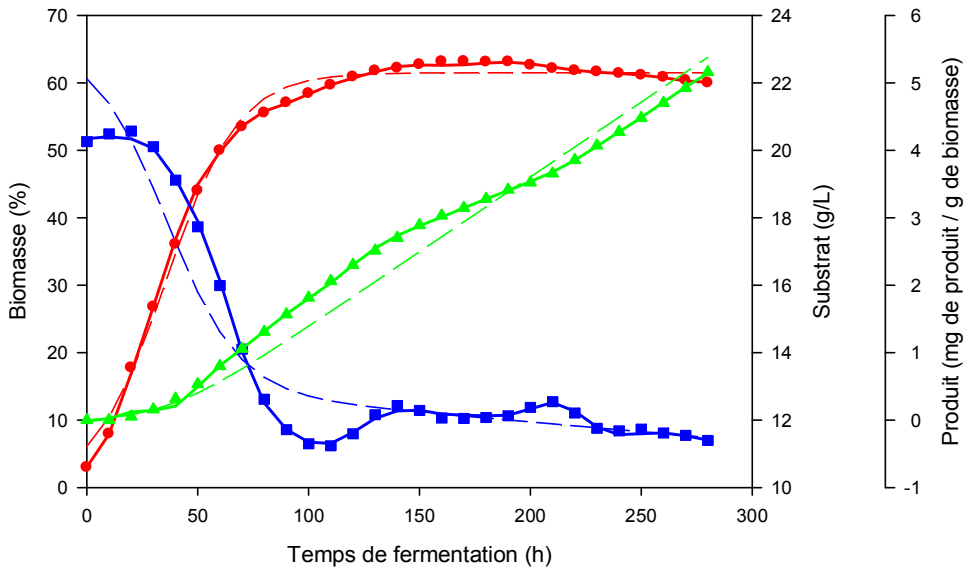
Connexions Couche d'entrée-Couche cachée						**Connexions Couche cachée-Couche de sortie**	
Poids					**Biais**	**Poids**	**Biais**
w_{j1}^{I}	w_{j2}^{I}	w_{j3}^{I}	w_{j4}^{I}	w_{j5}^{I}	b_{j}^{H}	w_{1j}^{H}	b_{1}^{0}
1.4029	0.8714	-1.3524	0.2145	0.6317	-0.5115	-0.7805	
0.4383	-0.5951	-1.3384	1.5204	1.6494	0.0779	-1.7608	
-1.3097	-0.2949	-0.8419	-0.1914	0.5173	-1.1387	2.3025	
-0.3193	-0.5655	-0.0312	2.5081	-0.6343	-0.0874	-3.1198	
-2.4027	1.8320	-0.9690	0.7135	1.0560	0.7349	0.2614	
-0.0598	0.2705	0.4831	-1.0267	1.9281	-0.6504	-1.2773	
-2.4997	-0.7740	0.8562	-2.1614	-1.1692	-2.8045	0.4456	
-2.2833	-0.8347	-0.8968	-1.7699	-1.0679	-1.6927	-1.2422	
0.8948	-0.6419	-0.3985	-0.7658	0.9061	0.6613	-2.0070	
2.1141	-0.2214	0.5923	0.9273	1.3326	1.4063	-1.6233	
0.8029	-0.2736	-0.2919	-1.6149	4.7477	2.2088	1.3013	
-0.9459	0.2275	1.1472	-1.4942	-2.8577	-0.5690	-0.9332	
1.5703	-0.0496	1.8765	-0.6482	2.0143	1.6645	0.8582	-0.1472
-0.7951	1.3593	0.8056	-1.2884	-1.8784	1.4389	1.6502	
2.2537	-0.0248	-0.3302	0.2093	0.7825	0.8012	1.2170	
0.7537	1.5335	0.9172	-2.9717	0.6344	0.0185	-1.2831	
-0.2896	0.5039	0.2871	2.3257	0.1097	-0.7387	2.1051	
-0.1246	-0.8703	-0.5654	2.1411	-4.6610	-1.7761	1.4487	
-1.1202	-0.3724	-0.7511	0.3485	-0.1365	-1.0577	-2.1164	
-1.2842	-0.6000	-0.8760	-0.1215	-1.5482	1.0101	-1.3748	
-0.6424	-0.3483	0.6935	2.6202	2.5838	-0.7696	-1.0314	
0.0404	-1.0799	-0.8518	0.6817	1.7254	-0.6435	1.5352	
2.3093	-2.7393	-1.8759	0.9396	0.2019	-0.1160	0.3811	
-0.6783	-1.7623	-1.9056	-0.9479	-0.4731	0.4227	1.2703	
0.6327	-0.2987	1.3779	1.5440	1.2748	0.0593	1.5599	

Tableau III.9. Poids et biais du RN*P* optimal.

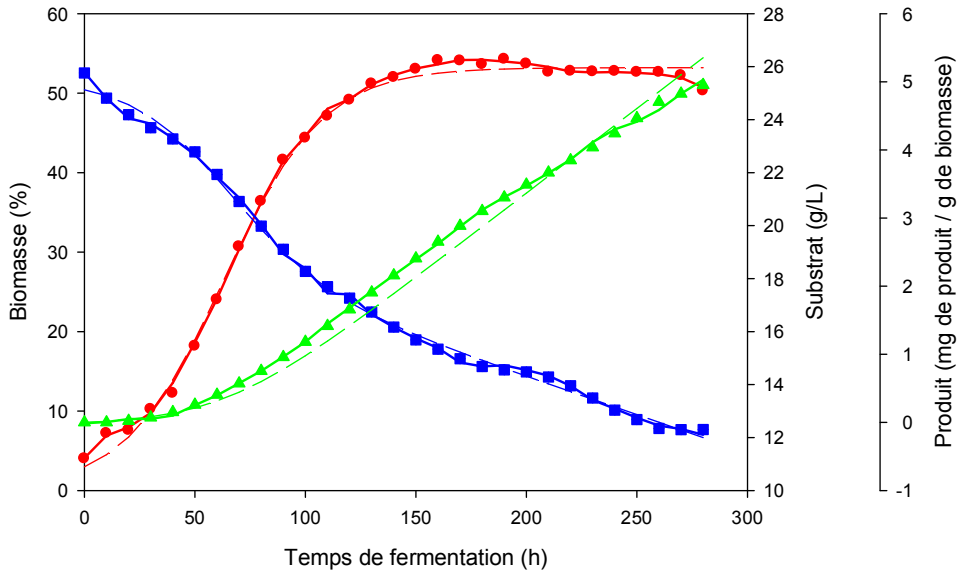
Connexions Couche d'entrée-Couche cachée						**Connexions Couche cachée-Couche de sortie**	
Poids					**Biais**	**Poids**	**Biais**
w_{j1}^{I}	w_{j2}^{I}	w_{j3}^{I}	w_{j4}^{I}	w_{j5}^{I}	b_{j}^{H}	w_{1j}^{H}	b_{1}^{0}
-2.9269	-1.4263	-7.2014	6.3928	1.8073	-3.9187	-1.6162	
1.1146	-1.2153	0.4311	-5.4334	-1.9241	2.9022	-6.4467	
0.3890	-2.3302	-6.0827	1.4968	2.1090	0.7316	-2.0319	
8.2538	-1.4353	1.3024	-6.0739	6.5412	-3.4397	-0.6263	
8.4439	-0.0844	-0.4511	-7.3537	5.1683	0.5437	-0.4524	
-0.7579	1.4007	0.4931	7.1490	4.8052	-6.4504	-8.0603	
-1.9866	-1.3324	-2.2629	2.9612	-1.5593	-0.9006	3.2565	
-4.1809	-0.9944	6.9592	-0.8699	-6.5638	6.4973	1.9482	
-6.7334	1.0292	3.5748	-0.7967	3.1997	2.3931	1.1469	
2.5570	-1.0137	-3.1749	4.1644	0.8190	-3.4140	4.6223	
0.4549	-2.2869	-1.9826	-8.0184	-7.3070	10.0426	-3.2575	
3.8836	-4.7567	-5.6697	2.3926	0.8384	1.5620	0.7641	
0.7223	-0.6695	0.4233	-4.7059	2.9251	-0.2904	3.8255	
-2.3481	0.6829	1.7264	-5.6177	-3.3177	5.3658	5.5468	-0.2192
2.2828	1.1771	-4.6640	-0.8588	4.7092	-4.5815	5.8573	
0.2189	0.6469	1.6344	-3.2380	-4.7119	3.4996	-3.8437	
-2.4529	0.2741	2.0833	0.3341	0.1276	0.3031	-6.1021	
1.2915	-1.5857	2.7867	6.1771	2.9904	-1.1936	4.9541	
1.4878	-1.5809	-0.5043	0.3089	-1.1925	0.7875	-5.0642	
-4.2033	1.7703	0.9559	3.3474	9.0597	-6.0420	1.9917	
4.0350	-0.5836	-2.2463	4.7719	8.4582	-8.7975	2.5084	
2.6220	-0.4797	-2.3259	-3.7764	6.7859	-3.8903	-2.1910	
1.8278	1.1449	-2.3150	0.3273	3.4368	-4.8187	-6.9692	
-1.6494	1.0795	-3.1726	-12.4793	-3.9320	2.5699	3.8432	
7.6263	-1.2788	-1.9833	-0.1160	-2.5358	0.0436	-0.8434	
2.1441	0.8696	1.7867	6.6175	-2.6741	-3.5789	1.6261	
2.7848	-1.5431	0.1194	-1.7920	-4.1354	2.8878	5.6397	

III.3.4. Comparaison avec les modèles cinétiques non structurés

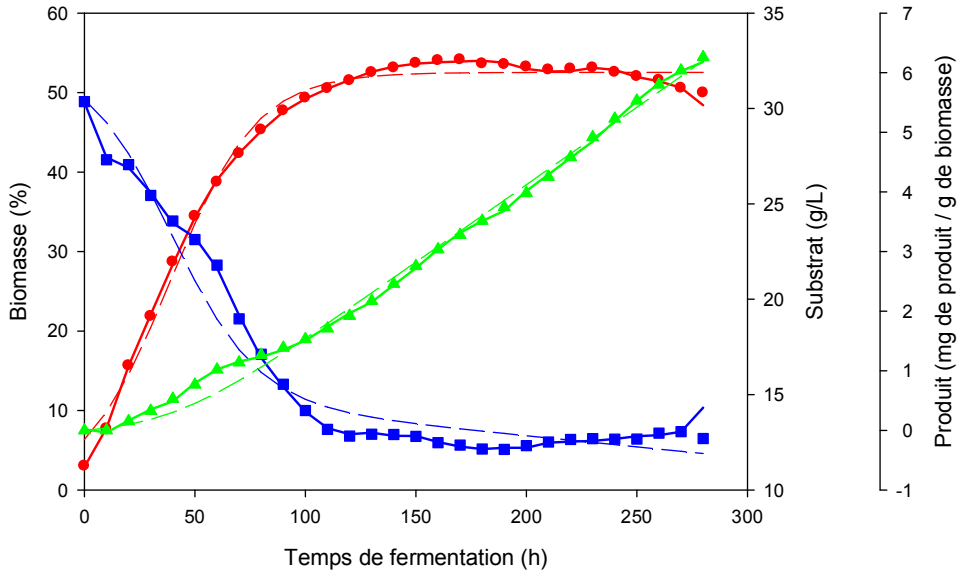
Afin d'établir les modèles neuronaux optimisés (RN*X*, RN*S* et RN*P*) en tant qu'alternative plausible aux modèles cinétiques non structurés (MC*X*, MC*S* et MC*P*), une comparaison entre les deux approches a été faite en termes de résultats de modélisation. Ces résultats sont illustrés sur la figure III.8 pour les lots 2, 4, 9 et 17. Les résultats des autres lots (lots 1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 18 et 19) sont représentés en Annexe. Ces résultats montrent que les modèles des RNFF développés surpassent les modèles cinétiques non structurés dans la prédiction des profils de concentration de la biomasse, du glucose et de la pleuromutiline ; il est à noté aussi que le phénomène de fermentation industrielle de pleuromutiline, avec sa phase d'accélération, sa phase exponentielle de croissance, sa phase de décélération et sa phase stationnaire, est bien présenté pour tous les lots. Les équations proposées pour décrire les taux spécifiques de croissance et de formation du produit sont appropriées pour quelques ensembles de données (lots) et peu convenables pour d'autres. Ceci peut être dû au changement du comportement de la souche *Pleurotus mutilus* d'un lot à un autre. Ceci exigera probablement une modification des équations des taux spécifiques d'un lot à un autre, ce qui rendra difficile leur incorporation dans une boucle de contrôle du processus.



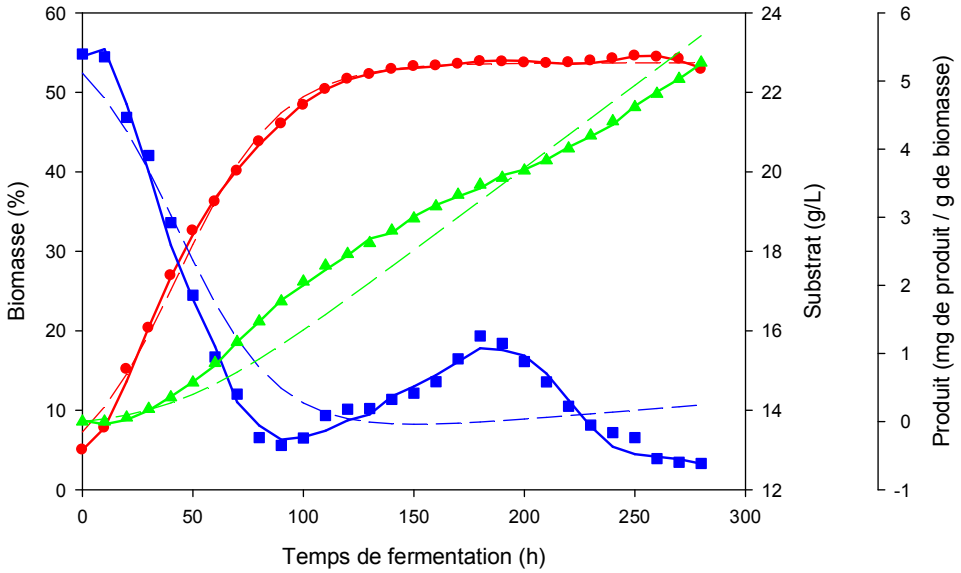
(a)


Biomasse (%)
Substrat (g/L)
Produit (mg de produit / g de biomasse)
Temps de fermentation (h)
0
50
100
150
200
250
300
0
10
20
30
40
50
60
10
12
14
16
18
20
22
24
26
28
-1
0
1
2
3
4
5
6

(b)


Biomasse (%)
Substrat (g/L)
Produit (mg de produit / g de biomasse)
Temps de fermentation (h)
0
50
100
150
200
250
300
0
10
20
30
40
50
60
10
15
20
25
30
35
-1
0
1
2
3
4
5
6
7

(c)



(d)

Figure III.8. Résultats de modélisation: **(a)** lot N°2, **(b)** lot N°4, **(c)** lot N°9, **(d)** lot N°17. (Les données expérimentales (symboles) pour la biomasse (●), glucose (■) et pleuromutiline (▲); Les résultats de la modélisation par RNFF sont représentés par des traits pleins et les résultats de la modélisation cinétique sont présentés par des lignes en pointillés)

Afin de visualiser encore plus les capacités prédictives des modèles neuronaux (RN) par rapport aux modèles cinétiques non structurés (MC), nous avons établi une deuxième comparaison en termes d'analyse statistique à travers le calcul de la *REQM* (Equation III.5) et l'erreur standard de prédiction (*ESP*) (Equation III.19).

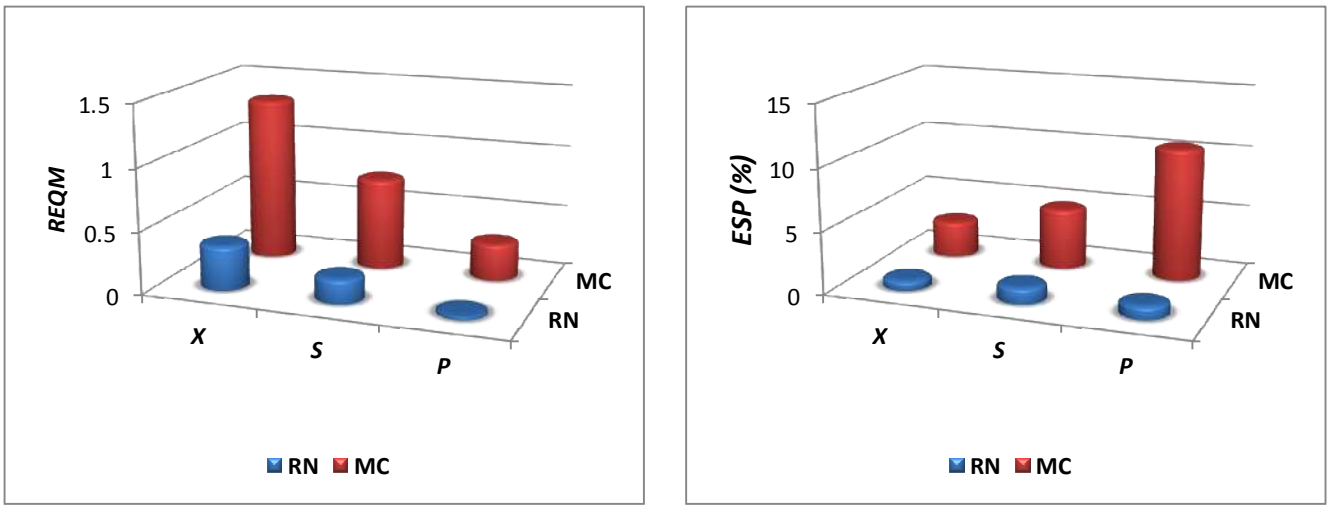
$$ESR(\%)=\frac{REQM}{Y_e}\times 100 \qquad \textbf{(III.19)}$$

$$Y_e=\frac{\sum_{i=1}^{n} Y_{i,\exp}}{n} \qquad \textbf{(III.20)}$$

Où $Y_{i,exp}$ est la valeur expérimentale ; Y_e est la moyenne des données expérimentales ; n est le nombre de points expérimentaux.

Les résultats de la comparaison entre les réseaux de neurones optimisés (RN) et les modèles cinétiques (MC), en termes de la *REQM* et l'*ESP*, sont présentés sur l'histogramme de la figure III.9 et dans le tableau III.10. Ces résultats indiquent que les déviations des prédictions obtenues par les

modèles neuronaux optimisées sont nettement plus faibles que celles obtenues par les modèles cinétiques non structurés.



(a) **(b)**

Figure III.9. Comparaison entre les réseaux de neurones optimisés (RN) et les modèles cinétiques (MC) pour la totalité de la base de données : **(a)** en termes de la *REQM*, **(b)** en termes de l'*ESP*.

Tableau III.10. Comparaison, en termes de la *REQM* et l'*ESP*, entre les réseaux de neurones optimisés (RN) et les modèles cinétiques (MC).

Lots	*REQM*						*ESP (%)*					
	Biomasse		Substrat		Produit		Biomasse		Substrat		Produit	
	RN*X*	MC*X*	RN*S*	MC*S*	RN*P*	MC*P*	RN*X*	MC*X*	RN*S*	MC*S*	RN*P*	MC*P*
Lot 1	0.2510	1.1132	0.2072	0.5854	0.0319	0.1139	0.5050	2.2403	1.4788	4.1773	1.4139	5.0502
Lot 2	0.3232	1.4080	0.0930	0.8741	0.0294	0.2702	0.6159	2.6833	0.6759	6.3552	1.1927	10.9449
Lot 3	0.3632	1.3390	0.1055	1.0631	0.0237	0.0975	0.7245	2.6709	0.6655	6.7055	1.0707	4.4109
Lot 4	0.4174	1.0602	0.1099	0.2195	0.0372	0.1846	1.0283	2.6120	0.6291	1.2563	1.6825	8.3508
Lot 5	0.6852	1.1420	0.3302	1.0695	0.0198	0.2889	1.6286	2.7142	2.0919	6.7762	0.6090	8.8698
Lot 6	0.4548	1.6057	0.2575	1.3489	0.0277	0.2941	0.9917	3.5016	1.7586	9.2119	0.8931	9.4934
Lot 7	0.2286	2.3259	0.1612	0.3096	0.0150	0.1322	0.5251	5.3417	1.0358	1.9893	0.7041	6.1884
Lot 8	0.3808	1.7642	0.1918	0.5418	0.0221	0.0751	0.8976	4.1588	1.2835	3.6253	0.9592	3.2581
Lot 9	0.3430	1.3522	0.3188	1.0505	0.0344	0.1480	0.7768	3.0628	1.9547	6.4409	1.2571	5.4125
Lot 10	0.1876	1.6143	0.1582	0.3973	0.0226	0.4178	0.4235	3.6437	1.0438	2.6213	0.6653	12.3046
Lot 11	0.1188	1.5770	0.1260	0.5196	0.0450	0.5531	0.2682	3.5597	0.9598	3.9577	1.1308	13.8950
Lot 12	0.3251	0.6266	0.1575	0.2756	0.0192	0.2075	0.7321	1.4109	1.1850	2.0741	0.6764	7.3027
Lot 13	0.2098	1.2592	0.1592	0.9099	0.0197	0.4401	0.4563	2.7382	1.1996	6.8565	0.7186	16.0178
Lot 14	0.2385	1.0587	0.1853	0.4143	0.0247	0.3021	0.5271	2.3398	1.1288	2.5241	0.7935	9.6953
Lot 15	0.2266	0.8469	0.1042	0.3867	0.0278	0.2846	0.5011	1.8723	0.6151	2.2835	0.9965	10.2021
Lot 16	0.4578	1.0504	0.2476	0.6278	0.0263	0.3760	1.0294	2.3616	1.5174	3.8466	0.7867	11.2582
Lot 17	0.3390	0.9668	0.2431	1.0985	0.0319	0.3902	0.7663	2.1859	1.5757	7.1189	1.2318	15.0731
Lot 18	0.3049	1.2522	0.1752	0.5615	0.0202	0.1689	0.6765	2.7780	1.1803	3.7825	0.8321	6.9608
Lot 19	0.4497	0.8417	0.2292	0.4131	0.0329	0.3380	1.0395	1.9458	1.6828	3.0336	1.1479	11.7813
Ensemble des lots	**0.3548**	**1.3297**	**0.1993**	**0.7427**	**0.0279**	**0.2965**	**0.7862**	**2.9465**	**1.3203**	**4.9213**	**1.0036**	**10.6760**

III.3.5. Performances d'interpolation

Pour vérifier encore plus la capacité de généralisation de ces trois modèles neuronaux précédemment développés et optimisés, deux types de bases de données d'interpolation ont été utilisés :

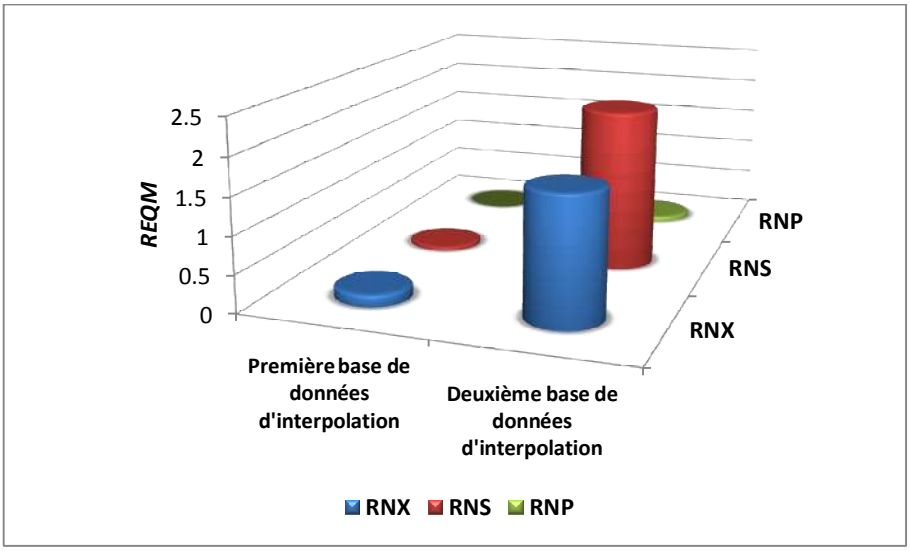
1. La première base de données contient un ensemble de points intermédiaires entre les points expérimentaux du lot numéro 6 (le lot numéro 6 a été choisi au hasard) ;
2. La deuxième base de données d'interpolation contient un ensemble de données complet jamais exploité pendant la phase d'apprentissage et de test: $X_0 = 3$ % et $S_0 = 23.5$g/L.

Les résultats de performances d'interpolation en termes de la *REQM* et en termes du coefficient de corrélation (*R*) sont résumés dans le tableau III.11 et présentés sur la figure III.10. La qualité de l'ajustement de l'interpolation du premier ensemble de données est représentée sur la figure III.11. Un excellent ajustement aux valeurs expérimentales de la biomasse, du sucre et du produit est noté. Les résultats de la simulation de la seconde base de données d'interpolation sont illustrés sur la figure III.12. On peut observer que le RN*X* et le RN*P* s'ajuste très bien aux points expérimentaux de la biomasse et du produit respectivement ; Cependant, les résultats du RN*S* s'ajustent moins bien aux concentrations expérimentales de glucose.

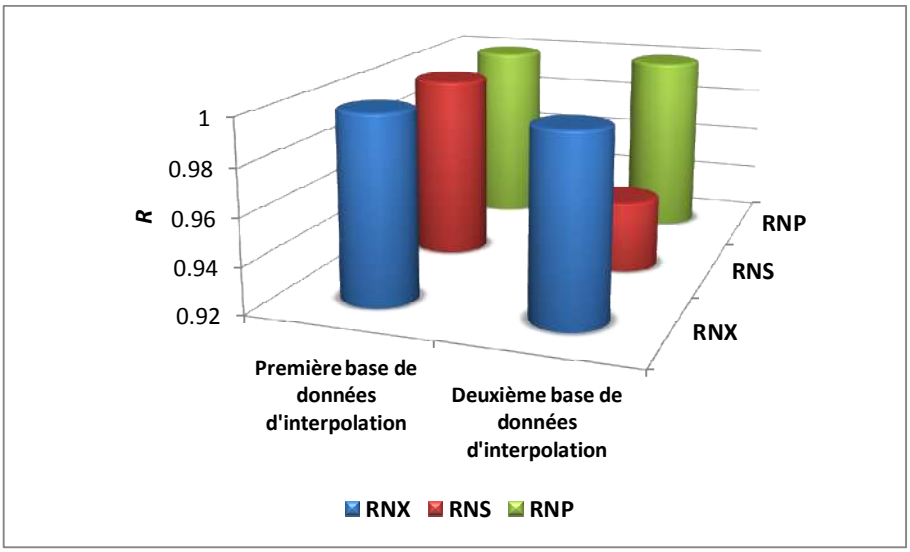
Bien que la précision des modèles neuronaux fût modérément satisfaisante, en particulier pour le RN*S*, plus de données historiques couvrant l'ensemble du champ d'intérêt doivent être ajoutées à la phase d'apprentissage afin d'augmenter de plus en plus la qualité prédictive des modèles neuronaux.

Tableau III.11. Résultats d'interpolation, en termes de la *REQM* et du coefficient de corrélation (*R*).

	1ère base de données d'interpolation		**2ème base de données d'interpolation**	
	REQM	***R***	***REQM***	***R***
RN*X*	0.1565	0.9997	1.7100	0.9988
RN*S*	0.0531	0.9991	2.1810	0.9508
RN*P*	0.0006	0.9999	0.0797	0.9995



(a)



(b)

Figure III.10. Représentation des résultats de l'interpolation : **(a)** en termes de la *REQM*, **(b)** en termes du coefficient de corrélation (*R*).

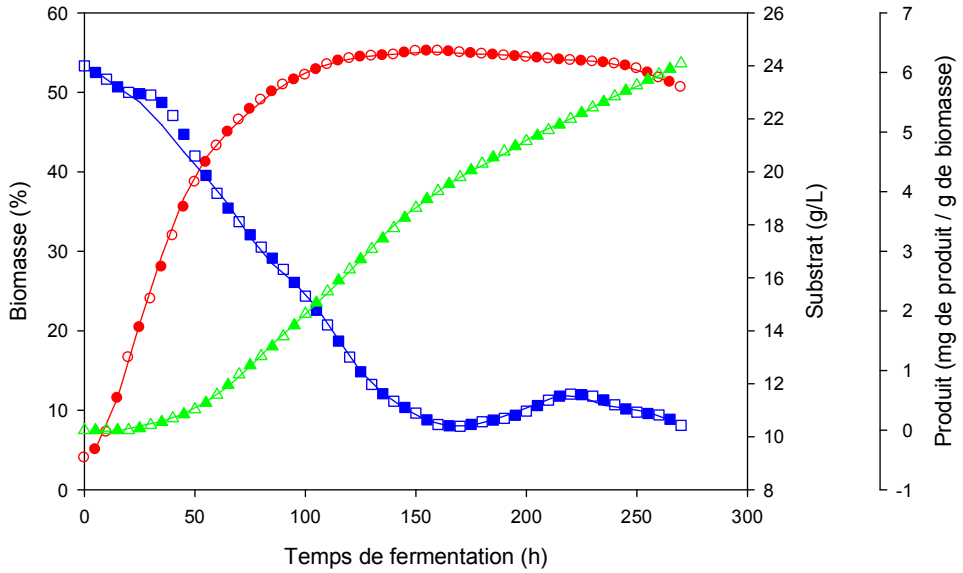


Figure III.11. Résultats de simulation de la première base de données d'interpolation.
(Les données interpolées expérimentales (symboles) pour la biomasse (○), glucose (□) et pleuromutiline (△); Les résultats de la modélisation par RNFF sont représentés par des traits pleins)

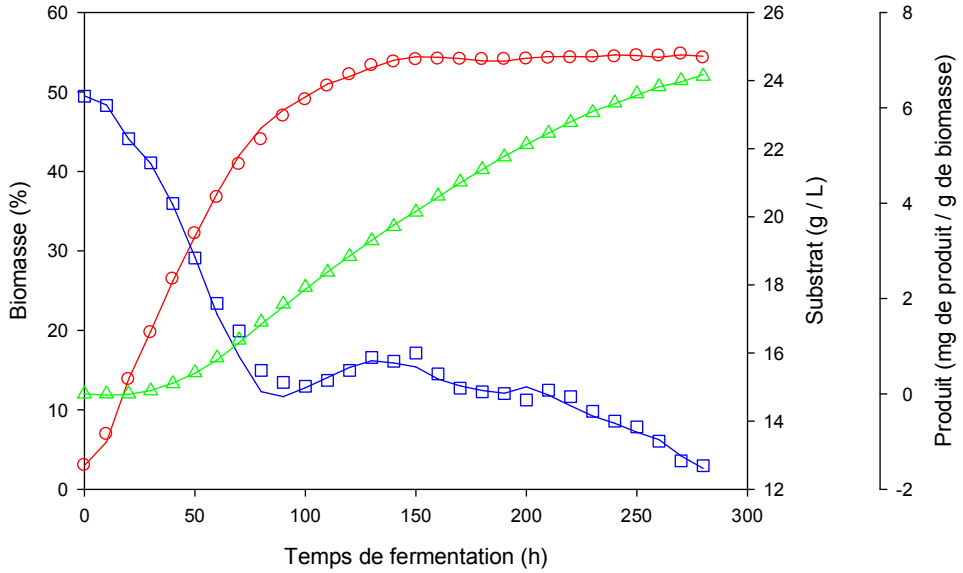


Figure III.12. Résultats de simulation de la seconde base interpolation: X_0=3% et S_0=23.5g/L.
(Les données expérimentales Interpolées (symboles) pour la biomasse (○), glucose (□) et pleuromutiline (△); Les résultats de modélisation par RNFF sont représentés par des traits pleins)

III.3.6. Analyse de sensibilité

Pour voir la contribution ainsi que le profil de variation de chaque variable d'entrée (*t, pH, IR, X_0 et S_0*) sur la sortie de chaque réseau (RN*X*, RN*S* et RN*P*), On a procédé à une analyse de sensibilité par la méthode de "Poids" et la méthode de "Profil".

III.3.6.1. Méthode de "Poids"

Cette méthode a été appliquée respectivement aux RN*X*, RN*S* et RN*P* en utilisant leurs poids respectifs illustrés dans les tableaux III.7, III.8 et III.9. Les contributions des variables d'entrée obtenues par la méthode de "poids" pour chaque RN sont présentées sur la figure III.13.

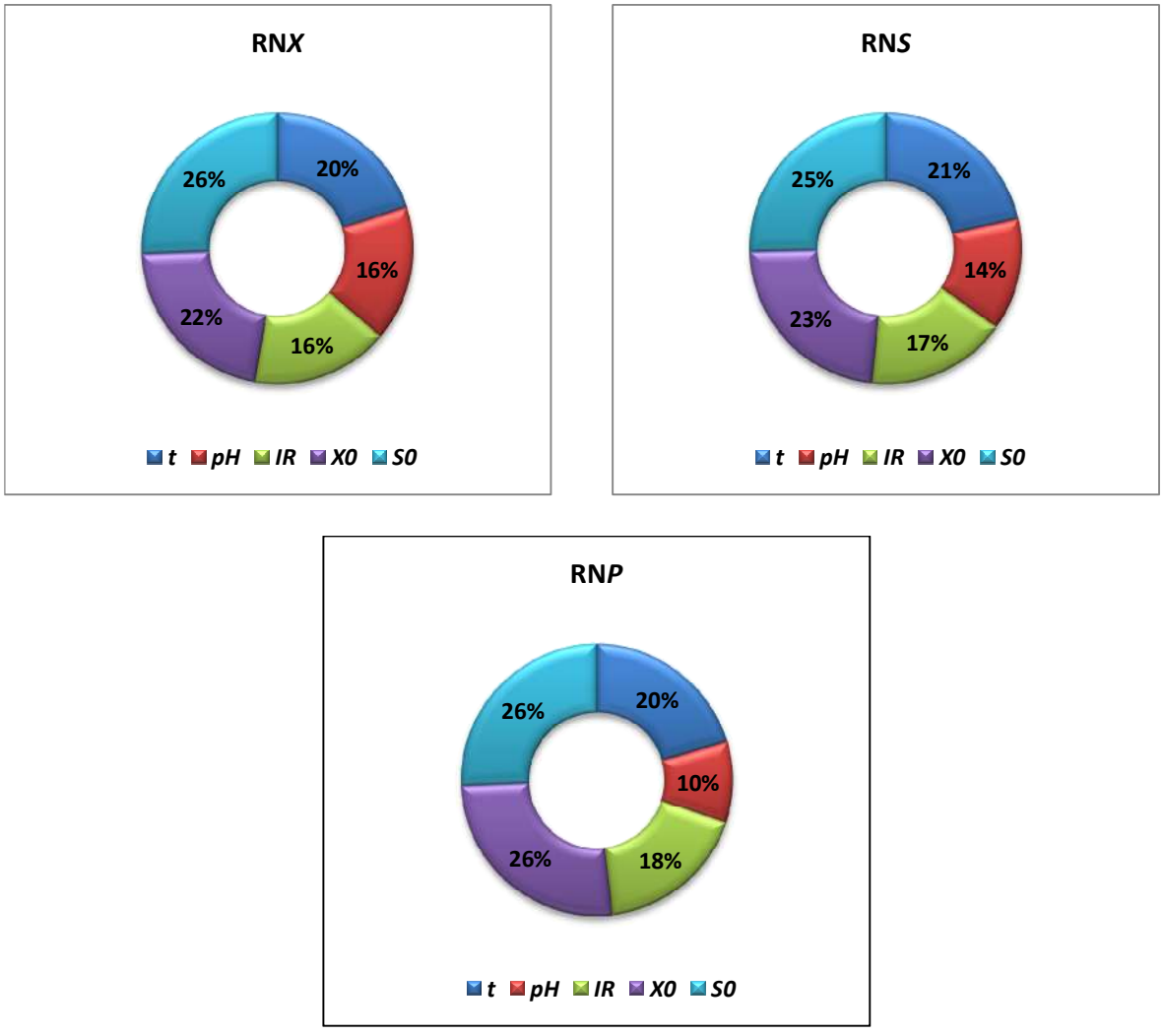


Figure III.13. Présentation des % de l'importance relative (*Ir*) de chaque variable d'entrée sur la biomasse (RN*X*), sur la consommation du sucre (RN*S*) et sur le produit (RN*P*).

D'après la figure III.13, on remarque que la concentration initiale du substrat, la concentration initiale d'inoculum et le temps de fermentation exercent les contributions les plus importantes et ceux pour chaque RN. Pour le RN*X*, les contributions du *pH* et de l'*IR* ne sont pas significativement différentes (ils ont la même contribution avec une *Ir* de 16%), alors que pour les deux autres RN (RN*S* et RN*P*), l'*IR* exerce une contribution plus importante que le *pH*.

III.3.6.2. Méthode de "Profil"

Nous avons appliqué cette méthode pour déterminer le profil de contribution de chaque variable d'entrées simulées (*t, pH, IR,* X_0 et S_0) sur la biomasse prédite par RN*X* (Figure III.14), la consommation du sucre calculée par RN*S* (Figure III.15) et la formation du produit prédite par RN*P* (Figure III.16).

D'après la figure III.14 présentant les graphes du profil de contribution des variables d'entrées (*t, pH, IR,* X_0 et S_0) sur la biomasse prédite par le RN*X*, on constate que :

- La biomasse augmente avec l'augmentation du temps de fermentation (*t*) jusqu'à la valeur de 180 h ; au delà de cette valeur la biomasse est presque constante, indiquant ainsi la phase stationnaire de croissance (Figure III.14 (a));
- L'augmentation du *pH* au cours de la fermentation est peu influente sur la biomasse, cependant, pour les valeurs de *pH* dépassant les 7.2, une diminution légère peu être observée (Figure III.14 (b));
- Les faibles valeurs de l'*IR* sont peu influentes sur la biomasse tandis que les grandes valeurs favorisent la croissance cellulaire (Figure III.14 (c));
- La concentration initiale de l'inoculum est peu influente sur la biomasse prédite par le RN*X* (Figure III.14 (d));
- La croissance cellulaire décroit avec l'augmentation de la concentration initiale du substrat S_0 (Figure III.14 (e)), indiquant ainsi une inhibition de la biomasse par S_0;

Pour la figure III.15 présentant les graphes du profil de contribution des variables d'entrées (*t, pH, IR,* X_0 et S_0) sur la concentration du substrat prédite par le RN*S*, on remarque que :

- La concentration du substrat décroit avec l'augmentation du temps de fermentation (*t*) (Figure III.15 (a)) et croit avec l'augmentation de l'*IR* (Figure III.15 (c));
- Pour les faibles valeurs du *pH* ($pH < 6.6$), la concentration du substrat augmente montrant ainsi la non consommation du substrat par la souche *Pleurotus mutilus* ; alors

que, pour les *pH* élevés ($pH > 6.6$), la concentration du substrat diminue indiquant la consommation du substrat par *Pleurotus mutilus* (Figure III.15 (b));

- La concentration initiale de l'inoculum est peu influente sur le substrat prédit par le RN*S* (Figure III.15 (d));
- Pour les valeurs de S_0 inférieures à 24 g/l, la concentration du substrat prédite augmente, montrant ainsi une accumulation du substrat ; alors que pour les S_0 dépassant les 24 g/l, la concentration du substrat diminue illustrant donc la consommation du substrat par la souche (Figure III.15 (e)) ;

A partir des graphes du profil de contribution des variables d'entrées (*t, pH, IR,* X_0 et S_0) sur la production de la pleuromutiline prédite par le RN*P* et qui sont illustrés sur la figure III.16, on peut tirer les constatations suivantes :

- La production de pleuromutiline débute à partir de 50h, puis augmente linéairement avec l'augmentation du temps de fermentation (Figure III.16 (a)) ;
- La production de pleuromutiline décroit avec l'augmentation du *pH* (Figure III.16 (b)) ;
- La concentration de pleuromutiline diminue avec l'augmentation de l'*IR* jusqu'à ce quelle s'annule pour les valeurs dépassant les 5 (Figure III.16 (c)) ;
- La forme de variation de la pleuromutiline prédite par le RN*P* en fonction de X_0 est presque sinusoïdale, montrant ainsi la forte contribution de cette variable sur la production de pleuromutiline, on remarque aussi que les valeurs de X_0 comprise entre 5.5 et 6.5 % sont les plus favorables pour avoir une production maximale (Figure III.16 (d)).
- Plus S_0 augmente plus la production augmente jusqu'à la valeur de 27g/l, au delà de cette valeur la production diminue (Figure III.16 (e)).

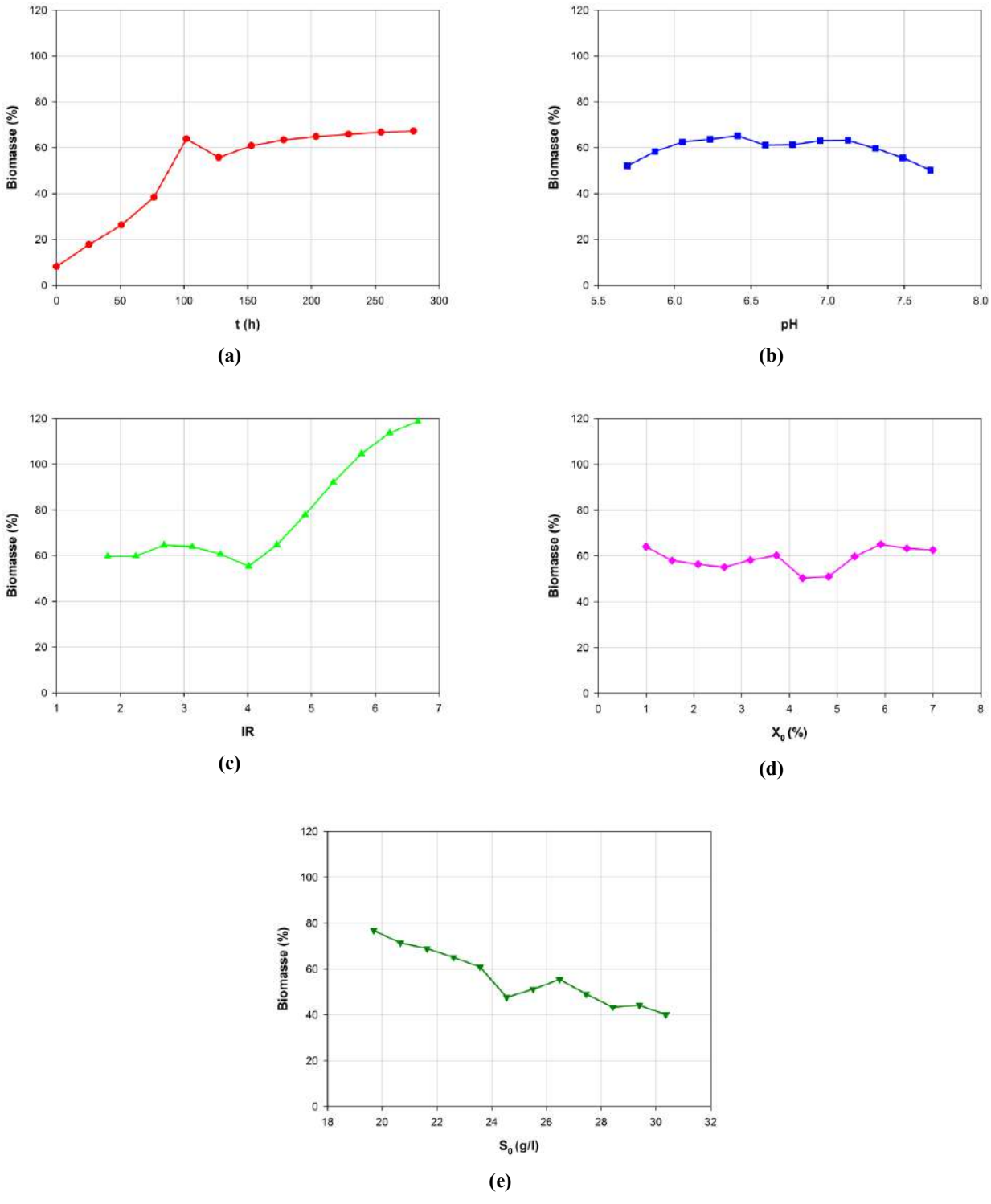


Figure III.14. Graphes du profil de contributions des variables simulées (*t, pH, IR, X_0* et *S_0*) sur la biomasse prédite par RN*X* : **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de *X_0*, **(e)** effet de *S_0*.

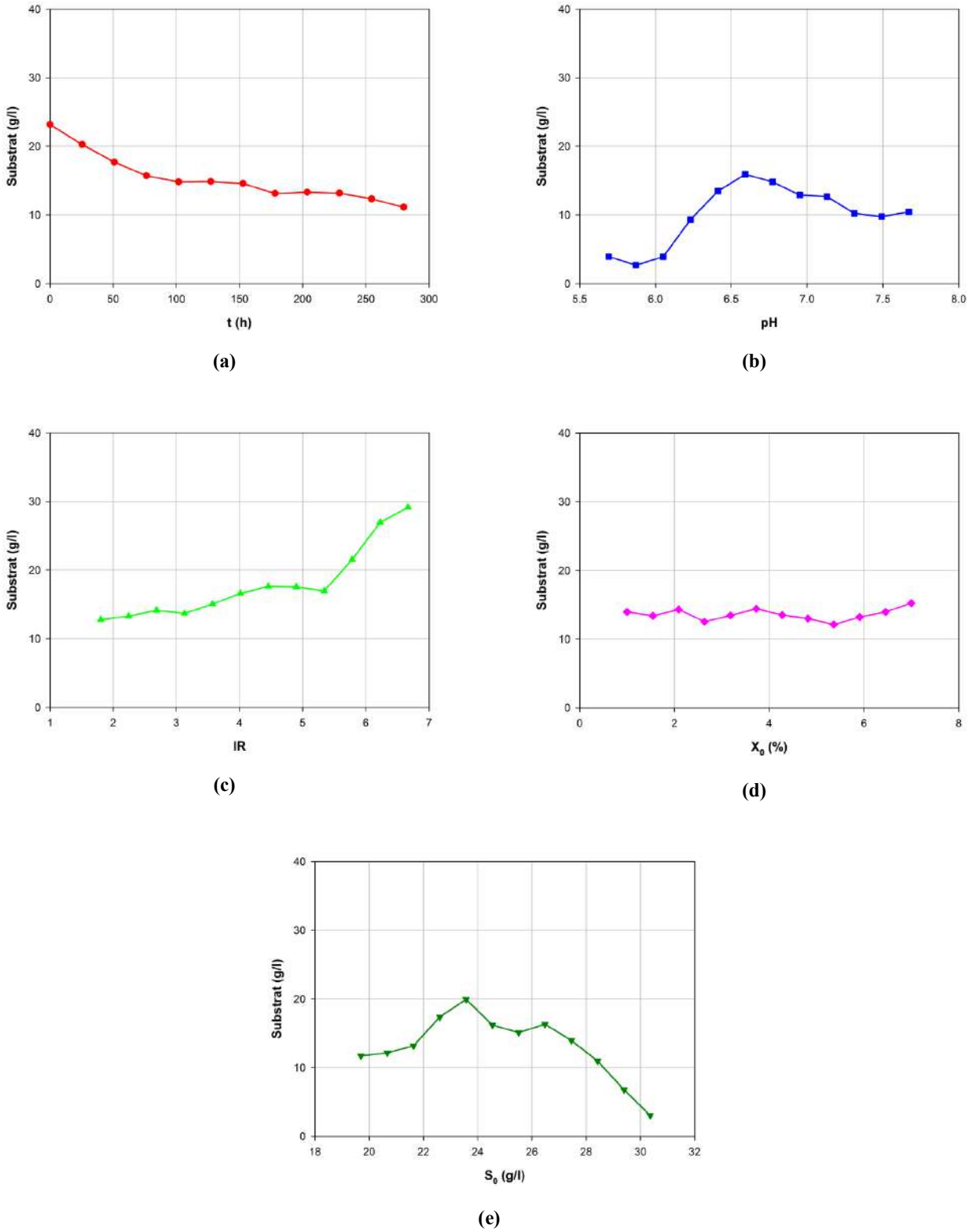


Figure III.15. Graphes du profil de contribution des variables simulées (*t, pH, IR, X_0* et *S_0*) sur le substrat prédit par RN*S*: **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de *X_0*, **(e)** effet de *S_0*.

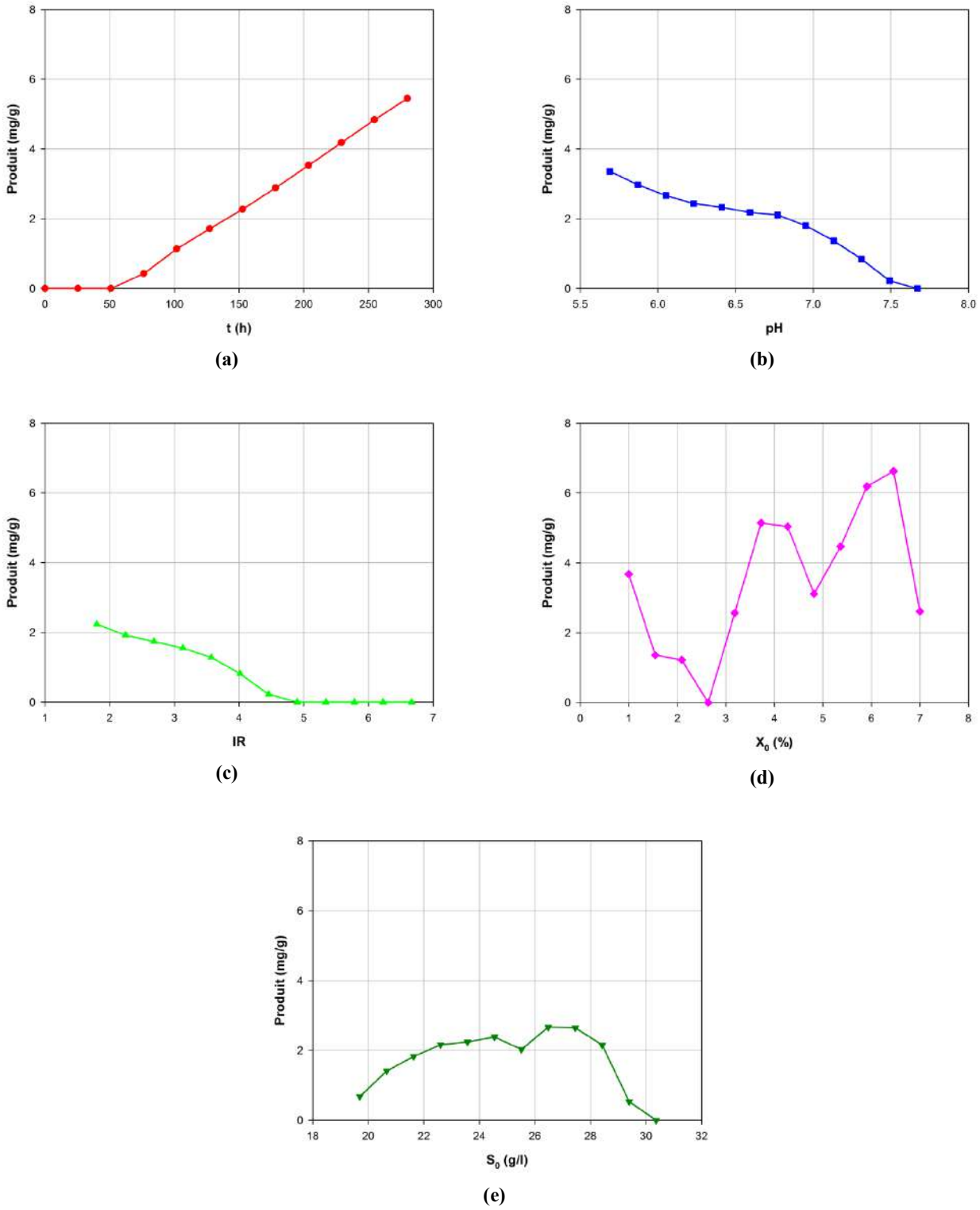


Figure III.16. Graphes du profil de contribution des variables simulées (*t, pH, IR, X_0* et *S_0*) sur le produit prédit par RN*P*: **(a)** effet de *t*, **(b)** effet du *pH*, **(c)** effet de l'*IR*, **(d)** effet de *X_0*, **(e)** effet de *S_0*.

Les contributions relatives de chaque variable d'entrée ont été exprimées par les valeurs de l'intervalle (maximum-minimum) de leurs contributions (Figure III.17). On constate que, pour chaque RN les % de contributions relatives de chaque variable dépasse les 5%, montrant ainsi l'importance des cinq variables sur les sorties des RN.

Cette analyse de sensibilité par les deux méthodes utilisées montre à posteriori l'importance des cinq paramètres choisis pour la modélisation de la cinétique de fermentation de la pleuromutiline et constitue par conséquent une justification du choix postulé antérieurement.

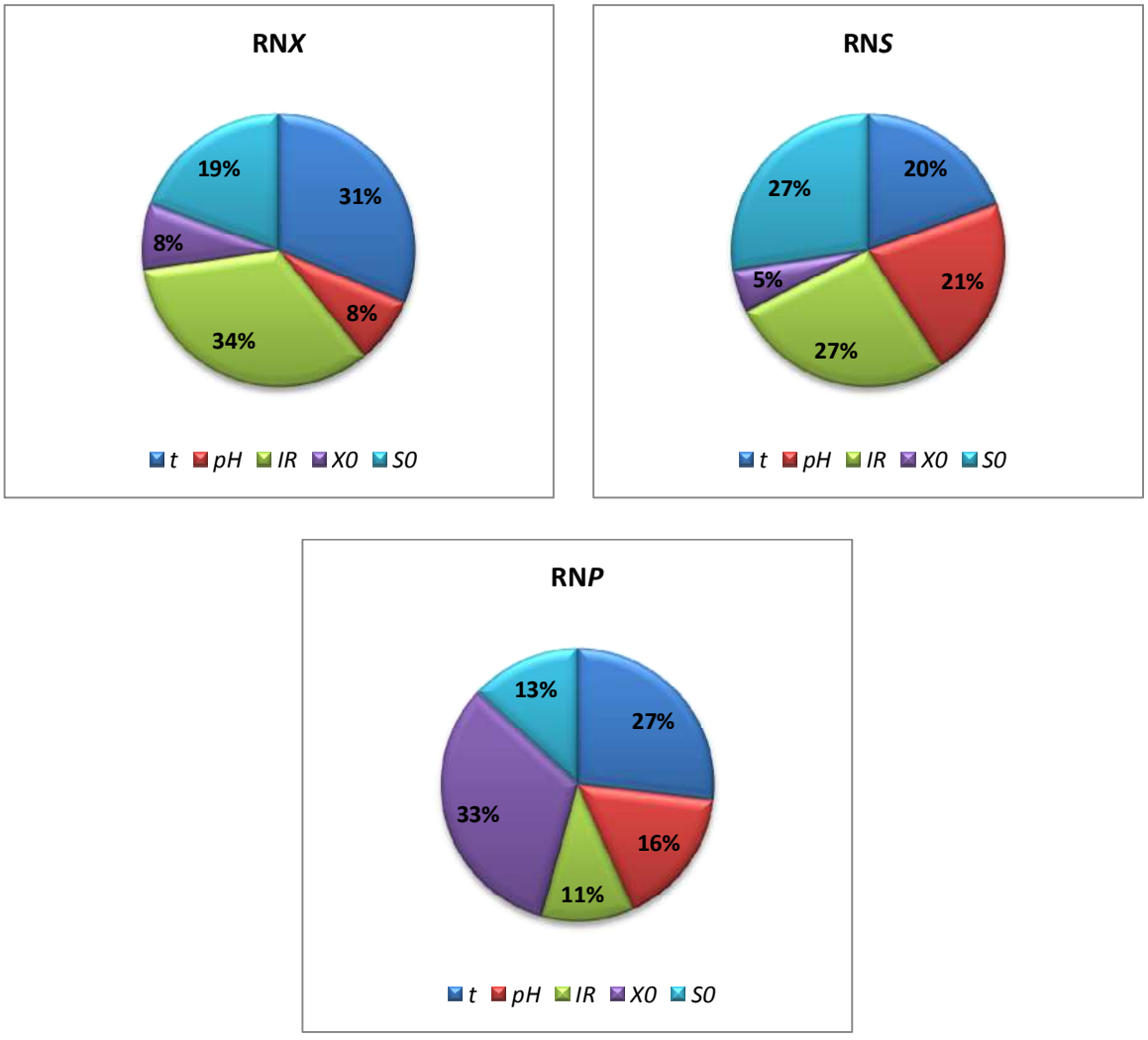


Figure III.17. Présentation des % de contribution relative de chaque variable d'entrée sur la biomasse (RN*X*), sur la consommation du sucre (RN*S*) et sur le produit (RN*P*).

III.4. Conclusion

Ce présent chapitre a donné une raison de plus pour l'utilisation généralisée des réseaux de neurones artificiels de type feedforward pour la modélisation de la fermentation d'antibiotiques. On a développé trois modèles de RNFF capables d'évaluer trois variables principales du bioprocédé de fermentation fed-batch de pleuromutiline à savoir la biomasse, la consommation du substrat et la formation du produit. Les résultats ont montré une bonne concordance entre les valeurs prédites et expérimentales pour chaque modèle neuronal (les *REQM* pour la base test étaient respectivement de 0,4624% - 0,1234 g/l et 0,0016 mg/g pour RN*X*, RN*S* et RN*P*).

Par comparaison aux modèles cinétiques non structurés, les modèles neuronaux optimisés ont pu prédire avec suffisamment de précision les profils de concentrations pour tous les ensembles de données. De plus, les formules mathématiques obtenues à partir de ces modèles optimisés englobent non seulement les éléments importants du processus, mais ils sont aussi moins complexes, faisant leur intégration dans une boucle de contrôle industriel du processus plus facile. Cependant, la modélisation par réseaux de neurones artificiels ne peut pas remplacer la modélisation cinétique lorsqu'on tente de comprendre le phénomène de fermentation.

L'analyse de sensibilité par les deux méthodes utilisées (méthode de "Poids" et méthode de "Profil") a montré à posteriori l'importance des cinq paramètres (*t, pH, IR,* X_0 et S_0) choisis pour la modélisation de la cinétique de fermentation de pleuromutiline.

Chapitre IV

Optimisation hybride des conditions de culture de Pleurotus mutilus

IV.1. Introduction

Dans ce chapitre, nous présentons une méthodologie hybride, basée sur un plan composite centré (PCC), un réseau de neurones (RN) et un algorithme d'optimisation par essaim de particules (OEP). Cette technique est utilisée pour optimiser les conditions de culture de *Pleurotus mutilus* (vitesse d'agitation, température et *pH* initial) et ceux dans le but d'obtenir une production maximale en pleuromutiline. Le PCC est utilisé pour concevoir l'expérience, le RN formé avec l'algorithme d'OEP (RNOEP) sert à développer une fonction mathématique reliant le rendement en pleuromutiline avec les trois facteurs. Ensuite, un algorithme générique d'OEP est appliqué pour exécuter la tâche d'optimisation de la fonction mathématique. Les résultats de l'optimisation des conditions de culture de *Pleurotus mutilus* ont été publiés dans le "Biotechnology and Bioprocess Engineering" (Khaouane *et al.* 2012a) et présenté à la 3ème Journée Scientifique sur le Génie des Procédés (JSGP3-2012) qui s'est tenue à l'Université Saad Dahlab de Blida, le 19 novembre 2012 (Khaouane *et al.* 2012b).

Le chapitre est organisé en cinq parties :

- La première partie aborde les matériels et les différentes techniques utilisées lors du suivi des expériences de fermentation ;
- Dans un deuxième temps, les différentes étapes utilisées dans le développement de la méthodologie hybride sont présentées ;
- Les différents logiciels utilisés, pour la modélisation et l'optimisation, sont ensuite présentés ;
- La quatrième partie est consacrée à la discussion des résultats significatifs obtenus ;
- Dans la cinquième partie, une comparaison de la méthodologie hybride proposée avec d'autres méthodologies d'optimisation à base des RN est illustrée.

IV.2. Matériels et méthodes

IV.2.1. Souche

La souche *Pleurotus mutilus* a été fournie par le Laboratoire de Microbiologie Industrielle de la filiale ANTIBIOTICAL du groupe industriel SAIDAL. La souche était conservée sur slants au réfrigérateur à +4 °C et repiquée à intervalles réguliers (1 mois) sur un milieu gélosé (Conseil de l'Europe 2001).

IV.2.2. Les milieux de culture

Deux milieux de culture ont été utilisés selon les objectifs poursuivis. Dans tous les cas le *pH* est ajusté avant la stérilisation avec la soude NaOH à 20 % et le milieu de culture est stérilisé par autoclavage à 121ºC pendant 30 minutes pour le milieu végétatif et pendant 20 minutes pour le milieu de fermentation (Conseil de l'Europe 2001).

IV.2.2.1. Milieu liquide végétatif

Ce milieu de germination de composition décrite dans le tableau IV.1 est ajusté à un *pH* de 6.0 (Conseil de l'Europe 2001).

Tableau IV.1. Composition du milieu liquide végétatif (Conseil de l'Europe 2001).

Constituant	**Composition**
Dextrose monohydraté ($C_6H_{12}O_6.H_2O$)	50 g
Liqueur de maïs macéré	37.5 g
Phosphate de potassium monobasique (KH_2PO_4)	1 g
Sulfate de magnésium heptahydraté ($MgSO_4.7H_2O$)	0.5 g
Nitrate de calcium tetrahydraté ($Ca(NO_3)_2.4H_2O$)	0.72 g
Chlorure de sodium (NaCl)	0.1 g
Sulfate ferrique heptahydraté ($FeSO_4.7H_2O$)	0.05 g

IV.2.2.2. Milieu liquide de fermentation

Ce milieu de fermentation de composition décrite dans le tableau IV.2 est ajusté à un *pH* de 7 (Conseil de l'Europe 2001).

Tableau IV.2. Composition du milieu liquide de fermentation (Conseil de l'Europe 2001).

Constituant	**Composition**
Dextrose monohydraté ($C_6H_{12}O_6.H_2O$)	60 g
Amidon	30 g
Liqueur de maïs macéré	45 g
Huile de soja	2 cm^3
Sulfate de magnésium heptahydraté ($MgSO_4.7H_2O$)	0.38 g

IV.2.3. Les conditions de culture

IV.2.3.1. Germination

Pour la phase de germination, des fioles de capacité de 500 ml et contenant chacune 70 ml du milieu liquide végétatif sont inoculées avec 4 ml de suspension de spores prélevées à partir des slants frais ou de conservation (faire un scraping de toute la surface blanche développée à l'aide d'une pipette stérile avec 8 ml d'eau physiologique stérile). Les fioles sont ensuite incubées à 27 °C pendant 72 h dans un agitateur rotatif à 240 tr/min. La forme végétative est considérée mature si elle répond aux critères suivants (Conseil de l'Europe 2001) :

- *pH* : entre 6,2 et 6,5 ;
- Biomasse : entre 20 et 35 % ;
- Observation à l'œil nu : culture légèrement jaunâtre ;
- Observation microscopique : le mycélium en pellettes claires avec hyphes siphonnés et ramifiées.

IV.2.3.2. Culture en batch

Pour la culture en batch, des fioles de capacité de 500 ml et contenant chacune 70 ml du milieu de fermentation sont inoculées avec 7 ml de bouillon prélevé à partir des fioles de germination bien développées. Les fioles sont ensuite incubées pendant neuf jours dans un agitateur rotatif (Conseil de l'Europe 2001). La vitesse d'agitation, la température d'incubation et le *pH* initial sont variées au cours de ce stade de fermentation selon un PCC.

IV.2.4. Méthodes analytique

IV.2.4.1. Analyse de la biomasse

La détermination du % de sédimentation dans l'échantillon du bouillon est faite par centrifugation. Pour cela des tubes coniques à la base et gradués jusqu'à 10 ml conçus pour la centrifugation (Figure IV.1) sont remplies avec 10 ml du bouillon puis centrifugés à 3000 tr/min pendant 10 min. Le pourcentage de sédimentation est calculé par la formule suivante (Conseil de l'Europe 2001) :

$$X(\%) = \frac{10(L_1 + L_2)}{2} L_{\max} \qquad \textbf{(IV.1)}$$

Ou L_1 et L_2 représentent les niveaux du culot et L_{max} le niveau du surnageant (Figure IV.2).

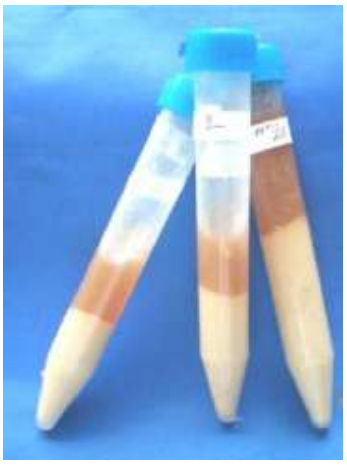


Figure IV.1. Tubes coniques pour sédimentation.

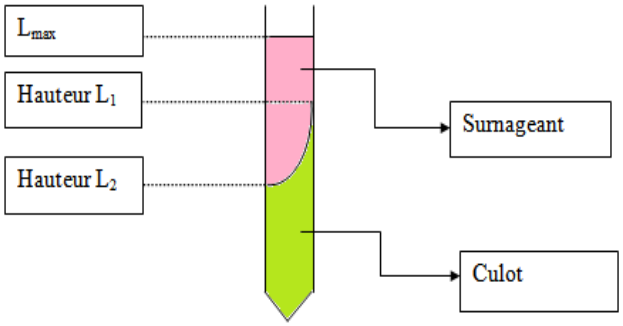


Figure IV.2. Représentation des hauteurs L_1, L_2 et L_{max}.

IV.2.4.2. Analyse du produit

Pour l'extraction de pleuromutiline, 20 ml de méthanol sont ajoutés à 10 g du bouillon de fermentation dans un ballon de 100 ml et mis sous agitation pendant 10 min. Après filtration avec un papier Wattman N° 42, les filtrats sont analysés par chromatographie en phase liquide à haute performance (HPLC) en utilisant une colonne en phase inverse C-18 à un débit de 1 ml/min avec une détection à 212 nm. La phase mobile est composée de méthanol (10 %), d'acétonitrile (40 %) et d'une solution de tampon phosphate (50 %) (Conseil de l'Europe 2001).

IV.3. Description de la méthodologie hybride

IV.3.1. Plan composite centré (PCC)

Une étude faite par Benkortbi (Benkortbi 2008) a révélé que la vitesse d'agitation, la température et le *pH* initial dans les conditions de culture de *Pleurotus mutilus* sont des variables

significatives pour la production de pleuromutiline. La température est un facteur qui affecte profondément la croissance des micro-organismes. Le *pH* est un indicateur de l'acidité ou de la basicité du bouillon de fermentation, il fournit des informations sur l'évolution favorable ou défavorable de la culture. La vitesse d'agitation est un facteur déterminant dans la fermentation où elle intervient à trois niveaux (Castro-Martinez 2007) :

i. Elle augmente les échanges entre les micro-organismes et les constituants du milieu de culture ;
ii. Elle augmente la concentration de l'oxygène dissout dans le milieu ;
iii. Elle entraine les effets de cisaillements sur les micro-organismes.

Le domaine expérimental et les niveaux des trois facteurs indépendants sont présentés dans le tableau IV.3. La variable dépendante est le rendement en pleuromutiline. Un PCC, regroupant 20 expériences au total, a été généré par le logiciel statistique Modde 5,0 (Umetrics). Le PCC comporte trois parties (Tableau IV.4) :

i. Un plan factoriel complet à deux niveaux et trois facteurs (2^3=8 expériences) ;
ii. Six points axiaux situés à une distance égale à deux du centre du domaine d'étude ;
iii. Six points centraux.

Tableau IV.3. Domaine expérimental et niveaux de chaque facteur.

Facteur	**Symbole**	**Intervalle et niveaux**				
		-2	**-1**	**0**	**1**	**2**
Vitesse d'agitation (tr/min)	X_1	200	220	240	260	280
Température (^{0}C)	X_2	25	26	27	28	29
pH	X_3	4	5	6	7	8

Tableau IV.4. Matrice d'expérimentation et rendement expérimental.

Essai	X_1	X_2	X_3	Rendement Expérimental[a] (µg/g)
1	220 (-1)	26 (-1)	5 (-1)	3658 ± 210
2	260 (+1)	26 (-1)	5 (-1)	5344 ± 151
3	220 (-1)	28 (+1)	5 (-1)	3078 ± 150
4	260 (+1)	28 (+1)	5 (-1)	5735 ± 370
5	220 (-1)	26 (-1)	7 (+1)	4383 ± 240
6	260 (+1)	26 (-1)	7 (+1)	6606 ± 21
7	220 (-1)	28 (+1)	7 (+1)	4179 ± 100
8	260 (+1)	28 (+1)	7 (+1)	4720 ± 560
9	200 (-2)	27 (0)	6 (0)	7150 ± 150
10	280 (+2)	27 (0)	6 (0)	7350 ± 150
11	240 (0)	25 (-2)	6 (0)	4925 ± 175
12	240 (0)	29 (+2)	6 (0)	6375 ± 125
13	240 (0)	27 (0)	4 (-2)	650 ± 98
14	240 (0)	27 (0)	8 (+2)	6210 ± 192
15	240 (0)	27 (0)	6 (0)	8186 ± 548
16	240 (0)	27 (0)	6 (0)	7899 ± 34
17	240 (0)	27 (0)	6 (0)	7897 ± 420
18	240 (0)	27 (0)	6 (0)	8111 ± 295
19	240 (0)	27 (0)	6 (0)	8125 ± 125
20	240 (0)	27 (0)	6 (0)	7925 ± 175

[a] Moyenne de deux lectures

IV.3.2. Modélisation prédictive : Description du modèle neuronal RNOEP

Afin de relier le rendement en pleuromutiline avec la vitesse d'agitation, la température et le *pH* initial, on a construit un modèle prédictif à base de RN type feedforward (RNFF). Le RNFF proposé renferme trois couches: une couche d'entrée avec trois nœuds représentant les facteurs étudiés, une seule couche cachée (une seule couche cachée est jugée suffisante vue le nombre limité des points expérimentaux) et une couche de sortie avec un neurone générant la valeur estimée du rendement en pleuromutiline (Y).

Afin de rendre l'apprentissage du RNFF plus efficace en empêchant la fonction de transfert de devenir saturé et rendre l'apprentissage très rapide, toutes les variables d'entrée et de sortie ont été normalisées dans l'intervalle (IN) 0 à 1 et -1 à 1 suivant la fonction de transfert attribuée à la couche cachée FTCC (log-sigmoïde ou tangente hyperbolique sigmoïde, respectivement), en utilisant l'équation de normalisation suivante :

$$x_N = (y_{\max} - y_{\min})\left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) + y_{\min} \qquad \textbf{(IV.2)}$$

Où x_N est la valeur normalisée du paramètre x (vitesse d'agitation, température, *pH* initial et *Y*) ; x_{max} et x_{min} sont respectivement les valeurs maximales et minimales de x; y_{min} *et* y_{max} peuvent prendre les valeurs 0 et 1 ou -1 et 1.

La fonction de transfert linéaire "purelin" a été attribuée à la couche de sortie.

L'algorithme d'OEP (trainpso en code MATLAB) est utilisé dans l'apprentissage du réseau de neurones (ce réseau est noté RNOEP). Cette méthode d'apprentissage recherche le nombre de neurones dans la couche cachée qui produisent de bonnes qualités de prédictions par optimisation des poids et des biais (Chau 2006; Lazzús 2009). Les quatre variantes de l'algorithme d'OEP (voir chapitre II) ont été testées dans l'apprentissage du RNOEP.

Vue le nombre limité des points expérimentaux, la base de données (20 expériences au total) a été aléatoirement répartie en deux sous-ensembles, en réservant 80 % des données pour la phase d'apprentissage et 20 % pour la phase de test.

Afin d'optimiser l'architecture du RNOEP, une procédure d'essai et erreur a été exécutée comme décrite sur la figure IV.3. Le nombre de neurones (N) dans la couche cachée a été varié de 2 à 10.

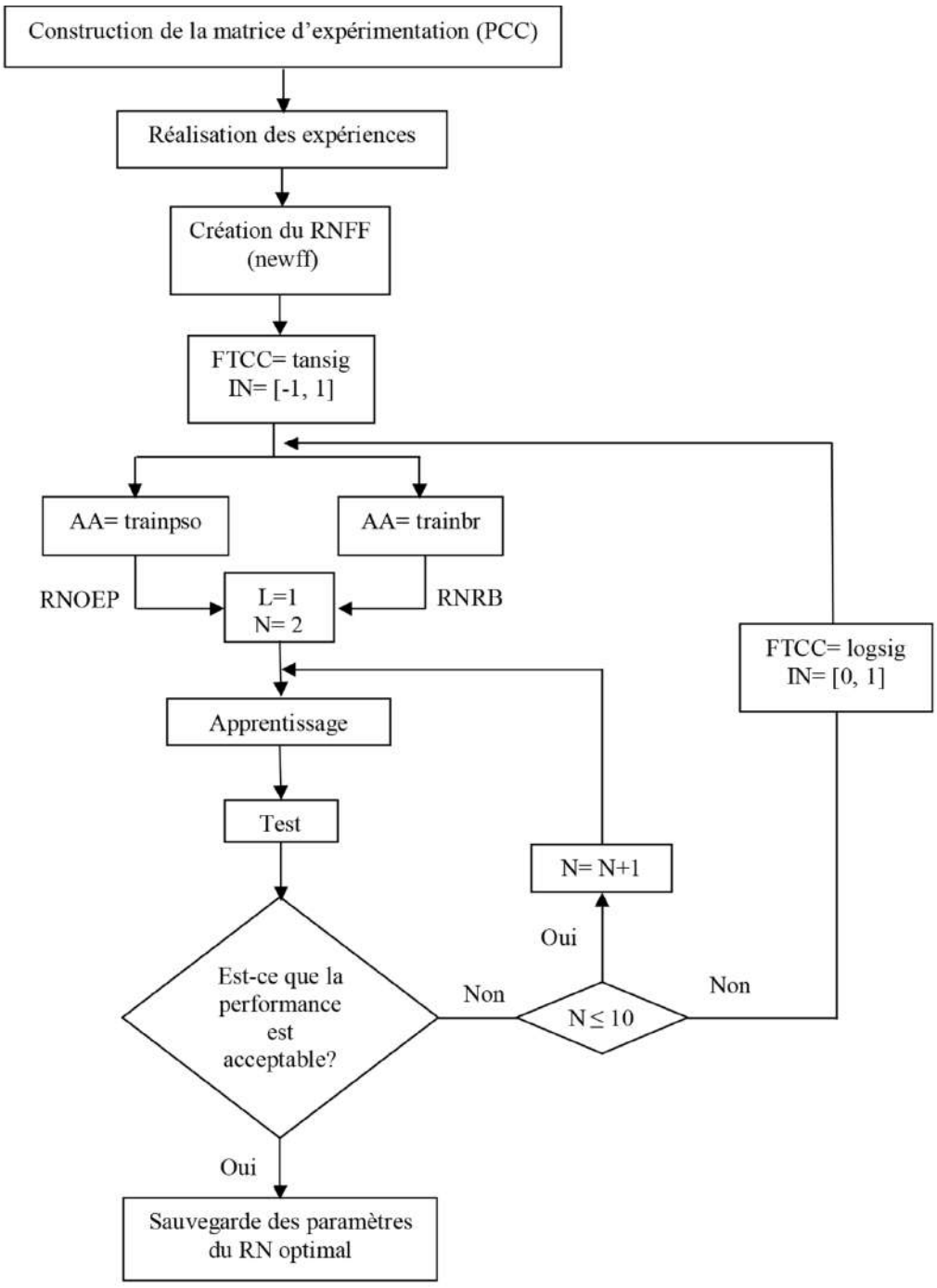


Figure IV.3. Procédure de développement du RNOEP et du RNRB.

IV.3.2.1. Comparaison avec le réseau de neurones formé avec l'algorithme de régularisation bayésienne (RNRB)

À titre de comparaison, un autre RNFF formé avec l'algorithme de rétro-propagation de type régularisation bayésienne (trainbr en code MATLAB) a été également utilisé pour décrire la relation entre le rendement en pleuromutiline avec la vitesse d'agitation, la température et le *pH* initial (ce réseau est noté RNRB). L'algorithme de régularisation bayésienne, qui s'est avérer très performant dans le troisième chapitre de cette thèse de doctorat, trouve de nombreuses applications dans l'apprentissage des RN utilisés à des fins d'optimisation (Rosa *et al.* 2010; Lu *et al.* 2011).

Le RNRB possède une structure similaire à celle du RNOEP (une couche d'entrée avec trois nœuds représentant les facteurs étudiés, une seule couche cachée et une couche de sortie avec un neurone générant la valeur estimée du rendement en pleuromutiline). La procédure utilisée pour optimiser l'architecture du RNRB est décrite sur la figure IV.3.

IV.3.2.2. Comparaison avec la méthodologie de surface de réponse (MSR)

En vertu de la MSR, la relation entre les variables sélectionnées s'exprime mathématiquement sous la forme d'une équation polynomiale du second degré :

$$Y(\mu g/g) = \beta_0 + \sum \beta_i X_i + \sum \beta_{ii} X_i^2 + \sum \beta_{ij} X_i X_j \qquad \textbf{(IV.3)}$$

Où Y est la variable de réponse prédite ; β_o est une constante représentant l'effet moyen général *;* β_i sont les coefficients de régression linéaire ; β_{ii} sont les coefficients de régression quadratique et β_{ij} sont les coefficients de régression associés aux termes d'interaction ; X_i et X_j ($i = 1, 2, 3$; $j = 1, 2, 3$) représentent les variables indépendantes sous forme de valeurs codées.

IV.3.2.3. Evaluation des performances de prédiction des modèles

Afin d'évaluer l'aptitude et la précision de prédiction des modèles neuronaux (RNOEP et RNRB) et du modèle polynomial, la racine de l'erreur quadratique moyenne (*REQM*), l'erreur standard de prédiction (*ESP*) et l'erreur de prédiction du modèle (*EPM*) sont utilisées en parallèle avec le facteur de biais (B_f) et le facteur de précision (A_f) (Kumar *et al.* 2011; Lu *et al.* 2011; Zafar *et al.* 2012a; Zafar *et al.* 2012b).

$$REQM = \sqrt{\frac{\sum_{i=1}^{n} \left(Y_{i,\exp} - Y_{i,pred}\right)^2}{n}} \qquad \textbf{(IV.4)}$$

$$ESR(\%)=\frac{REQM}{Y_e}\times 100 \quad \textbf{(IV.5)}$$

$$EPM(\%)=\frac{100}{n}\sum_{i=1}^{n}\left|\left(Y_{i,\exp}-Y_{i,pred}\right)/Y_{i,\exp}\right| \quad \textbf{(IV.6)}$$

$$B_f=10^{\left(\sum_{i=1}^{n}\log\left(Y_{i,pred}/Y_{i,\exp}\right)/n\right)} \quad \textbf{(IV.7)}$$

$$A_f=10^{\left(\sum_{i=1}^{n}\left|\log\left(Y_{i,pred}/Y_{i,\exp}\right)\right|/n\right)} \quad \textbf{(IV.8)}$$

Où $Y_{i,exp}$ est la valeur expérimentale ; $Y_{i,pred}$ représente la valeur prédite ; Y_e est la moyenne des données expérimentales ; n est le nombre d'expériences.

IV.3.3. Optimisation par essaim de particules

Une fonction générique de l'algorithme d'OEP (PSO en code MATLAB), récemment développée pour MATLAB par Ebbesen, Kiwitz et Guzzella (Ebbesen *et al.* 2011), a été utilisée pour optimiser l'espace d'entrée du modèle neuronal en vue d'obtenir une production maximale en sortie.

IV.4. Logiciels utilisés

Le logiciel statistique Modde 5,0 (Umetrics) a été utilisé dans la conception de la matrice expérimentale du plan composite centré (PCC), la modélisation des données expérimentales par le polynôme quadratique et l'analyse des données.

La sixième version du logiciel MATLAB et le toolbox de l'algorithme d'OEP ont été utilisés pour la conception et l'apprentissage des modèles neuronaux.

La fonction générique PSO (Ebbesen *et al.* 2011) et la septième version de MATLAB ont été utilisés dans la tâche d'optimisation.

IV.5. Résultats et discussion

IV.5.1. Modélisation par le RNOEP

Un modèle neuronal formé avec l'algorithme d'OEP a été construit pour modéliser la relation entre le rendement en pleuromutiline et les trois conditions de culture de *Pleurotus mutilus* (vitesse d'agitation, température et *pH* initial).

L'architecture optimale obtenue du RNOEP a une topologie de 3-4-1 (Tableau IV.5) : trois neurones dans la couche d'entrée (vitesse d'agitation, température et *pH* initial), quatre neurones dans la couche cachée et un neurone dans la couche de sortie (rendement en pleuromutiline). La variante du facteur de constriction de l'algorithme d'OEP utilisés dans la phase d'apprentissage s'est avéré la plus performante par rapport aux autres variantes. Les paramètres de contrôles de cette variante ont été obtenus par essai et erreur et sont regroupés dans le tableau IV.6. L'équation extraite de ce modèle neuronal optimisé (Figure IV.4) est comme suit :

$$Y\left(10^3\,\mu g/g\right)=\sum_{j=1}^{4} w_{1j}^{H}\left[\frac{2}{1+\exp(-2\sum_{i=1}^{3} w_{ji}^{I}X_i+b_j^{H})}-1\right]+b_1^{O} \qquad \textbf{(IV.9)}$$

Avec :

$$\begin{cases} 200\ tr/min \le X_1 \le 280\ tr/min \\ 25°C \le X_2 \le 29°C \\ 4 \le X_3 \le 8 \end{cases}$$

Où w_{ji}^{I} représente le poids synaptique de la connexion du neurone *i* de la couche d'entrée au neurone *j* de la couche cachée ; w_{1j}^{H} représente le poids synaptique de la connexion du neurone *j* de la couche cachée au neurone de la couche de sortie ; b_j^{H} et b_1^{O} représentent respectivement les biais du neurone *j* de la couche cachée et du neurone de la couche de sortie. Les poids et les biais du RNOEP optimisé sont regroupés dans le tableau IV.7.

Tableau IV.5. Structure des réseaux de neurones optimisés (RNOEP et RNRB).

Type du réseau	Algorithme d'apprentissage	Couche d'entrée	Couche cachée		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
RNOEP	trainpso	3	4	tansig	1	purelin
RNRB	trainbr	3	8	tansig	1	purelin

Tableau IV.6. Paramètres de commande utilisés pour l'exécution de la variante du facteur de constriction de l'algorithme d'OEP lors de l'apprentissage du RNOEP.

Paramètre	Valeur
Nombre de population	50
Nombre de générations	2000
Facteurs d'apprentissage	1.3
Poids d'inertie	0.6 et 0.9
c_1	2.05
c_2	2.05

Tableau IV.7. Poids et biais du RNOEP optimal.

Connexions Couche d'entrée-Couche cachée				Connexions Couche cachée-Couche de sortie	
Poids			Biais	Poids	Biais
w_{j1}^I	w_{j2}^I	w_{j3}^I	b_j^H	w_{1j}^H	b_1^0
0.4932	-0.7624	5.1285	-2.1210	-0.4951	-0.3097
42.9328	0.1389	39.4624	0.3183	0.3582	
12.0855	42.3870	-29.8827	-0.1253	-0.3545	
0.2405	-72.8371	-83.5661	-5.4771	-0.4730	

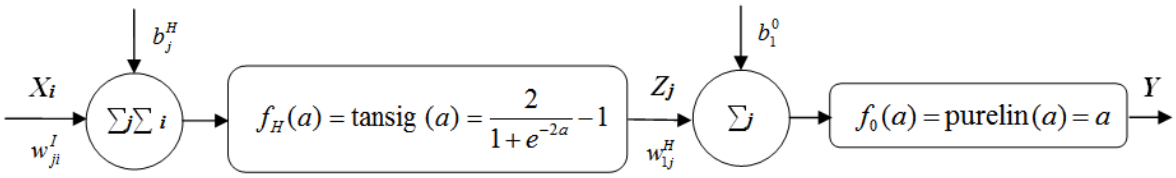


Figure IV.4. Représentation schématique du modèle neuronal (RNOEP ou RNRB).

On a procédé à une analyse de sensibilisé du RNOEP optimal par la méthode de "Poids" et la méthode de "Profil" (voir la section III.2.4 du chapitre III pour plus de détail). La figure IV.5 donne une présentation des % de l'importance relative (*Ir*), obtenue par la méthode de "Poids", de chaque variable d'entrée du RNOEP sur le rendement. D'après cette figure, le *pH* initial de la fermentation a la contribution la plus importante suivie de la température. La vitesse d'agitation possède le % de l'*Ir* le plus faible par rapport aux deux autres variables, mais sa contribution est aussi importante que les deux autres (% de l'Ir > 5%).

La figure IV.6 illustre les graphes du profil de contribution des variables d'entrées simulées sur le rendement (*Y*) prédit par le RNOEP optimal, on constate que :

- Les valeurs des vitesses d'agitations comprises entre 200 et 243 tr/min ont peu d'effet sur le rendement, entre 243 et 250 tr/min on observe une augmentation de *Y*, alors qu'au delà de 250 tr/min, le rendement diminue. On peut dire que la zone optimale se situe entre 243 et 250 tr/min (Figure IV.6 (a)).
- Plus la température augmente, plus le *Y* prédit par le RNOEP augmente jusqu'à la température de 27.5°C ; au delà de cette valeur, le rendement diminue puis devient peu influent (constant). On conclue que la zone optimale de température se situe entre 27 et 27.5°C (Figure IV.6 (b)).
- Le *pH* initial est le facteur le plus influent sur le rendement, plus le *pH* augmente plus le *Y* prédit par le RNOEP augmente jusqu'au *pH* de 6.2 ; au delà de cette valeur le rendement diminue. La zone optimale du *pH* se situe entre 6 et 6.5 (Figure IV.6 (c)).

Les contributions relatives de chaque variable d'entrée, obtenues par la méthode de profil, ont été exprimées par les valeurs de l'intervalle (maximum-minimum) de leurs contributions (Figure IV.5). On constate que les % de contributions relatives de chaque variable dépasse les 5%, montrant ainsi l'importance des trois variables sur les sorties du RNOEP. Cette analyse de sensibilité par les deux méthodes utilisées montre à posteriori l'importance des trois conditions de

culture choisies pour la modélisation du rendement de production de pleuromutiline et constituent par conséquent une justification du choix postulé antérieurement.

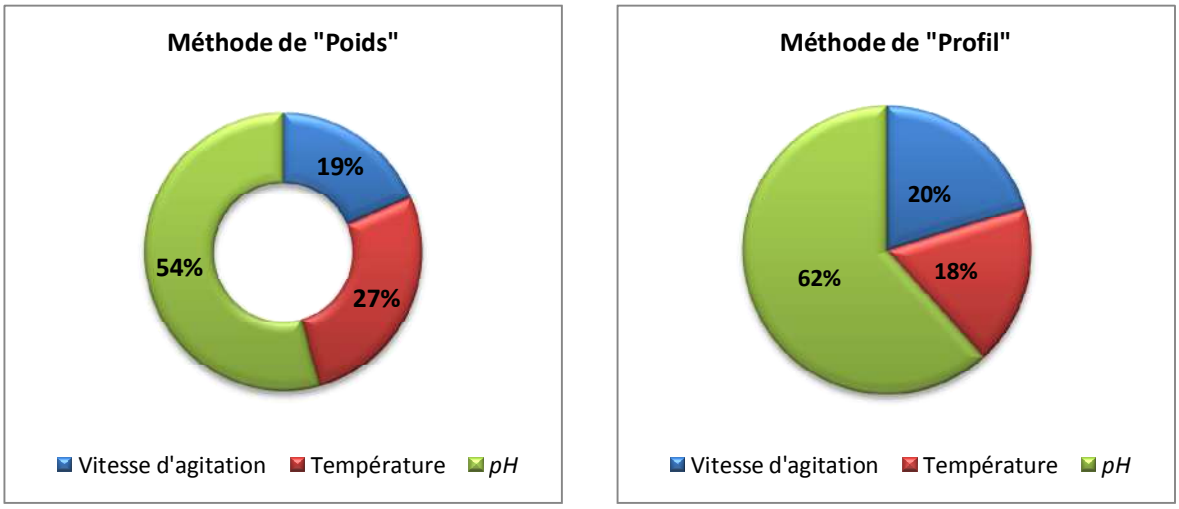
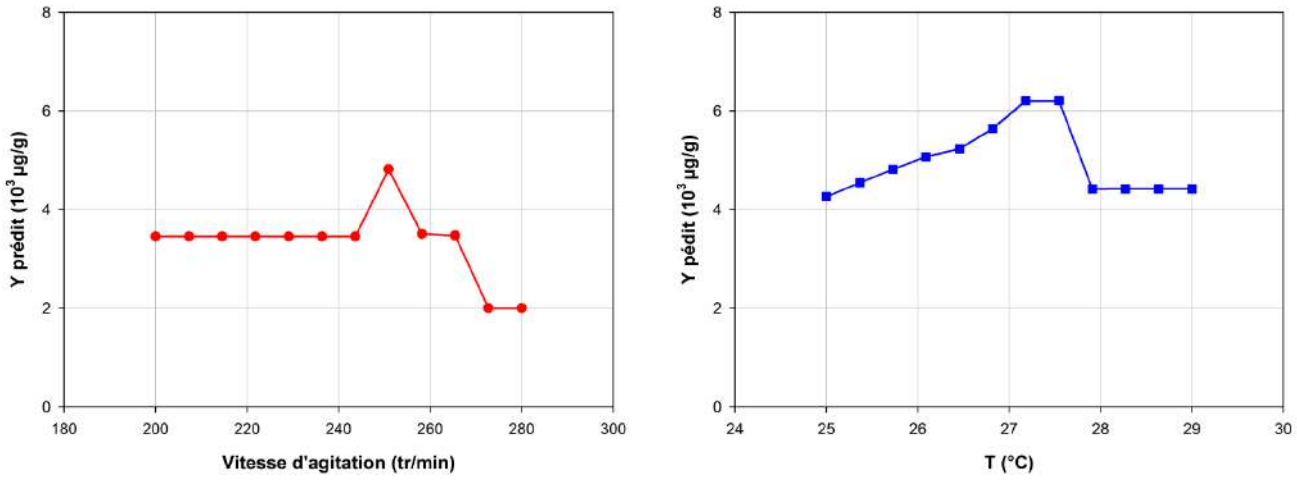
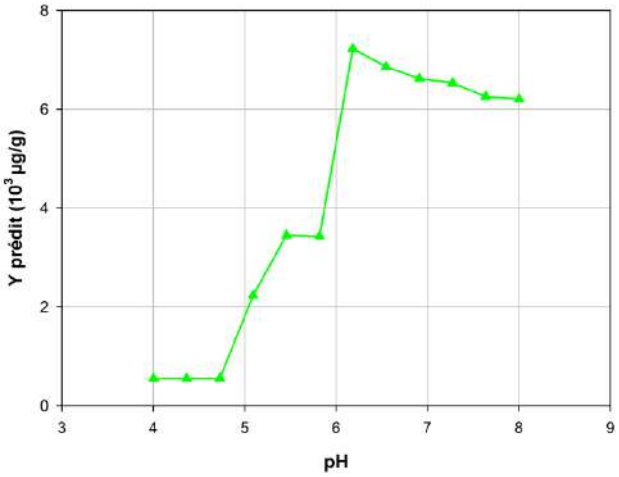


Figure IV.5. Présentation des % de l'importance relative (*Ir*) (obtenus par la méthode de "poids") et des % de contribution relative (obtenus par la méthode de "profil") de chaque variable d'entrée du RNOEP sur le rendement.



(a) **(b)**



(c)

Figure IV.6. Graphes du profil de contributions des variables simulées (Vitesse d'agitation, température, *pH* initial) sur le rendement prédit par le RNOEP : **(a)** effet de la vitesse d'agitation, **(b)** effet de la température, **(c)** effet du *pH* initial.

IV.5.2. Modélisation par le RNRB

La topologie optimale obtenue du RNRB est de 3-8-1 (Tableau IV.5) : trois neurones dans la couche d'entrée, huit neurones dans la couche cachée et un neurone dans la couche de sortie. L'équation extraite de ce modèle neuronal (RNRB) optimisé est comme suit :

$$Y\left(10^3\,\mu g/g\right)=\sum_{j=1}^{8} w_{1j}^{H}\left[\frac{2}{1+\exp(-2\sum_{i=1}^{3} w_{ji}^{I}X_i+b_j^{H})}-1\right]+b_1^{O} \qquad \textbf{(IV.10)}$$

Avec :

$$\begin{cases} 200\ tr/min \le X_1 \le 280\ tr/min \\ 25°C \le X_2 \le 29°C \\ 4 \le X_3 \le 8 \end{cases}$$

Où w_{ji}^{I} représente le poids synaptique de la connexion du neurone i de la couche d'entrée au neurone j de la couche cachée ; w_{1j}^{H} représente le poids synaptique de la connexion du neurone j de la couche cachée au neurone de la couche de sortie ; b_j^{H} et b_1^{O} représentent respectivement les biais

du neurone *j* de la couche cachée et du neurone de la couche de sortie. La matrice des poids et biais du RNRB optimisé est présentée dans le tableau IV.8.

Tableau IV.8. Poids et biais du RNRB optimal.

Connexions Couche d'entrée-Couche cachée				**Connexions Couche cachée-Couche de sortie**	
Poids			**Biais**	**Poids**	**Biais**
w_{j1}^{I}	w_{j2}^{I}	w_{j3}^{I}	b_{j}^{H}	w_{1j}^{H}	b_{1}^{0}
0.3404	-0.0496	2.1947	-0.5848	-1.4658	-0.9691
-0.0267	0.0309	0.0930	-0.2332	0.2652	
-0.0267	0.0309	0.0930	-0.2329	0.2649	
-0.0114	-0.8538	-0.7748	-0.4143	-0.8710	
1.2162	0.5303	0.5769	0.0664	1.0575	
0.1717	1.5359	-0.0385	-0.5000	-0.9831	
-0.4016	-0.0915	1.3834	0.2917	1.0807	
-0.0267	0.0309	0.0930	-0.2330	0.2650	

On a procédé à une analyse de sensibilisé du RNRB optimal par la méthode de "Poids" et la méthode de "Profil" (voir chapitre III pour plus de détail). La figure IV.7 donne une présentation des % de l'importance relative (*Ir*), obtenue par la méthode de "Poids", de chaque variable d'entrée du RNRB sur le rendement. D'après cette figure le *pH* initial de la fermentation a la contribution la plus importante, suivie de la température. La vitesse d'agitation possède le % de l'*Ir* le plus faible par rapport au deux autres variables, mais sa contribution est aussi importante que les deux autres.

La figure IV.8 illustre les graphes du profil de contribution des variables d'entrées simulées sur le rendement (*Y*) prédit par le RNRB optimal, on constate que :

- Les valeurs des vitesses d'agitations comprise entre 200 et 230 tr/min ont peu d'effet sur le rendement, entre 230 et 250 tr/min on observe une augmentation de *Y*, alors qu'au delà de 250 tr/min, le rendement prédit par le RNRB diminue. On peut dire que la zone optimale se situe entre 240 et 250 tr/min (Figure IV.8 (a)).
- Plus la température augmente, plus le *Y* prédit par le RNRB augmente jusqu'à la température de 26.5°C ; au delà de cette valeur, le rendement diminue. On conclue que la zone optimale de température se situe entre 26 et 27°C (Figure IV.8 (b)).

- Le *pH* initial est le facteur le plus influent sur le rendement, plus le *pH* augmente plus le *Y* prédit par le RNRB augmente jusqu'au *pH* de 6.2 ; au delà de cette valeur le rendement diminue. La zone optimale du *pH* initial se situe entre 6 et 6.2 (Figure IV.8 (c)).

Les contributions relatives de chaque variable d'entrée, obtenues par la méthode de "profil", ont été exprimées par les valeurs de l'intervalle (maximum-minimum) de leurs contributions (Figure IV.7). On constate que les % de contributions relatives de chaque variable dépasse les 5%, montrant ainsi l'importance des trois variables sur les sorties du RNRB. Cette analyse de sensibilité par les deux méthodes utilisées montre à posteriori l'importance des trois conditions de culture choisies pour la modélisation du rendement de production de pleuromutiline et constituent par conséquent une justification du choix postulé antérieurement.

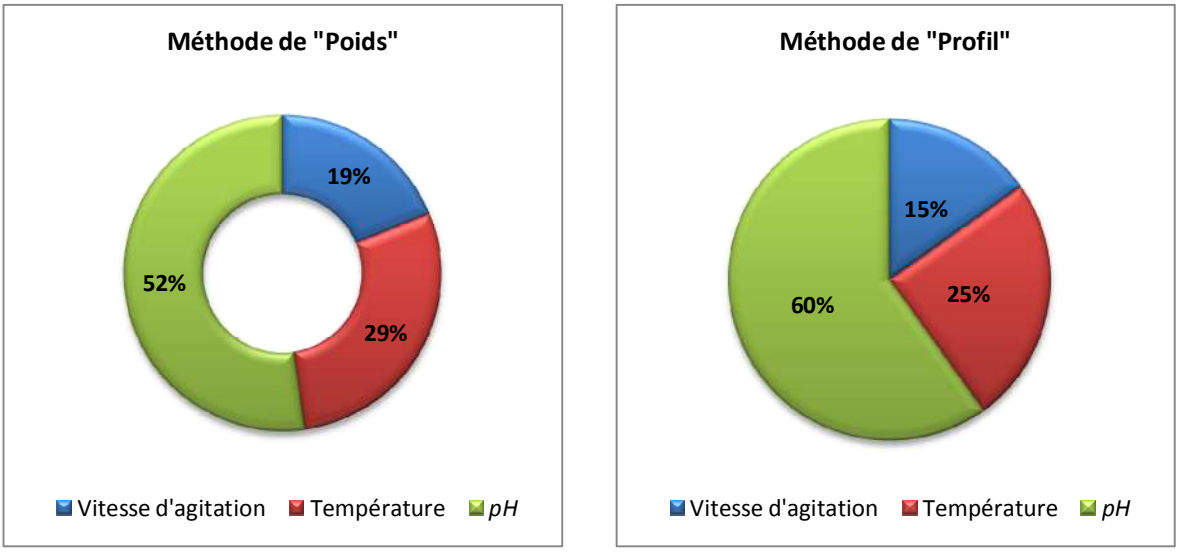


Figure IV.7. Présentation des % de l'importance relative (*Ir*) (obtenus par la méthode de "poids") et des % de contribution relative (obtenus par la méthode de "profil") de chaque variable d'entrée du RNRB sur le rendement.

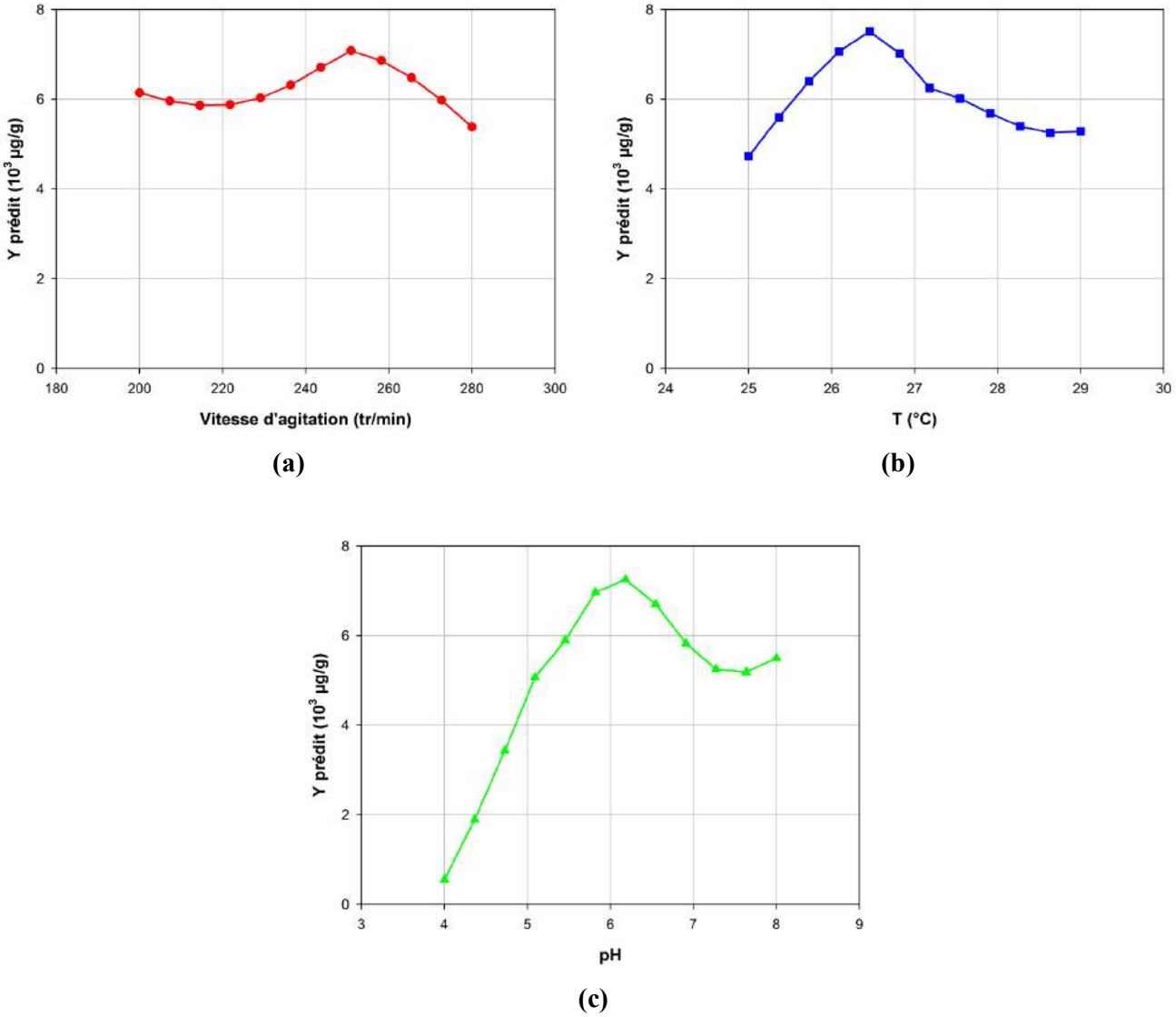


Figure IV.8. Graphes du profil de contributions des variables simulées (Vitesse d'agitation, Température, *pH* initial) sur le rendement prédit par le RNRB: **(a)** effet de la vitesse d'agitation, **(b)** effet de la température, **(c)** effet du *pH* initial.

IV.5.3. Modélisation par la MSR

Le modèle polynomial quadratique obtenu sous la MSR (Equation IV.11) a donné lieu à un mauvais ajustement (les coefficients de régression globale (R^2) et ajusté ($R^2_{ajusté}$) ont des valeurs très faibles de l'ordre de 0.765 et de 0.554).

$$Y(\mu g/g) = 7773.73 + 469.188\ X_1 + 38.8124\ X_2 + 824.562\ X_3 - 318.511\ X_1^2 - 718.511\ X_2^2 - 1273.51\ X_3^2 - 88.875\ X_1 X_2 - 197.376\ X_1 X_3 - 237.625\ X_2 X_3 \quad \textbf{(IV.11)}$$

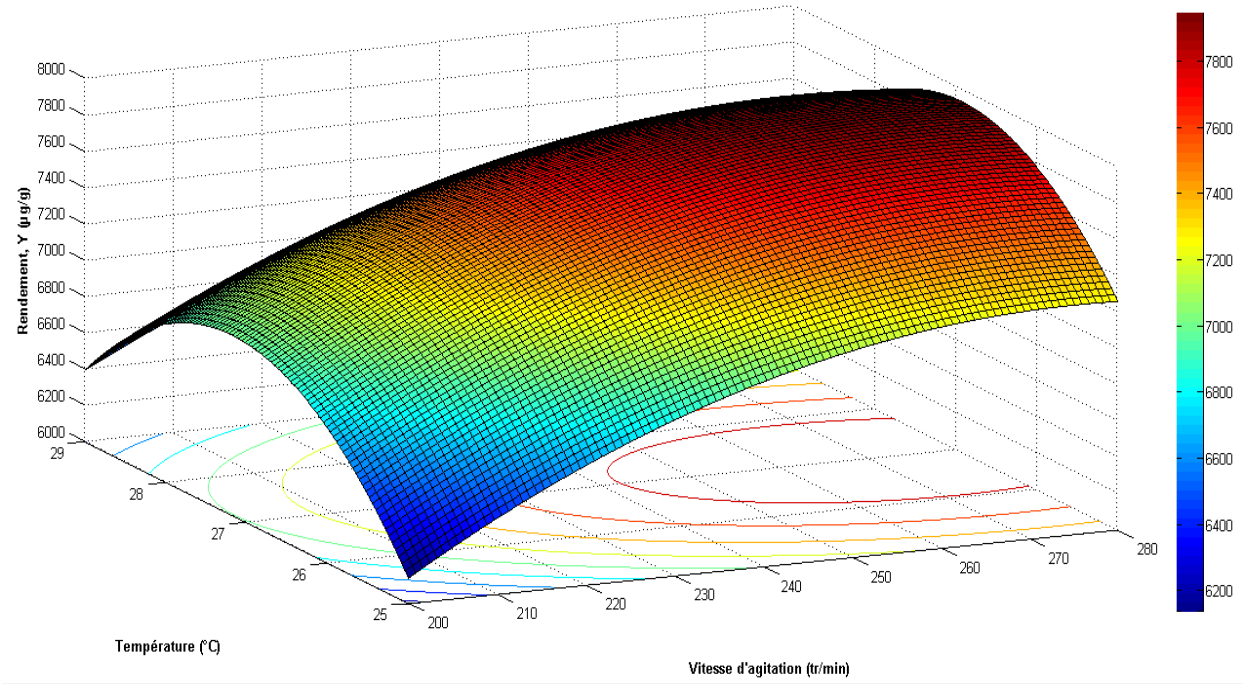
L'estimation des divers paramètres inconnus du modèle (Tableau IV.9) révèle que seuls trois coefficients du modèle polynomial sont jugés significatifs (*P*-value <0.05) : l'effet linéaire du *pH* et l'effet quadratique de la température et du *pH*. Ces résultats sont jugés acceptables tant que la valeur de $R^2_{ajusté}$ est plus petite que R^2, le modèle contient plusieurs termes et la taille de l'échantillon est faible. L'utilisation d'un $R^2_{ajusté}$ est une tentative de prendre en compte le phénomène de rétrécissement statistique, la tendance des modèles, à moins bien réussir, hors échantillon, que dans l'échantillon. Le $R^2_{ajusté}$ est une modification en raison de R^2 qui ajuste le nombre de termes explicatifs dans un modèle par rapport au nombre de points de données.

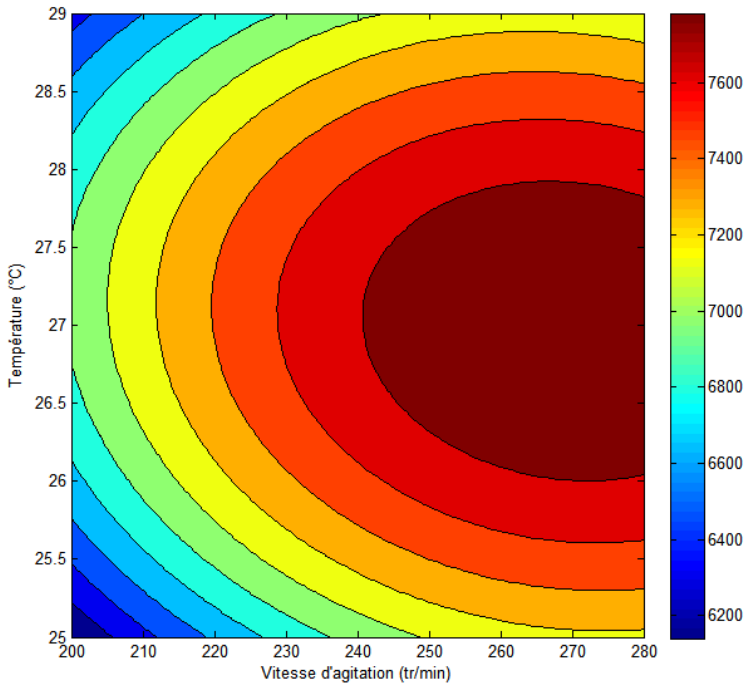
Les effets d'interaction entre deux paramètre tout en gardant le troisième constant (à son niveau central) sont représentés par des tracés de surfaces de réponses et de contours (Figure IV.9). Ces représentations graphiques sont issues du modèle de l'équation IV.11. Il a été noté que le rendement augmente avec l'augmentation de la vitesse d'agitation, de la température et du *pH*. La zone du rendement maximal se situe dans les intervalles suivants : Température (°C) = [25.5-28] ; Vitesse d'agitation (tr/min) = [240-280] et *pH* = [5.5-7.5].

Tableau IV.9. Coefficients du modèle estimé par analyse de régression linéaire multiple.

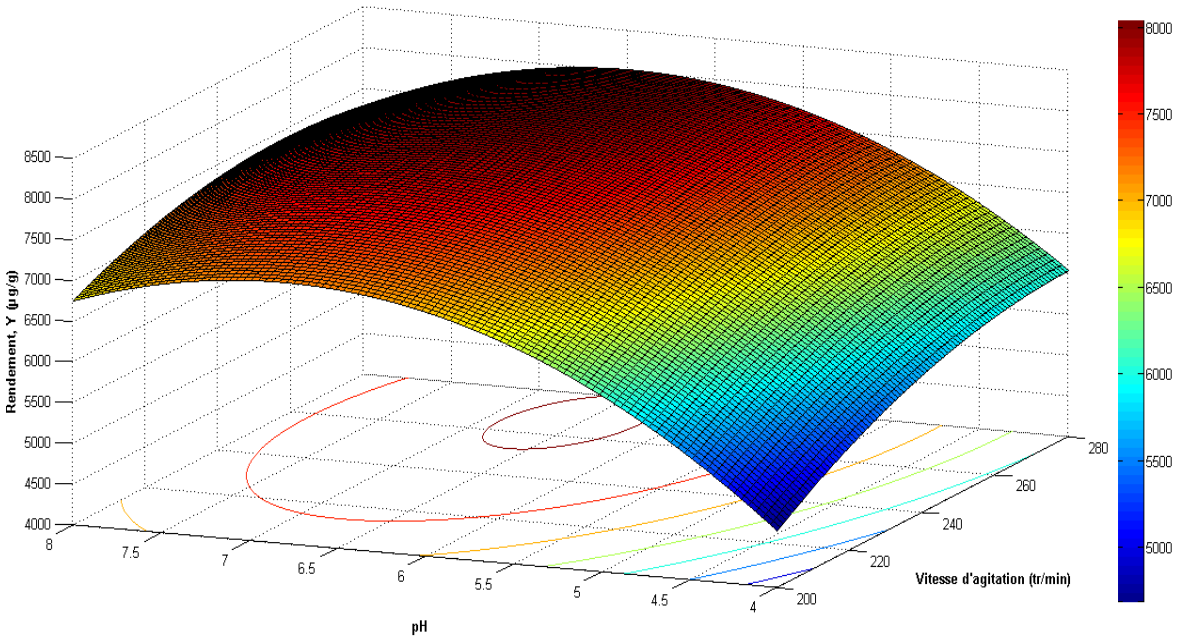
Sources	**Coeff. SC**	**Std. Err.**	***P-value***
Moyenne / intercepté	7773.73	547.267	$5.90\ 10^{-8}$ [a]
X_1	469.188	343.018	0.2013
X_2	38.8124	343.018	0.9122
X_3	824.562	343.017	0.0371 [a]
X_1^2	-318.511	273.633	0.2714
X_2^2	-718.511	273.633	0.0253 [a]
X_3^2	-1273.51	273.633	0.0009 [a]
X_1X_2	-88.875	485.1	0.8583
X_1X_3	-197.376	485.1	0.6927
X_2X_3	-237.625	485.1	0.6348

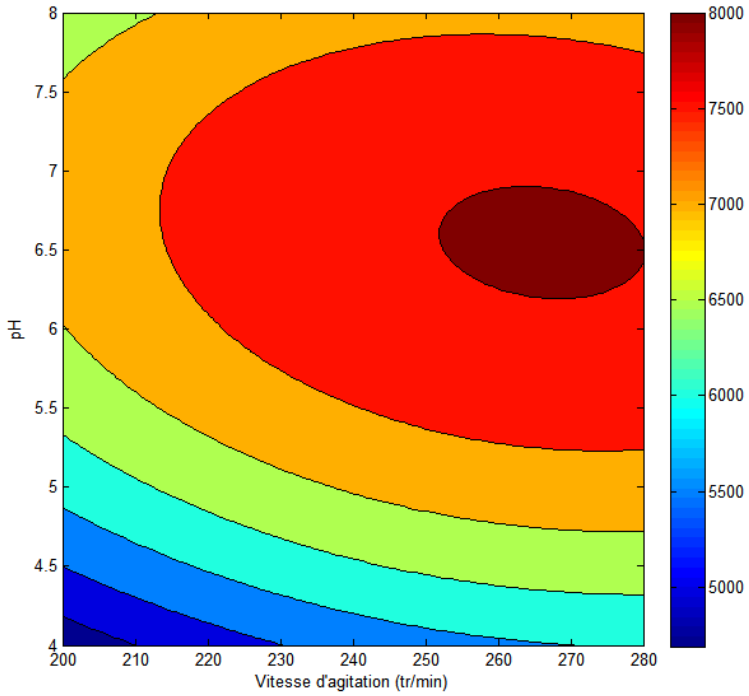
[a] Termes significatifs (*P*-value < 0.05)


Rendement, Y (µg/g)
8000
7800
7600
7400
7200
7000
6800
6600
6400
6200
6000
Température (°C)
29
28
27
26
25
Vitesse d'agitation (tr/min)
200
210
220
230
240
250
260
270
280
7800
7600
7400
7200
7000
6800
6600
6400
6200

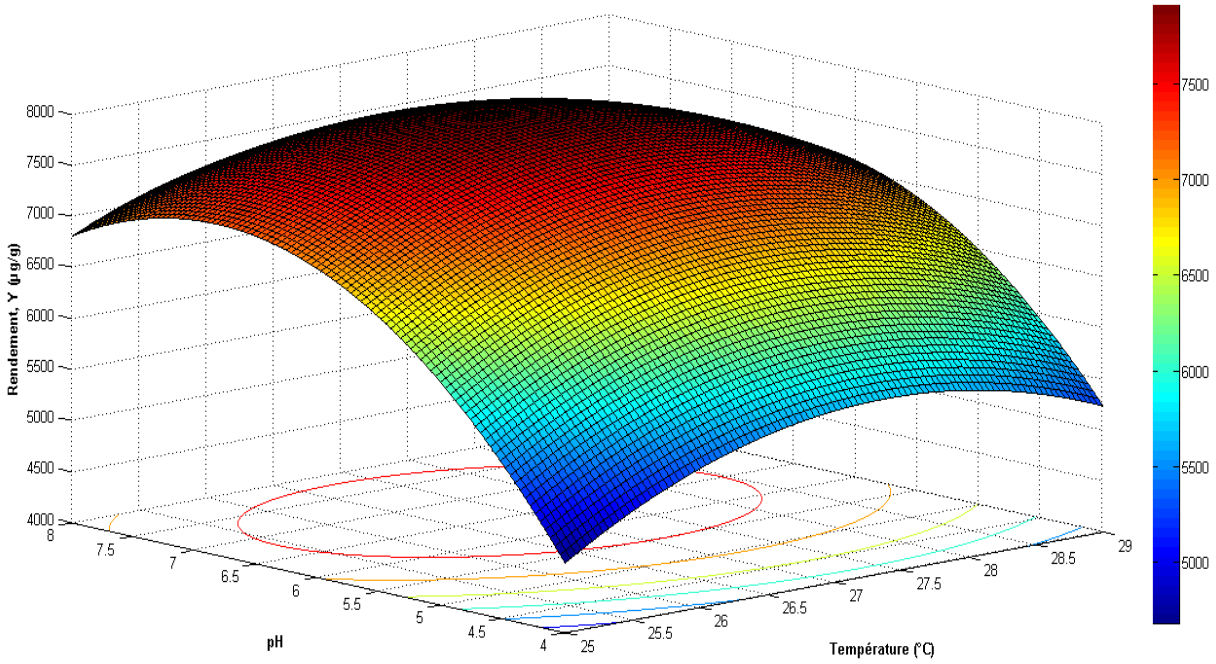

Température (°C)
29
28.5
28
27.5
27
26.5
26
25.5
25
Vitesse d'agitation (tr/min)
200
210
220
230
240
250
260
270
280
7600
7400
7200
7000
6800
6600
6400
6200

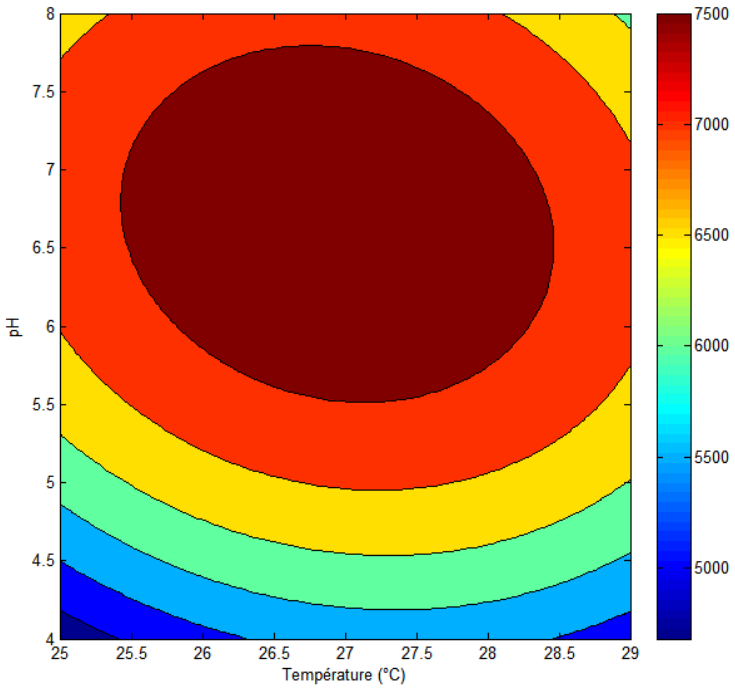
(a)


Rendement, Y (µg/g)
8500
8000
7500
7000
6500
6000
5500
5000
4500
4000
8
7.5
7
6.5
6
5.5
5
4.5
4
pH
200
220
240
260
280
Vitesse d'agitation (tr/min)
8000
7500
7000
6500
6000
5500
5000


pH
8
7.5
7
6.5
6
5.5
5
4.5
4
200
210
220
230
240
250
260
270
280
Vitesse d'agitation (tr/min)
8000
7500
7000
6500
6000
5500
5000

(b)





(c)

Figure IV.9. Tracés de surfaces de réponses et de contours de la MSR : **(a)** effet de la température et de la vitesse d'agitation, **(b)** effet du *pH* initial et de la vitesse d'agitation, **(c)** effet du *pH* initial et de la température.

IV.5.4. Comparaison des modèles prédictifs

Il nous a fallu 7 mois pour la réalisation des 20 expériences batch et la programmation des RN en MATLAB. L'exécution des programmes est réalisée dans l'interface Windows 7 et Windows XP avec 32 bits et une RAM de 2 GB, le temps d'exécution des programmes est illustré sur le tableau IV.10.

Tableau IV.10. Temps d'exécution.

RN	Temps d'exécution (s) du programme du RN	Temps d'exécution (s) de simulation du RNO
RNOEP	$1.4943\ 10^4$	0.0359
RNRB	993.4002	0.0386

Les valeurs de prédictions des trois modèles (MSR, RNOEP et RNRB) sont illustrées dans le tableau IV.11. La figure IV.10 montre une comparaison entre les valeurs expérimentales et calculées par les modèles neuronaux (RNRB et RNOEP) et le modèle polynomial de la MSR, pour la totalité de l'ensemble de données. Comme on peut le constater, les modèles neuronaux (RNRB et RNOEP) s'ajustent avec une excellente précision aux données expérimentales. Le tableau IV.12 rassemble les vecteurs de régression linéaire pour les trois modèles, ce tableau montre que les paramètres de régression les moins favorables sont ceux obtenus par le modèle polynomiale de la MSR.

Il est de pratique courante d'utiliser le coefficient de corrélation (R) pour comparer différents modèles statistiques (Buchanan et Bagi 1994), mais avec le modèle de régression non linéaire, ce terme doit être utilisé avec prudence (Ratkowsky 1990; Panagou et Kodogiannis 2009). Donc, d'autres indices statistiques (*REQM, ESP, EPM,* B_f *et* A_f) sont ajoutés pour la comparaison (Figure IV.11 et Tableau IV.11).

Comme le montre le tableau IV.11, le modèle de la MSR donne les plus grandes valeurs des erreurs par rapport aux modèles neuronaux : la *REQM* est de 970.3 µg/g et l'*ESP* est de 16.38 %. En comparant les deux modèles neuronaux, on constate que la *REQM* et l'*ESP* pour le RNOEP (149.94 µg/g et 2.53 %) sont légèrement inférieurs à ceux du RNRB (154.95 µg/g et 2.62 %). Ces résultats montrent que le modèle neuronal (RNOEP ou RNRB) possède une capacité de modélisation élevée par rapport au modèle polynomial de la MSR.

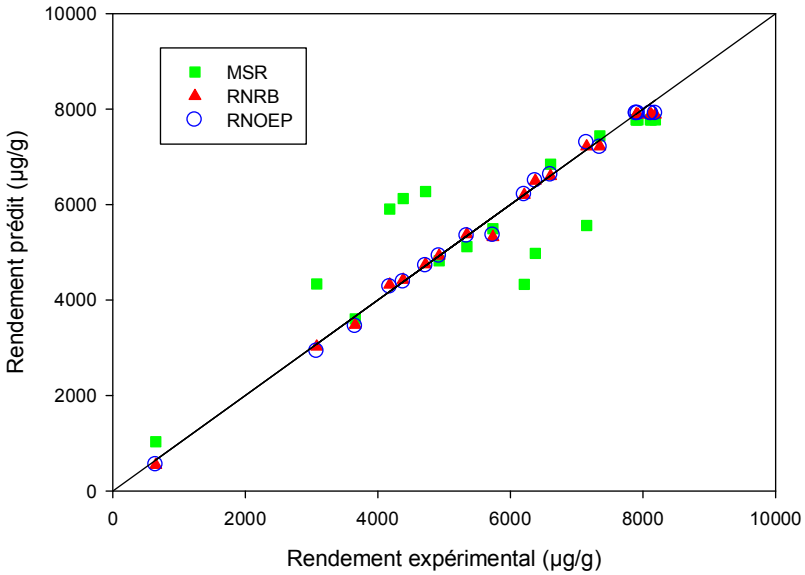


Figure IV.10. Comparaison entre les valeurs expérimentales et calculées par les modèles (MSR, RNRB et RNOEP) pour la totalité de l'ensemble de données.

Le facteur de biais B_f du modèle polynomial de la MSR qui est de 1.03 (supérieur à un et inférieur à 1,1) indique que ce modèle échoue dangereusement ("fail-dangerous" en anglais). Les facteurs de biais B_f du RNOEP et du RNRB, qui sont de 0.98 pour les deux, indiquent que ces modèles sont des modèles à sûreté ("fail-safe" en anglais).

Le facteur de précision A_f est beaucoup plus proche de l'unité pour les deux modèles neuronaux (RNOEP et RNRB) que pour le modèle de la MSR, montrant ainsi une bonne concordance entre les valeurs observées et les valeurs prédites par les modèles neuronaux.

Des erreurs de prédiction (*EPM*) d'environ 2.73 %, 2.7 % et 14.25 % sont respectivement observées pour le RNRB, le RNOEP et le modèle de la MSR, indiquant la supériorité des modèles neuronaux.

La supériorité de l'aptitude de prédiction et de la précision de prédiction d'un modèle neuronal est due à sa capacité universelle d'approximer des systèmes non-linéaires de bien meilleure façon, en particulier pour les processus biologiques, alors que le modèle de la MSR est limité à une équation polynomiale du second degré. Des observations semblables ont été faites sur les données de quelques articles sur la MSR (Andersson et Adlercreutz 1999; Beg *et al.* 2002; Singh *et al.* 2009; Gurunathan et Sahadevan 2011). Par conséquent, il est fiable pour les modèles du

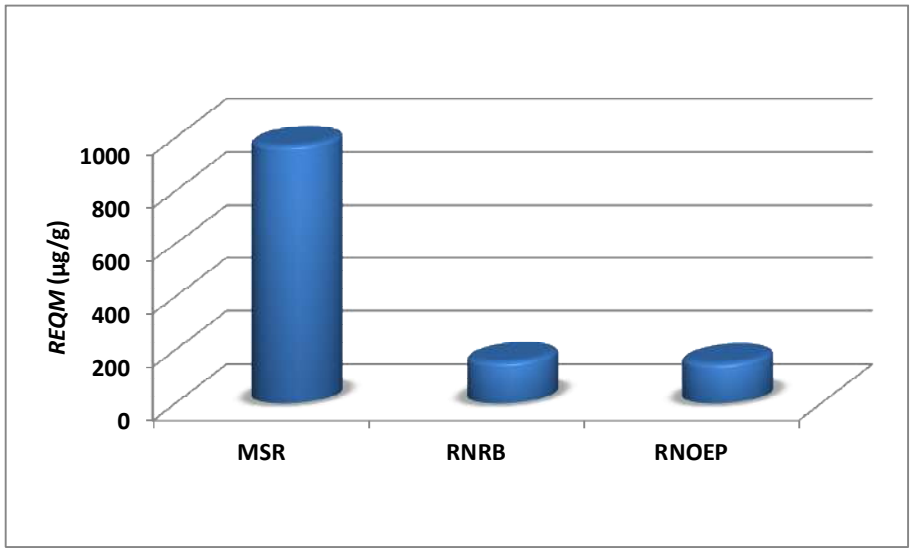
RNRB et du RNOEP d'être employé en tant que fonctions objet ("fitness function" en anglais) dans l'étape de l'optimisation.

Tableau IV.11. Rendement expérimental et prédit par les modèles.

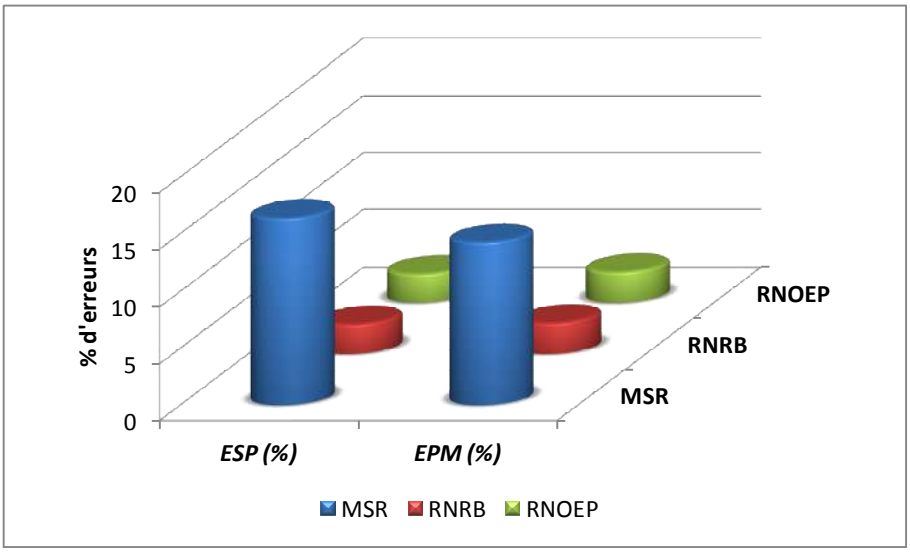
Essai	Rendement de production (µg/g)			
	Expérimental	Prédit par		
		MSR	RNRB	RNOEP
1	3658 ± 210	3606.76	3480.13	3450.2
2	5344 ± 151	5117.63	5379.25	5343.3
3	3078 ± 150	4337.38	3024.59	2928.1
4	5735 ± 370	5492.76	5320.02	5359.4
5	4383 ± 240	6125.88	4422.99	4382.7
6	6606 ± 21	6847.26	6596.53	6626
7	4179 ± 100	5906.01	4321.80	4278.7
8	4720 ± 560	6271.88	4749.64	4719.7
9	7150 ± 150	5561.3	7219.07	7298.7
10	7350 ± 150	7438.05	7218.42	7200.2
11	4925 ± 175	4822.06	4929.26	4924.8
12	6375 ± 125	4977.31	6498.32	6499.7
13	650 ± 98	1030.56	546.73	550.8
14	6210 ± 192	4328.81	6201.23	6210.3
15	8186 ± 548	7773.73	7884.66	7910
16	7899 ± 34	7773.73	7884.66	7910
17	7897 ± 420	7773.73	7884.66	7910
18	8111 ± 295	7773.73	7884.66	7910
19	8125 ± 125	7773.73	7884.66	7910
20	7925 ± 175	7773.73	7884.66	7910
REQM (µg/g)		**970.30**	**154.95**	**149.94**
ESP (%)		**16.38**	**2.62**	**2.53**
EPM (%)		**14.25**	**2.73**	**2.7**
B_f		**1.03**	**0.98**	**0.98**
A_f		**1.16**	**1.03**	**1.03**

Tableau IV.12. Vecteurs de régression linéaire issue de l'équation linéaire $y^{cal} = \alpha y^{exp} + \beta$, [$\alpha$ (pente), β (ordonnée à l'origine), R (coefficient de corrélation)].

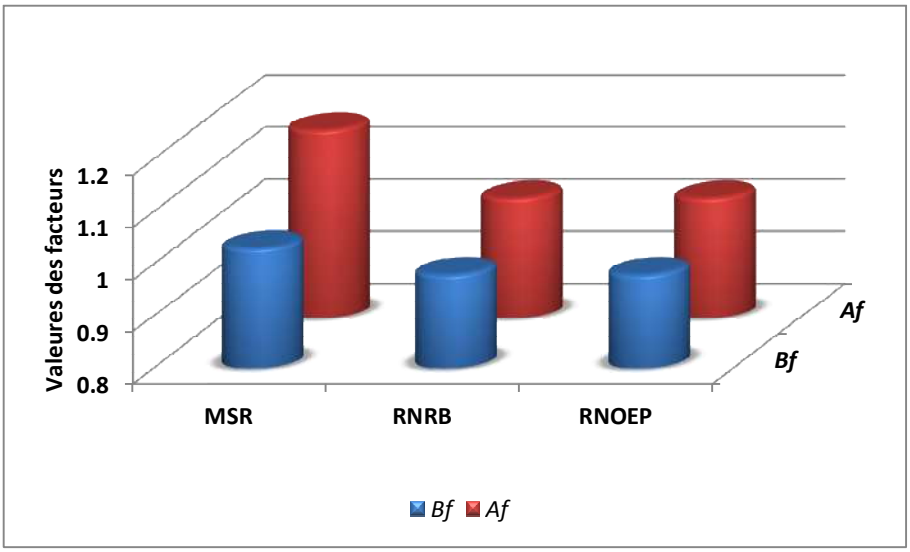
		Totalité de la base
α	MSR	0.765
	RNOEP	0.998
	RNRB	0.9864
β	MSR	1392.3
	RNOEP	-51.812
	RNRB	15.867
R	MSR	0. 8746
	RNOEP	0.9977
	RNRB	0.9975



(a)



(b)



(c)

Figure IV.11. Comparaison entre les modèles : **(a)** en termes de la *REQM*, **(b)** en termes des % d'erreurs (*ESP* et *EPM*), **(c)** en termes des facteurs A_f et B_f.

IV.5.5. Optimisation des modèles neuronaux

Vu que le RNOEP a donné des résultats légèrement meilleurs que ceux du RNRB, alors les deux équations extraites de ces deux modèles neuronaux (Equation IV.9 et Equation IV.10) ont été utilisées comme des fonctions objets dans la tâche d'optimisation.

L'algorithme générique d'OEP (PSO en code MATLAB) a été utilisé pour optimiser l'espace d'entrée (c.-à-d., les conditions de culture) pour chaque modèle neuronal en vue d'obtenir une sortie maximale (c.-à-d., un rendement de production maximal).

Un nombre de particules entre 30 et 100 a été étudié et aucune différence n'a été observée dans la recherche du rendement optimal. Le nombre de particules et les coefficients d'accélération c_1 et c_2 sont respectivement de 50, 1.5 et 1.5. Le poids d'inertie débute avec une valeur élevée de 0,9 et décroit linéairement à 0,4 quand le nombre de générations atteint 50.

Les figures IV.12 et IV.13 illustrent respectivement le profil de convergence des tâches de résolution de l'optimisation du RNRB et du RNOEP avec l'algorithme d'OEP. Pour la méthodologie "RNRB-OEP" (Figure IV.12), lorsque le nombre de générations atteint 12, les conditions optimales sont trouvées et la convergence est très rapide. Lorsque le nombre de générations atteint 28, toutes les particules convergent vers la meilleure solution trouvée. Pour la méthodologie "RNOEP-OEP" (Figure IV.13) lorsque le nombre de générations atteint 17, les conditions optimales sont trouvées et la convergence est moins rapide que celle du "RNRB-OEP". Lorsque le nombre de générations atteint 26, toutes les particules convergent vers la meilleure solution trouvée.

L'approche "RNRB-OEP" prédit une production maximale de 8270 µg/g à une vitesse d'agitation de 258 tr/min, un niveau de température de 26.78 °C et un *pH* initial de 5.98. Sous l'approche "RNOEP-OEP", un rendement maximal en pleuromutiline de 10149 µg/g a été prédit à une vitesse d'agitation de 242 tr/min, une température de 26.88 °C et un *pH* initial de 6.06. On constate que la méthodologie "RNOEP-OEP" prédit une réponse plus élevée avec différentes conditions de culture que celle prédites par la méthodologie "RNRB-OEP". Le niveau optimal de la vitesse d'agitation prédit par "RNOEP-OEP" (242 tr/min) est plus faible que celui prédit par "RNRB-OEP" (258 tr/min). Le niveau de température optimal estimé par "RNOEP-OEP" est proche de celui estimé par "RNRB-OEP". Le *pH* optimal prédit par "RNOEP-OEP" (6.06) est proche de celui prédit par "RNRB-OEP" (5.98). Comme indiqué dans le tableau IV.13, la validation des réponses prédites montre que la méthode "RNOEP-OEP" possède la meilleure précision et aptitude de prédiction.

Le tableau IV.13 et la figure IV.14 montrent que l'approche "RNOEP-OEP" donne un rendement de 27.5 % supérieur à celui obtenu par "RNRB-OEP". Le rendement en pleuromutiline prédit par "RNOEP-OEP" également s'ajuste à celui observé dans les niveaux optimaux beaucoup mieux que dans le cas de "RNRB-OEP" (Tableau IV.13).

Ces résultats indiquent que, même si la vitesse d'agitation optimisée par "RNRB-OEP" était supérieure à celle optimisée par "RNOEP-OEP", ce qui pourrait fournir une plus grande quantité d'oxygène dissous dans le bouillon de culture et pourrait renforcer la production d'antibiotiques, mais en raison des forces et des contraintes de cisaillement élevées, ainsi qu'une abrasion plus importante, la production de pleuromutiline a diminué.

Les résultats de cette étude ont montré que la capacité d'estimation de l'approche hybride "RNOEP-OEP" dans les niveaux optimaux de variables était plus précise que l'approche "RNRB-OEP". Ceci pourrait être attribué au fait que l'algorithme d'apprentissage de régularisation bayésienne (trainbr) présente l'inconvénient de rester coincée à un minimum local, alors que l'algorithme d'apprentissage d'OEP (trainpso), peut rechercher un optimum global en mettant à jour les générations (Zhang *et al.* 2007). À notre connaissance, il n'y a eu aucun rapport sur l'optimisation des bioprocédés par la méthodologie "RNOEP-OEP".

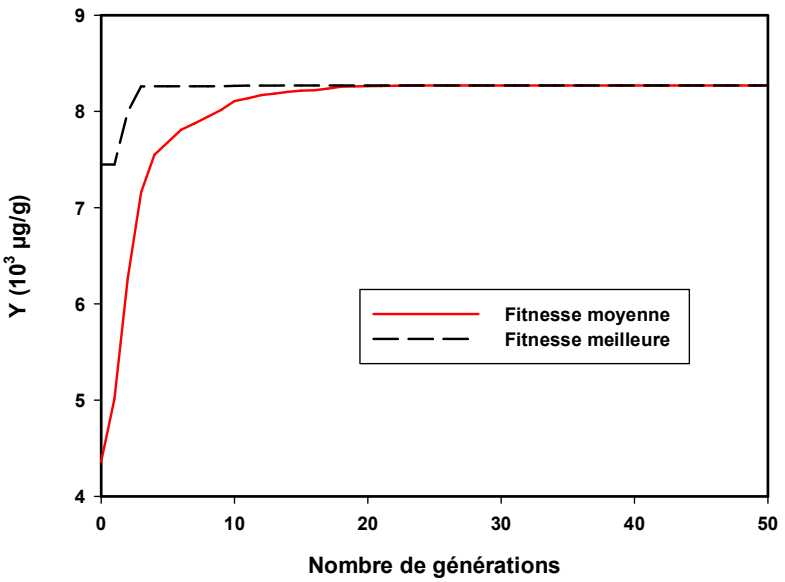


Figure IV.12. Evolutions de la fonction de fitness meilleure et moyenne au cours de la méthodologie "RNRB-OEP".

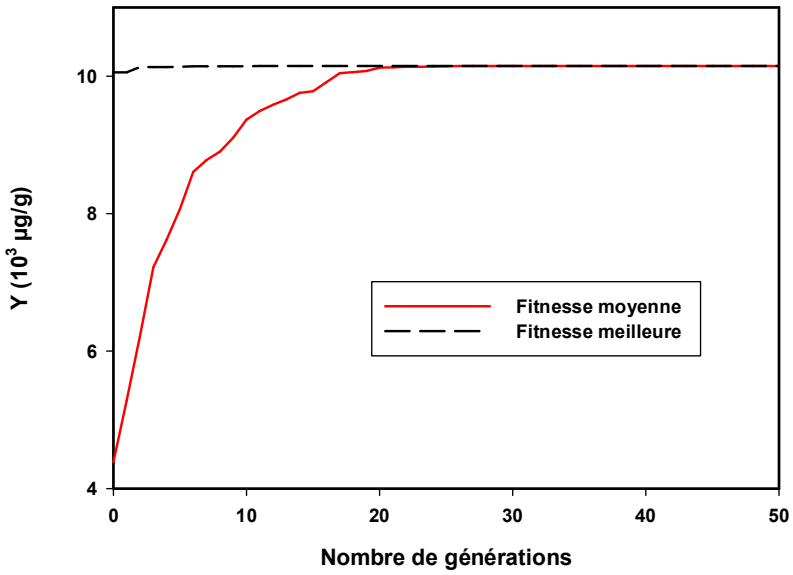
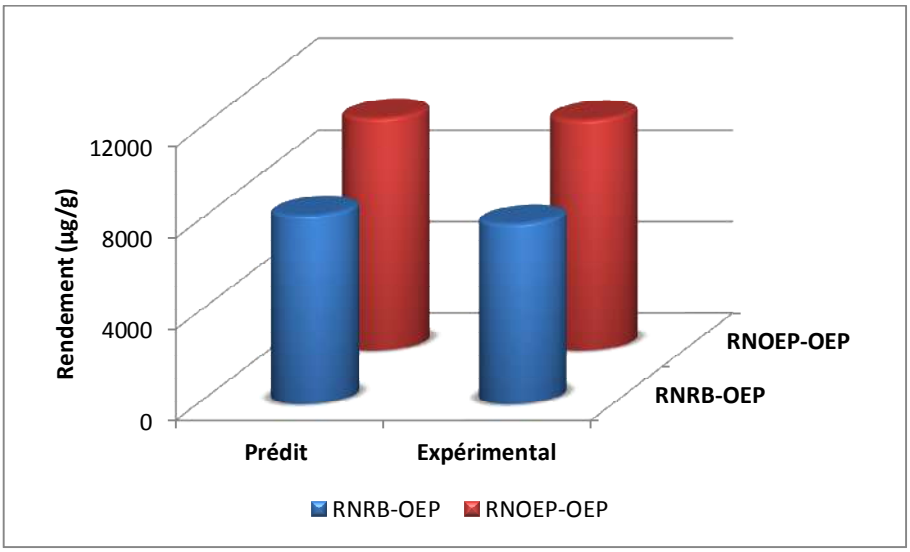


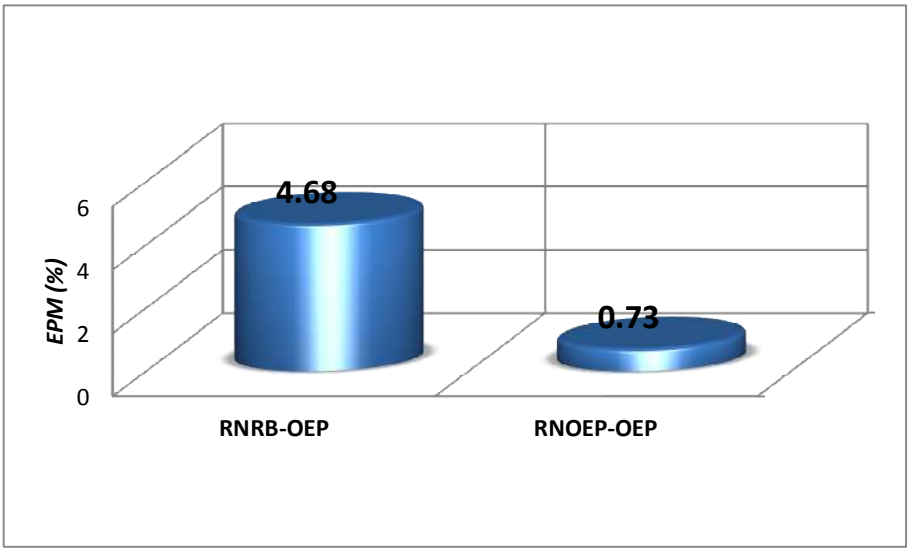
Figure IV.13. Evolutions de la fonction de fitness meilleure et moyenne au cours de la méthodologie hybride "RNOEP-OEP".

Tableau IV.13. Résultats de l'optimisation.

Méthodologie	**Conditions de culture**			**Réponse prédite (µg/g)**	**Réponse expérimentale (µg/g)**	***EPM* (%)**
	X1	***X2***	***X3***			
RNRB-OEP	258	26.78	5.98	8270	7900 ± 200	4.68%
RNOEP-OEP	242	26.88	6.06	10149	10074 ± 500	0.73%



(a)



(b)

Figure IV.14. Comparaison entre les deux méthodologies (RNOEP-OEP et RNRB-OEP) : **(a)** en termes de rendement prédit et expérimental, **(b)** en termes de l'*EPM*.

IV.6. Comparaison indirecte avec d'autres méthodologies d'optimisation à base des RN

Nous avons opté pour une comparaison indirect de fourchette de la méthodologie hybride proposée dans ce chapitre (RNOEP-OEP) avec d'autres méthodologies d'optimisation à base des RN les plus utilisés en bioprocédés et plus particulièrement l'optimisation des procédés de fermentation. Le tableau IV.14 donne une description de ces différentes méthodologies d'optimisation et le tableau IV.15 donne une comparaison des résultats de ces méthodologies d'optimisation en termes des erreurs obtenues lors des phases de modélisation et d'optimisation. Pour encore mieux illustrer cette comparaison nous avons présenté ces résultats sous formes de deux histogrammes, un illustrant les erreurs obtenues lors des phases de modélisation (Figure IV.15) et un autre représentant les erreurs obtenues lors des phases d'optimisation (Figure IV.16). Pour les méthodologies qui font appel à l'algorithme d'OEP (RN-OEP), notre modèle neuronale (Khaouane *et al.* 2012a) donne une *ESP* et une *EPM* plus élevées que les modèles de (Huang *et al.* 2007) et de (Liu *et al.* 2009) (Figure IV.15), d'où son optimisation a également donné une erreur plus élevée que les deux autres (Figure IV.16). Pour les méthodologies qui font appel à l'algorithme génétique (RN-AG), le modèle neuronale de (Singh *et al.* 2009) donne le % d'erreur le plus faible par rapport aux deux modèles de Zafar (Zafar *et al.* 2012a; Zafar *et al.* 2012b) (Figures IV.15), d'où son optimisation a également donné une erreur plus faible (Figure IV.16). Ces résultats prouvent que la performance de l'optimisation des équations mathématiques issues des modèles neuronaux est strictement liée à la capacité de prédiction et de généralisation de ces derniers. En effet, plus le modèle neuronal est performant plus son optimisation donne des résultats plus précis.

La figure IV.16 indique aussi que l'optimisation par l'algorithme d'OEP est plus performante que l'optimisation par l'AG avec des erreurs inférieures à 1%. On peut dire que l'algorithme d'OEP constitue une alternative intéressante à l'algorithme génétique dans l'optimisation des modèles neuronaux, c'est ce qui justifie notre choix de l'utiliser au cours de la phase d'optimisation dans la méthodologie hybride proposée dans ce chapitre. Cependant, l'inconvénient majeur qui accompagne l'utilisation de l'algorithme d'OEP est le réglage des ses paramètres qui s'avère plus de l'art que de la science.

Tableau IV.14. Description de différentes méthodologies d'optimisation à base des RN dans la littérature.

Référence	Produit	Souche	Variables indépendantes	Variables dépendantes	Méthodologie d'optimisation à base des RN			
					plan d'expérience	Réseau de neurones	Algorithme d'optimisation	Notation
(Huang *et al.* 2007)	Acide-Gamma-Aminobutyrique (GABA)	*Lactobacillus brevis* (catalyseur)	*pH*, Température et $FeSO_4$. 7H2O	Rendement de GABA	Plan Composite Centré (PCC)	RNFF	Algorithme d'optimisation par essaim de particules (OEP)	RN-OEP
(Liu *et al.* 2009)	Acide hyaluronique	*Streptococcus zooepidemicus*	Vitesse d'agitation, taux d'aération et le nombre d'agitateur	Rendement de production	Box-behnken	Réseau à fonction radiale (RFR)	Algorithme d'optimisation par essaim de particules (OEP)	RN-OEP
Notre résultat	Pleuromutiline	*Pleurotus mutilus*	Vitesse d'agitation, Température et *PH*	Rendement en produit	Plan Composite Centré (PCC)	RNFF	Algorithme d'optimisation par essaim de particules (OEP)	RNOEP-OEP
(Singh *et al.* 2009)	Actinomycine V	*Streptomyces triostinicus*	MgSO4, NaCl, glucose, la farine de soja et $CaCO_3$	Rendement de l'antibiotique	Plan Composite Centré (PCC)	RNFF	Algorithme génétique (AG)	RN-AG
(Zafar *et al.* 2012a)	Poly (3-hydroxybutyrate-co-3-hydroxyvalérate)	*Azohydromonas lata* MTCC 2311	Can molases, urée et acide propionique	Biomasse et produit	Box-behnken	RNFF	Algorithme génétique (AG)	RN-AG
(Zafar *et al.* 2012b)	Polyhydroxybutyrate	*Azohydromonas lata* MTCC 2311	Le saccharose, l'urée et la solution de TE	Biomasse et Produit	Plan Composite Centré (PCC)	RNFF	Algorithme génétique (AG)	RN-AG

Tableau IV.15. Comparaison des résultats des différentes méthodologies d'optimisation à base des RN dans la littérature.

Référence	Notation	Dimension de la base de données	Architecture	Phase de modélisation neuronale		Phase d'optimisation
				ESP (%)	*EPM (%)*	*ESP (%) = EPM (%)*
(Huang *et al.* 2007)	RN-OEP	21	RNFF : 3-4-1	0.08	0.05	0.47
(Liu *et al.* 2009)	RN-OEP	20	RFR : 3-1	2.04	1.96	0.72
Notre résultat	RNOEP-OEP	20	RNFF : 3-4-1	2.53	2.7	0.73
(Singh *et al.* 2009)	RN-AG	36	RNFF : 5-5-1	2.59	1.21	2.03
(Zafar *et al.* 2012a)	RN-AG	17	RNFF : 3-4-1	5.31	1.74	2.08
(Zafar *et al.* 2012b)	RN-AG	20	RNFF : 3-4-4	2.84	2.14	2.59

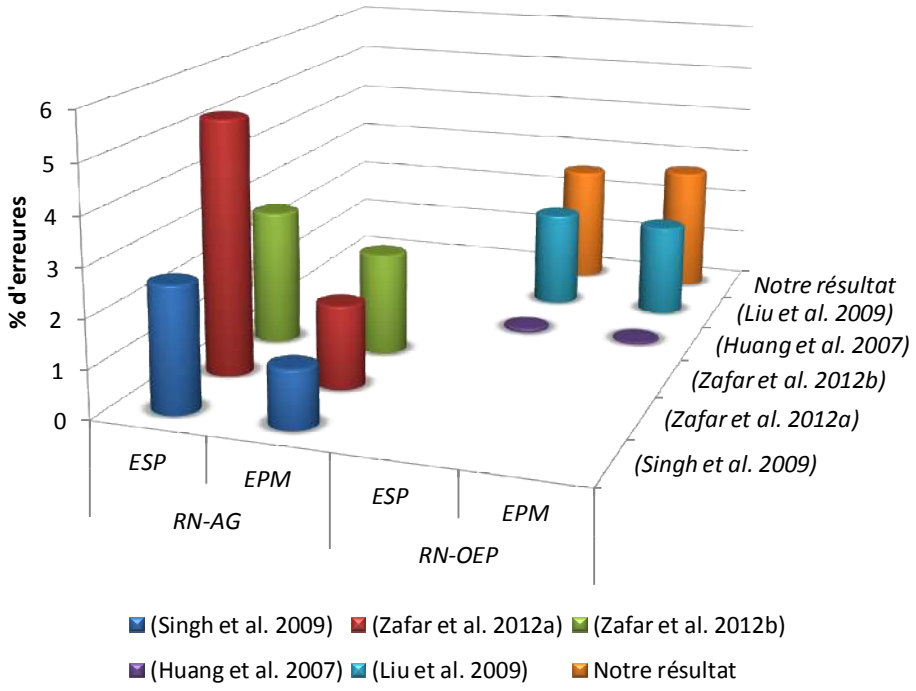


Figure IV.15. Comparaison, en termes des % d'erreurs (*ESP* et *EPM*) de la phase de modélisation, de différentes méthodologies d'optimisation à base des RN dans la littérature.

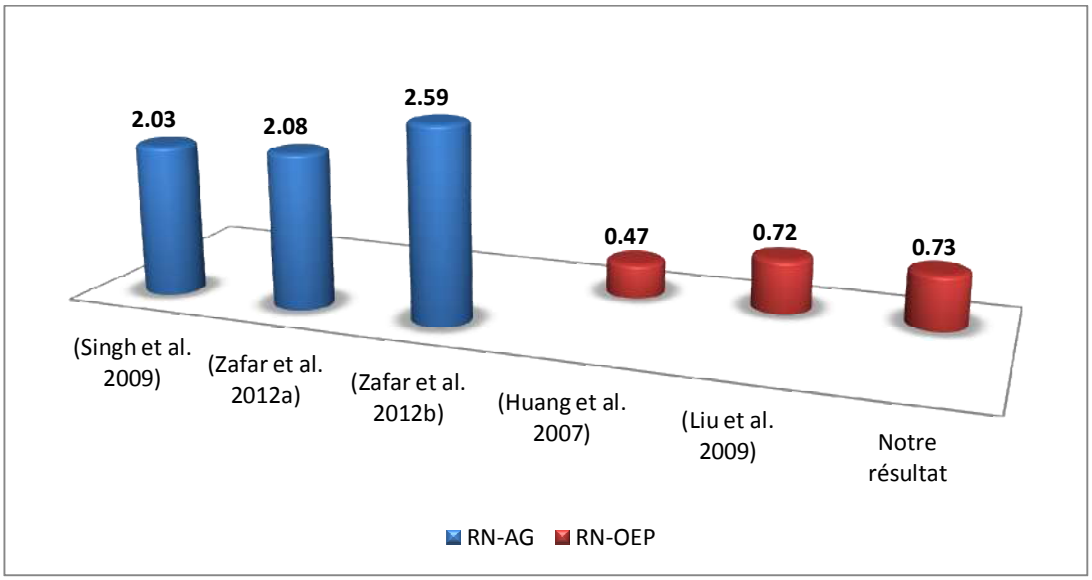


Figure IV.16. Comparaison, en termes de l'*EPM* de la phase d'optimisation, des différentes méthodologies d'optimisation à base des RN dans la littérature.

IV.7. Conclusion

Dans ce chapitre on a développé une méthodologie hybride, incluant un plan composite centré (PCC), un réseau de neurones artificiel (RN) et un algorithme d'optimisation par essaim de particules (OEP) et ceux afin d'optimisé les conditions de culture de *Pleurotus mutilus* (vitesse d'agitation, température et *pH* initial). Le PCC et le RN ont été utilisés respectivement pour la planification des expériences et la modélisation du processus de fermentation. L'algorithme d'OEP a été utilisé pour deux fins : le remplacement de la rétro-propagation standard dans l'apprentissage du RN (RNOEP) et l'optimisation du processus (RNOEP-OEP).

En vertu de cette approche hybride (RNOEP-OEP), les niveaux optimaux des conditions de culture sont : une vitesse d'agitation de 242 tr/min, une température de 26,88 °C et un *pH* de 6.06. Une production de 10074 ± 500 µg/g qui est en très bon accord avec la prédiction (10149 µg/g), a été observée dans l'expérience de vérification.

La combinaison du RNOEP avec l'algorithme générique d'OEP a une bonne prévisibilité et une bonne précision dans l'optimisation du bio-processus. Cette approche hybride est suffisamment générale et peut ainsi être très utile pour modéliser et optimiser d'autres bioprocédés industriels.

Conclusion générale

et perspectives

Les réseaux de neurones possèdent une propriété remarquable dont l'apport est manifesté pour la modélisation des processus non linéaires : ce sont des approximateurs universels parcimonieux. Les procédés biologiques se différencient des autres procédés par la variabilité de leur comportement dynamique générée par la présence des organismes vivants. Compte tenu de la complexité de ces procédés, liée à leur nature non linéaire et non stationnaire, les techniques neuronales présentent un intérêt accru dans la modélisation de ces procédés. L'objectif de cette thèse visait à développer des méthodologies basées sur les réseaux de neurones pour la modélisation de la fermentation de pleuromutiline et ceux pour un but de prévision et d'optimisation.

Notre première contribution a porté sur l'utilisation des réseaux de neurones artificiels pour la modélisation de la biosynthèse de pleuromutiline produite par la souche *Pleurotus mutilus* en mode fed-batch. Pour cela, trois modèles neuronaux de type feedforward sont construits et optimisés dans le but de prédire l'évolution des paramètres difficilement mesurables on line, à savoir les concentrations de la biomasse, du substrat et du produit, à partir de variables facilement mesurables dans la fermentation fed-batch de pleuromutiline. Les trois modèles neuronaux sont caractérisés par une structure similaire : cinq neurones dans la couche d'entrée (le temps de fermentation, le *pH*, l'indice de réfraction, la concentration initiale d'inoculum et la concentration initiale en glucose), une couche cachée avec un nombre de neurones optimisé au cours du processus d'apprentissage et un neurone dans la couche de sortie (biomasse, consommation du sucre ou formation du produit). Une bonne concordance, entre les valeurs expérimentales et les valeurs prédites, pour chaque modèle neuronal a été obtenue (les racines des erreurs quadratiques moyennes pour la base de test étaient de : 0,4624 % pour le modèle décrivant la croissance cellulaire ; 0,1234 g/L pour le modèle décrivant l'évolution de la consommation du substrat et 0,0016 mg/g pour le modèle décrivant la formation du produit). Par ailleurs, la comparaison entre les modèles neuronaux optimisés et les modèles cinétiques non structurés, en termes de prédiction et de simulation, a montré la supériorité des modèles neuronaux. En plus de l'excellente corrélation des données expérimentales, les trois modèles neuronaux utilisés ont montré une bonne capacité prédictive d'interpolation avec des coefficients de corrélations approchant l'unité. D'autre part, l'analyse de sensibilité appliquée aux trois réseaux optimisés a démontré l'importance des cinq paramètres choisis pour modéliser la cinétique de fermentation de pleuromutiline.

Notre deuxième contribution s'est basée sur le développement d'une méthodologie hybride d'optimisation, incluant un plan composite centré (PCC), un réseau de neurones artificiel (RN) et un algorithme d'optimisation par essaim de particules (OEP). Cette technique a été appliquée à l'optimisation des conditions de culture de *Pleurotus mutilus* (vitesse d'agitation, température et *pH*

initial) lors de la production batch de pleuromutiline et ceux dans le but d'obtenir une production maximale en produit. Le PCC a été utilisé pour planifier et concevoir l'expérience afin d'obtenir des données expérimentales statistiquement bien réparties dans le domaine d'entrée, le RN a été utilisé pour modéliser le processus non linéaire et ainsi obtenir une fonction mathématique reliant le rendement en pleuromutiline avec les trois facteurs. L'algorithme d'OEP a été utilisé pour deux fins: le remplacement de la rétro-propagation standard dans l'apprentissage du RN (RNOEP) et l'optimisation du modèle neuronal obtenu (RNOEP-OEP). La comparaison, en termes de modélisation, du modèle neuronal formé avec l'algorithme d'OEP (RNOEP) avec le modèle quadratique de la méthodologie des surfaces de réponse (MSR) et avec le réseau de neurones formé avec l'algorithme de régularisation bayésienne (RNRB), a démontré la grande capacité de modélisation des modèles neuronaux (RNOEP et RNRB). Par la suite, un algorithme générique d'OEP a été utilisé pour optimiser l'espace d'entrées des deux modèles neuronaux à savoir le RNOEP (RNOEP-OEP) et le RNRB (RNRB-OEP). En vertu de l'approche hybride "RNOEP-OEP", les niveaux optimaux des conditions de culture obtenues sont : 242 tr/min pour la vitesse d'agitation, une température de 26,88 °C et un *pH* de 6.06. Une production de 10074 ± 500 µg/g qui est en très bon accord avec la prédiction (10149 µg/g), a été observée dans l'expérience de vérification. L'approche "RNRB-OEP" prédit une production maximale de 8270 µg/g à une vitesse d'agitation de 258 tr/min, un niveau de température de 26.78 °C et un *pH* initial de 5.98. Sous ces conditions, une production de 7900 µg/g a été obtenue dans l'expérience de validation. L'approche hybride "RNOEP-OEP" (10074 µg/g) a donné un rendement de 27,5 % supérieur à celui obtenu par l'approche "RNRB-OEP" (7900 µg/g) démontrant ainsi la supériorité de la méthodologie hybride proposée.

A la lumière des résultats obtenus on peut tirer les conclusions suivantes :

- Les réseaux de neurones sont des outils statistiques, qui permettent d'ajuster des fonctions non linéaires très générales à des ensembles de points comme toute méthode statistique ;
- Les réseaux de neurones fonctionnent comme une boite noire et les paramètres du réseau (poids et biais) n'ont pas de signification physique. L'application des réseaux neuronaux ne développe pas directement de la théorie de description du système, mais rend faisable la modélisation et la simulation des procédés de fermentation ;
- Dans la modélisation cinétique, les valeurs des paramètres du modèle ont une signification physique, contiennent toujours les conséquences des suppositions

introduites et des simplifications du fait de la complexité du cycle de fermentation, alors que les modèles neuronaux sont libres de ces dernières ;

- La modélisation par réseau de neurones est une méthode qui est basée seulement sur des données expérimentales et/ou semi-expérimentales (déduites des modèles analytiques, empiriques, numériques, …), donc la qualité et la quantité de données décident de l'exactitude et la robustesse du modèle ;

- Les réseaux de neurones peuvent être très robustes et performants même en présence de peu de données expérimentales à condition que ces dernières soient statistiquement bien réparties dans le domaine d'entrée du modèle neuronal, ce qui est le cas avec le plan composite centré ;

- Les formules mathématiques extraites des modèles neuronaux peuvent trouver des applications intéressantes dans la prédiction, l'optimisation et même le contrôle des procédés de fermentation ;

- La combinaison des réseaux de neurones avec l'algorithme d'OEP présente une bonne prévisibilité et une bonne précision dans la prédiction et l'optimisation des bio-processus ;

- L'approche "hybride" proposée pour l'optimisation des conditions de culture de *Pleurotus mutilus* est suffisamment générale et peut ainsi être très utile pour optimiser d'autres bioprocédés ;

- La performance de l'optimisation des équations mathématiques, issues des modèles neuronaux, est strictement liée à la capacité de prédiction et de généralisation des ces derniers. En effet plus le modèle neuronal est performant plus son optimisation donne des résultats plus précis. Pour augmenter plus cette précision, le choix de l'algorithme d'optimisation le plus performant est nécessaire ;

- L'algorithme d'OEP peut constituer une alternative intéressante non seulement aux algorithmes conventionnelles d'apprentissage des réseaux de neurones, mais aussi à l'algorithme génétique dans l'optimisation des modèles neuronaux, c'est ce qui justifie notre choix de l'utilisé dans la méthodologie hybride proposée. Cependant, l'inconvénient majeur qui accompagne l'utilisation de l'algorithme d'OEP est le choix de la variante et le réglage des ses paramètres qui relèvent plus de l'art que de la science ;

- La modélisation neuronale peut servir comme une alternative intéressante et fiable aux modèles cinétiques dans la modélisation des procédés de fermentation.

- Compte tenu qu'il n'y a pas de règle pour le choix du modèle neuronal pour un problème donné, la stratégie d'essai et erreur devient parfois laborieuse.

De ce travail de thèse ont émergé de nombreuses pistes de recherche qui feront l'objet de futurs travaux. Les perspectives concernent plus particulièrement la modélisation d'autre bio-processus pour des fins de prédiction, d'optimisation et de contrôle. Nous pouvons aussi étendre l'idée de ce travail à d'autres techniques récentes de modélisation et d'optimisation telle que les approches de modélisation types "boite grise" et d'autres méta-heuristiques d'optimisation (simulated annealing, bee colony optimization, ….).

Références bibliographiques

Ammar M. Y. (2007). "Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu." *Thèse de Doctorat*, Institut National Polytechnique de Toulouse. Toulouse, France.

Andersson M. et Adlercreutz P. (1999). "Evaluation of simple enzyme kinetics by response surface modelling." *Biotechnology Techniques* **13**(12): 903-907.

Asensio-Cuesta S., Diego-Mas J. A. et Alcaide-Marzal J. (2010). "Applying generalised feedforward neural networks to classifying industrial jobs in terms of risk of low back disorders." *International Journal of Industrial Ergonomics* **40**(6): 629-635.

Bachinger T. et Mandenius C. F. (2000). "Searching for process information in the aroma of cell cultures." *Trends in Biotechnology* **18**(12): 494-500.

Banik R. M., Santhiagu A. et Upadhyay S. N. (2007). "Optimization of nutrients for gellan gum production by *Sphingomonas paucimobilis* ATCC-31461 in molasses based medium using response surface methodology." *Bioresource Technology* **98**(4): 792-797.

Baş D. et Boyacı İ. H. (2007). "Modeling and optimization II: Comparison of estimation capabilities of response surface methodology with artificial neural networks in a biochemical reaction." *Journal of Food Engineering* **78**(3): 846-854.

Basheer I. A. et Hajmeer M. (2000). "Artificial neural networks: Fundamentals, computing, design, and application." *Journal of Microbiological Methods* **43**(1): 3-31.

Baughman D. R. et Liu Y. A. (1995). "Neural networks in bioprocessing and chemical engineering." Academic Press, Florida, USA.

Becker T., Enders T. et Delgado A. (2002). "Dynamic neural networks as a tool for the online optimization of industrial fermentation." *Bioprocess and Biosystems Engineering* **24**(6): 347-354.

Beg Q. K., Saxena R. K. et Gupta R. (2002). "Kinetic constants determination for an alkaline protease from *Bacillus mojavensis* using response surface methodology." *Biotechnology and Bioengineering* **78**(3): 289-295.

Benkortbi O. (2008). "Etude de la fermentation des antibiotiques : Application à la pleuromutiline." *Thèse de Doctorat*, Université d'USTHB. Alger, Algérie.

Benkortbi O., Hanini S. et Bentahar F. (2007). "Batch kinetics and modelling of pleuromutilin production by *Pleurotus mutilis*." *Biochemical Engineering Journal* **36**(1): 14-18.

Bermejo S. et Cabestany J. (1999). "On-line gradient learning algorithms for K-nearest neighbor classifiers." Dans le livre: "Foundations and tools for neural modeling." Springer Berlin Heidelberg. **1606:** 546-555.

Bermejo S. et Cabestany J. (2001). "Learning with nearest neighbour classifiers." *Neural Processing Letters* **13**(2): 159-181.

Bishop C. M. (1995). "Neural networks for pattern recognition." Oxford University Press, New York, USA.

Bryjak J., Ciesielski K. et Zbicinski I. (2004). "Modelling of glucoamylase thermal inactivation in the presence of starch by artificial neural network." *Journal of Biotechnology* **114**(1-2): 177-185.

Buchanan R. L. et Bagi L. K. (1994). "Expansion of response surface models for the growth of *Escherichia coli* O157 : H7 to include sodium nitrite as a variable." *International Journal of Food Microbiology* **23**(3-4): 317-332.

Burkert C. A. V., Barbosa G. N. O., Mazutti M. A. et Maugeri F. (2011). "Mathematical modeling and experimental breakthrough curves of cephalosporin C adsorption in a fixed-bed column." *Process Biochemistry* **46**(6): 1270-1277.

Burkert J. F. M., Maugeri F. et Rodrigues M. I. (2004). "Optimization of extracellular lipase production by *Geotrichum sp.* using factorial design." *Bioresource Technology* **91**(1): 77-84.

Carpenter G. A. et Grossberg S. (1987). "ART 2: Self-organization of stable category recognition codes for analog input patterns." *Applied optics* **26**(23): 4919-30.

Carpenter G. A. et Grossberg S. (1988). "The ART of adaptive pattern recognition by a self-organizing neural network." *Computer* **21**(3): 77-88.

Castro-Martinez C. (2007). "*Brettanomyces bruxellensis* : Etude métabolique, cinétique et modélisation. Influence des facteurs environnementaux." *Thèse de Doctorat*, Institut National Polytechnique de Toulouse. Toulouse, France.

Chamekh A. (2007). "Optimisation des procédés de mise en forme par les réseaux de neurones artificiels." *Thèse de Doctorat*, Université d'Angers. Angers, France.

Chau K. W. (2006). "Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River." *Journal of Hydrology* **329**(3-4): 363-367.

Clerc M. et Kennedy J. (2002). "The particle swarm - explosion, stability, and convergence in a multidimensional complex space." *IEEE Transactions on Evolutionary Computation* **6**(1): 58-73.

Cockshott A. R. et Hartman B. E. (2001). "Improving the fermentation medium for Echinocandin B production part II: Particle swarm optimization." *Process Biochemistry* **36**(7): 661-669.

Conseil de l'Europe (2001). Pharmacopée européenne. Strasbourg, France, Direction Européenne de la Qualité du Médicament & des Soins de Santé.

Cooren Y. (2008). "Perfectionnement d'un algorithme adaptatif d'optimisation par essaim particulaire. Applications en génie médical et en électronique." *Thèse de Doctorat*, Université de Paris 12. Val de Marne, France.

Cruz A. J. G., Silva A. S., Araujo M. L. G. C., Giordano R. C. et Hokka C. O. (1999). "Modelling and optimization of the cephalosporin C production bioprocess in a fed-batch bioreactor with invert sugar as substrate." *Chemical Engineering Science* **54**(15-16): 3137-3142.

Cutlip M. B. et Shacham M. (2008). "Problem solving in chemical and biochemical engineering with POLYMATH, Excel, and MATLAB." Prentice Hall, New Jersey, USA.

Dames M. (2005). "Gestion de procèdes discontinus : Méthodologie de modélisation et d'optimisation d'opérations de dispersion liquide-liquide en cuve agitée." *Thèse de Doctorat*, Institut National Polytechnique de Toulouse. Toulouse, France.

Danaher S., Datta S., Waddle I. et Hackney P. (2004). "Erosion modelling using bayesian regulated artificial neural networks." *Wear* **256**(9-10): 879-888.

Davalo É. et Naïm P. (1989). "Des réseaux de neurones." Eyrolles, Paris, France.

Demuth H. et Beale M. (2000). Neural network toolbox for use with Matlab. User's Guide version 4. USA, The MathWorks Inc.

Desai K. M., Akolkar S. K., Badhe Y. P., Tambe S. S. et Lele S. S. (2006). "Optimization of fermentation media for exopolysaccharide production from *Lactobacillus plantarum* using artificial intelligence-based techniques." *Process Biochemistry* **41**(8): 1842-1848.

Desai K. M., Survase S. A., Saudagar P. S., Lele S. S. et Singhal R. S. (2008). "Comparison of artificial neural network (ANN) and response surface methodology (RSM) in fermentation media optimization: Case study of fermentative production of scleroglucan." *Biochemical Engineering Journal* **41**(3): 266-273.

Di Massimo C., Montague G. A., Willis M. J., Tham M. T. et Morris A. J. (1992). "Towards improved penicillin fermentation via artificial neural networks." *Computers & Chemical Engineering* **16**(4): 283-291.

Dorigo M. et Caro G. D. (1999). "The ant colony optimization meta-heuristic." Dans le livre: "New ideas in optimization." McGraw-Hill Ltd., UK**:** 11-32.

Dragoi E. N., Curteanu S., Galaction A. I. et Cascaval D. (2013). "Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process." *Applied Soft Computing* **13**(1): 222-238.

Dreyfus G. (2008). "Réseaux de neurones: Méthodologie et applications." Eyrolles, Paris, France.

Ebbesen S., Kiwitz P. et Guzzella L. (2011). "A generic particle swarm optimization Matlab function." Consulté le 1 décembre 2011, à partir de: http://www.idsc.ethz.ch/Downloads/pso/index.

Eberhart R. C. et Kennedy J. (1995). "A new optimizer using particles swarm theory." *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, IEEE Computer Society. 4-6 octobre 1995.

Eberhart R. C., Shi Y. et Kennedy J. (2001). "Swarm intelligence." Morgan Kaufmann, San Francisco, USA.

Eberhart R. C., Simpson P. K., Dobbins R. et Dobbins R. W. (1996). "Computational intelligence PC tools." AP Professional, Boston, USA.

Egger H. et Reinshagen H. (1976). "New pleuromutilin derivatives with enhanced antimicrobial activity I. Synthesis." *Journal of Antibiotics* **29**(9): 915-922.

Fan H. Y. et Shi Y. (2001). "Study on Vmax of particle swarm optimization." *Proceedings of the Workshop on Particle Swarm Optimization*, Purdue school of engineering and technology, Indianapolis. Avril 2001.

Feyo de Azevedo S., Dahm B. et Oliveira F. R. (1997). "Hybrid modelling of biochemical processes: A comparison with the conventional approach." *Computers & Chemical Engineering* **21, Supplement**(0): S751-S756.

Fiorentin L. D., Trigueros D. E. G., Modenes A. N., Espinoza-Quinones F. R., Pereira N. C., Barros S. T. D. et Santos O. A. A. (2010). "Biosorption of reactive blue 5G dye onto drying orange bagasse in batch system: Kinetic and equilibrium modeling." *Chemical Engineering Journal* **163**(1-2): 68-77.

Fontaine J. L. et Germain A. (2001). "Model-based neural networks." *Computers and Chemical Engineering* **25**(7): 1045-1054.

Galván I. M., Zaldívar J. M., Hernández H. et Molga E. (1996). "The use of neural networks for fitting complex kinetic data." *Computers & Chemical Engineering* **20**(12): 1451-1465.

Gao H., Liu M., Liu J., Dai H., Zhou X., Liu X., Zhuo Y., Zhang W. et Zhang L. (2009). "Medium optimization for the production of avermectin B1a by *Streptomyces avermitilis* 14-12A using response surface methodology." *Bioresource Technology* **100**(17): 4012-4016.

Garson G. D. (1991). "Interpreting neural-network connection weights." *Artificial Intelligence Expert* **6**(4): 46-51.

Gevrey M., Dimopoulos I. et Lek S. (2003). "Review and comparison of methods to study the contribution of variables in artificial neural network models." *Ecological Modelling* **160**(3): 249-264.

Glassey J., Ignova M., Ward A. C., Montague G. A. et Morris A. J. (1997). "Bioprocess supervision: Neural networks and knowledge based systems." *Journal of Biotechnology* **52**(3): 201-205.

Goh A. T. C. (1995). "Back-propagation neural networks for modeling complex systems." *Artificial Intelligence in Engineering* **9**(3): 143-151.

Gurunathan B. et Sahadevan R. (2011). "Design of experiments and artificial neural network linked genetic algorithm for modeling and optimization of L-asparaginase production by *Aspergillus terreus* MTCC 1782." *Biotechnology and Bioprocess Engineering* **16**(1): 50-58.

Hagan M. T., Demuth H. B. et Beale M. H. (1996). "Neural network design." PWS Publishing Company, Boston, USA.

Haider M., Pakshirajan K., Singh A. et Chaudhry S. (2008). "Artificial neural network-genetic algorithm approach to optimize media constituents for enhancing lipase production by a soil microorganism." *Applied Biochemistry and Biotechnology* **144**(3): 225-235.

Hannan P. C. T., Windsor H. M. et Ripley P. H. (1997). "In vitro susceptibilities of recent field isolates of Mycoplasma hyopneumoniae and Mycoplasma hyosynoviae to valnemulin (Econor®), tiamulin and enrofloxacin and the in vitro development of resistance to certain antimicrobial agents in Mycoplasma hyopneumoniae." *Research in Veterinary Science* **63**(2): 157-160.

Haridas N. et Devi.M N. (2010). "PSO learning on artificial neural networks." *Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India*, Coimbatore, India, ACM.

Hassoun M. H. (1995). "Fundamentals of artificial neural networks." Mit Press, Massachusetts, USA.

Haykin S. S. (1999). "Neural networks: A comprehensive foundation." Prentice Hall, New Jersey, USA.

Heaton J. (2005). "Introduction to neural networks with Java." Heaton Research, Inc., Chesterfield, UK.

Hebb D. O. (1949). "The organization of behavior: A neuropsychological theory." Wiley, New York, USA.

Hirokawa Y., Kinoshita H., Tanaka T., Nakamura T., Fujimoto K., Kashimoto S., Kojima T. et Kato S. (2008). "Pleuromutilin derivatives having a purine ring. Part 1: New compounds with promising antibacterial activity against resistant Gram-positive pathogens." *Bioorganic & Medicinal Chemistry Letters* **18**(12): 3556–3561.

Hirschen K. et Schafer M. (2006). "Bayesian regularization neural networks for optimizing fluid flow processes." *Computer Methods in Applied Mechanics and Engineering* **195**(7-8): 481-500.

Hopfield J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences* **79**(8): 2554-2558.

Hornik K. (1991). "Approximation capabilities of multilayer feedforward networks." *Neural Networks* **4**(2): 251-257.

Huang J., Mei L. H. et Xia J. (2007). "Application of artificial neural network coupling particle swarm optimization algorithm to biocatalytic production of GABA." *Biotechnology and Bioengineering* **96**(5): 924-931.

Huang J., Shimizu H. et Shioya S. (2002). "Data preprocessing and output evaluation of an autoassociative neural network model for online fault detection in virginiamycin production." *Journal of Bioscience and Bioengineering* **94**(1): 70-77.

Ignova M., Montague G. A., Ward A. C. et Glassey J. (1999). "Fermentation seed quality analysis with self-organising neural networks." *Biotechnology and Bioengineering* **64**(1): 82-91.

Jin S., Ye K., Shimizu K. et Nikawa J. (1996). "Application of artificial neural network and fuzzy control for fed-batch cultivation of recombinant *Saccharomyces cerevisiae*." *Journal of Fermentation and Bioengineering* **81**(5): 412-421.

Kalil S. J., Maugeri F. et Rodrigues M. I. (2000). "Response surface analysis and simulation as a tool for bioprocess design and optimization." *Process Biochemistry* **35**(6): 539-550.

Kang S. G., Lee D. H., Ward A. C. et Lee K. J. (1998). "Rapid and quantitative analysis of clavulanic acid production by the combination of pyrolysis mass spectrometry and artificial neural network." *Journal of microbiology and biotechnology* **8**(5): 523-530.

Karim M. N., Yoshida T., Rivera S. L., Saucedo V. M., Eikens B. et Oh G. S. (1997). "Global and local neural network models in biotechnology: Application to different cultivation processes." *Journal of Fermentation and Bioengineering* **83**(1): 1-11.

Kavanagh F., Hervey A. et Robbins W. J. (1951). "Antibiotic substances from basidiomycetes. VIII. *Pleurotus mutilus (Fr.) Sacc.* and *Pleurotus passeckerianus Pilat*." *Proceedings of the National Academy of Sciences of the United States of America* **37**(9): 570-574.

Keedwell E. et Narayanan A. (2005). "Intelligent bioinformatics: The application of artificial intelligence techniques to bioinformatics problems." Wiley, Chichester, UK.

Kennedy J. (1999). "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance." *IEEE Congress on Evolutionary Computation. CEC 99*. 1999.

Kennedy M., Prapulla S. G. et Thakur M. S. (1992). "Designing fermentation media: A comparison of neural networks to factorial design." *Biotechnology Techniques* **6**(4): 293-298.

Kennedy M. et Spooner N. (1996). "Using fuzzy logic to design fermentation media: A comparison to neural networks and factorial design." *Biotechnology Techniques* **10**(1): 47-52.

Khaouane L., Benkortbi O., Hanini S. et Si-Moussa C. (2013). "Modeling of an industrial process of pleuromutilin fermentation using feed-forward neural networks." *Brazilian Journal of Chemical Engineering* **30**(1): 105-116.

Khaouane L., Si-Moussa C., Hanini S. et Benkortbi O. (2012a). "Optimization of culture conditions for the production of pleuromutilin from *Pleurotus mutilus* using a hybrid method based on central composite design, neural network, and particle swarm optimization." *Biotechnology and Bioprocess Engineering* **17**(5): 1048-1054.

Khaouane L., Si-Moussa C., Hanini S. et Benkortbi O. (2012b). "Optimization of culture conditions for the production of pleuromutilin from *Pleurotus mutilus* using neural network and particle swarm optimization." *3ème Journée Scientifique sur le Génie des Procédés (JSGP3-2012)*, Blida, Algérie. 19 novembre 2012.

Khedam R. (2008). "Contribution au développement de méthodologies de fusion/classification contextuelles d'images satellitaires multi-sources : Application à la cartographie thématique du milieu urbain de la ville d'Alger." *Thèse de Doctorat*, Université d'USTHB. Alger, Algérie.

Kim H. O., Lim J. M., Joo J. H., Kim S. W., Hwang H. J., Choi J. W. et Yun J. W. (2005). "Optimization of submerged culture condition for the production of mycelial biomass and exopolysaccharides by *Agrocybe cylindracea*." *Bioresource Technology* **96**(10): 1175-1182.

Kohonen T. (1982). "Self-organized formation of topologically correct feature maps." *Biological Cybernetics* **43**(1): 59-69.

Kovarova-Kovar K., Gehlen S., Kunze A., Keller T., von Daniken R., Kolb M. et van Loon A. P. G. M. (2000). "Application of model-predictive control based on artificial neural networks to optimize the fed-batch process for riboflavin production." *Journal of Biotechnology* **79**(1): 39-52.

Kuhn R. C. (2010). "Purification of fructooligosaccharides obtained trhough enzymatic synthesis using fixed-bed columns and nanofiltration membranes." *Ph.D. Thesis*, University of Campinas. Campinas, Brazil.

Kumar S., Zafar M., Prajapati J. K., Kumar S. et Kannepalli S. (2011). "Modeling studies on simultaneous adsorption of phenol and resorcinol onto granular activated carbon from simulated aqueous solution." *Journal of Hazardous Materials* **185**(1): 287-294.

Laursen S., Webb D. et Ramirez W. F. (2007). "Dynamic hybrid neural network model of an industrial fed-batch fermentation process to produce foreign protein." *Computers & Chemical Engineering* **31**(3): 163-170.

Lazzús J. (2009). "Estimation of density as a function of temperature and pressure for imidazolium-based ionic liquids using a multilayer net with particle swarm optimization." *International Journal of Thermophysics* **30**(3): 883-909.

Lee D. S. et Park J. M. (1999). "Neural network modeling for on-line estimation of nutrient dynamics in a sequentially-operated batch reactor." *Journal of Biotechnology* **75**(2-3): 229-239.

Lek S., Belaud A., Baran P., Dimopoulos I. et Delacoste M. (1996a). "Role of some environmental variables in trout abundance models using neural networks." *Aquatic Living Resources* **9**(01): 23-29.

Lek S., Belaud A., Dimopoulos I., Lauga J. et Moreau J. (1995). "Improved estimation, using neural networks, of the food consumption of fish populations." *Marine and Freshwater Research* **46**(8): 1229-1236.

Lek S., Delacoste M., Baran P., Dimopoulos I., Lauga J. et Aulagnier S. (1996b). "Application of neural networks to modelling nonlinear relationships in ecology." *Ecological Modelling* **90**(1): 39-52.

Linko S., Luopa J. et Zhu Y. H. (1997). "Neural networks as `software sensors' in enzyme production." *Journal of Biotechnology* **52**(3): 257-266.

Linko S., Zhu Y. H. et Linko P. (1999). "Applying neural networks as software sensors for enzyme engineering." *Trends in Biotechnology* **17**(4): 155-162.

Liu L., Sun J., Zhang D., Du G., Chen J. et Xu W. (2009). "Culture conditions optimization of hyaluronic acid production by *Streptococcus zooepidemicus* based on radial basis function neural network and quantum-behaved particle swarm optimization algorithm." *Enzyme and Microbial Technology* **44**(1): 24-32.

Lu Z. M., Lei J. Y., Xu H. Y., Shi J. S. et Xu Z. H. (2011). "Optimization of fermentation medium for triterpenoid production from *Antrodia camphorata* ATCC 200183 using artificial intelligence-based techniques." *Applied Microbiology and Biotechnology* **92**(2): 371-379.

MacKay D. J. C. (1991). "Bayesian interpolation." *Neural Computation* **4**(3): 415-447.

Minsky M. L. et Papert S. (1969). "Perceptrons: An introduction to computational geometry." MIT Press, Massachusetts, USA.

Montague G. et Morris J. (1994). "Neural-network contributions in biotechnology." *Trends in Biotechnology* **12**(8): 312-324.

Moody J. et Darken C. J. (1989). "Fast learning in networks of locally-tuned processing units." *Neural computation* **1**(2): 281 - 294.

Moraes C. C., Mazutti M. A., Rodrigues M. I., Filho F. M. et Kalil S. J. (2009). "Mathematical modeling and simulation of inulinase adsorption in expanded bed column." *Journal of Chromatography A* **1216**(20): 4395-4401.

Nascimento C. A. O., Giudici R. et Guardani R. (2000). "Neural network based approach for optimization of industrial chemical processes." *Computers & Chemical Engineering* **24**(9-10): 2303-2314.

Nasr N., Hafez H., El Naggar M. H. et Nakhla G. (2013). "Application of artificial neural networks for modeling of biohydrogen production." *International Journal of Hydrogen Energy* **38**(8): 3189-3195.

Nelofer R., Ramanan R., Rahman R., Basri M. et Ariff A. (2012). "Comparison of the estimation capabilities of response surface methodology and artificial neural network for the optimization of recombinant lipase production by *E. coli* BL21." *Journal of Industrial Microbiology & Biotechnology* **39**(2): 243-254.

Nicoletti M. C. (2009). "Computational intelligence techniques for bioprocess modelling, supervision and control." Springer-Verlag, Düsseldorf, Germany.

Panagou E. Z. et Kodogiannis V. S. (2009). "Application of neural networks as a non-linear modelling technique in food mycology." *Expert Systems with Applications* **36**(1): 121-131.

Panigrahi B. K., Shi Y. et Lim M. H. (2011). "Handbook of swarm intelligence: Concepts, principles and applications." Springer-Verlag, Düsseldorf, Germany.

Patnaik P. R. (2001). "Penicillin fermentation: Mechanisms and models for industrial-scale bioreactors." *Critical Reviews in Microbiology* **27**(1): 25-39.

Paul G. C. et Thomas C. R. (1996). "A structured model for hyphal differentiation and penicillin production using *Penicillium chrysogenum*." *Biotechnology and Bioengineering* **51**(5): 558-72.

Pigram G. M. et MacDonald T. R. (2000). "Use of neural network models to predict industrial bioreactor effluent quality." *Environmental Science & Technology* **35**(1): 157-162.

Pintér J. D. (2006). "Global optimization: Scientific and engineering case studies." Springer-Verlag, Düsseldorf, Germany.

Pirt S. J. (1979). "Fed-batch culture of microbes." *Annals of the New York Academy of Sciences* **326**(1): 119-125.

Porto V. W., Saravanan N., Waagen D., Eiben A. E., Shi Y. et Eberhart R. (1998). "Parameter selection in particle swarm optimization." Dans le livre: "Evolutionary programming VII." Springer Berlin Heidelberg. **1447:** 591-600.

Potocnik P. et Grabec I. (1999). "Empirical modeling of antibiotic fermentation process using neural networks and genetic algorithms." *Mathematics and Computers in Simulation* **49**(4-5): 363-379.

Ratkowsky D. A. (1990). "Handbook of nonlinear regression models." Marcel Dekker Inc., New York, USA.

Reynolds C. W. (1987). "Flocks, herds and schools: A distributed behavioral model." *SIGGRAPH Computer Graphics* **21**(4): 25-34.

Rosa S. M., Soria M. A., Vélez C. G. et Galvagno M. A. (2010). "Improvement of a two-stage fermentation process for docosahexaenoic acid production by *Aurantiochytrium limacinum* SR21 applying statistical experimental designs and data analysis." *Bioresource Technology* **101**(7): 2367-2374.

Rosenblatt F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Reviews* **65**(6): 386-408.

Rumelhart D. E. et McClelland J. L. (1986). "Parallel distributed processing: Explorations in the microstructure of cognition. Foundations." Mit Press, Massachusetts, USA.

Saraceno A., Curcio S., Calabro V. et Iorio G. (2009). "A hybrid neural approach to model batch fermentation of "ricotta cheese whey" to ethanol." *Computers & Chemical Engineering* **34**(10): 1590-1596.

Schenker B. et Agarwal M. (1998). "Predictive control of a bench-scale chemical reactor based on neural-network models." *Control Systems Technology, IEEE Transactions on* **6**(3): 388-400.

Shee B. K., Vipsita S. et Rath S. K. (2011). "Protein feature classification using particle swarm optimization and artificial neural networks." *Proceedings of the 2011 International Conference on Communication, Computing & Security*, Rourkela, Odisha, India, ACM.

Shen Q., Shi W. M., Yang X. P. et Ye B. X. (2006). "Particle swarm algorithm trained neural network for QSAR studies of inhibitors of platelet-derived growth factor receptor phosphorylation." *European Journal of Pharmaceutical Sciences* **28**(5): 369-376.

Shene C., Diez C. et Bravo S. (1999). "Neural networks for the prediction of the state of *Zymomonas mobilis* CP4 batch fermentations." *Computers & Chemical Engineering* **23**(8): 1097-1108.

Shimizu H., Yasuoka K., Uchiyama K. et Shioya S. (1998). "Bioprocess fault detection by nonlinear multivariate analysis: Application of an artificial autoassociative neural network and wavelet filter bank." *Biotechnology Progress* **14**(1): 79-87.

Si-Moussa C. (2009). "Modélisation des équilibres de phases a pressions élevées par les réseaux de neurones artificiels." Ecole Nationale Supérieure Polytechnique. Alger, Algérie.

Si-Moussa C., Hanini S., Derriche R., Bouhedda M. et Bouzidi A. (2008). "Prediciton of high-pressure vapor liquid equilibrium of six binary systems, carbon dioxide with six esters, using an artificial neural network model." *Brazilian Journal of Chemical Engineering* **25**(1): 183-199.

Silva J. A., Costa Neto E. H., Adriano W. S., Ferreira A. L. O. et Gonçalves L. R. B. (2008). "Use of neural networks in the mathematical modelling of the enzymic synthesis of amoxicillin catalysed by penicillin G acylase immobilized in chitosan." *World Journal of Microbiology and Biotechnology* **24**(9): 1761-1767.

Silva R. G., Cruz A. J. G., Hokka C. O., Giordano R. L. C. et Giordano R. C. (2000). "A hybrid feedforward neural network model for the cephalosporin C production process." *Brazilian Journal of Chemical Engineering* **17**: 587-598.

Singh A., Majumder A. et Goyal A. (2008). "Artificial intelligence based optimization of exocellular glucansucrase production from *Leuconostoc dextranicum* NRRL B-1146." *Bioresource Technology* **99**(17): 8201-8206.

Singh V., Khan M., Khan S. et Tripathi C. (2009). "Optimization of actinomycin V production by *Streptomyces triostinicus* using artificial neural network and genetic algorithm." *Applied Microbiology and Biotechnology* **82**(2): 379-385.

Slimani L. (2009). "Contribution à l'application de l'optimisation par des méthodes métaheuristiques à l'écoulement de puissance optimal dans un environnement de l'électricité dérégulé." *Thèse de Doctorat*, Université de Batna. Batna, Algérie.

Stanbury P. F., Whitaker A. et Hall S. J. (1999). "Principles of fermentation technology." Butterworth-Heinemann, Oxford, UK.

Stone M. (1977). "An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion." *Journal of the Royal Statistical Society B* **39**(1): 44-47.

Talbi M. L. (2011). "Analyse et traitement du signal électrocardiographique (ECG)." *Thèse de Doctorat*, Université Mentouri de Constantine. Constantine, Algérie.

Tompos A., Margitfalvi J. L., Tfirst E. et Heberger K. (2007). "Predictive performance of "highly complex" artificial neural networks." *Applied Catalysis, A: General* **324**(0): 90-93.

Trelea I. C. (2003). "The particle swarm optimization algorithm: Convergence analysis and parameter selection." *Information Processing Letters* **85**(6): 317-325.

Trigueros D. E. G., Modenes A. N., Kroumov A. D. et Espinoza-Quinones F. R. (2010). "Modeling of biodegradation process of BTEX compounds: Kinetic parameters estimation by using particle swarm global optimizer." *Process Biochemistry* **45**(8): 1355-1361.

Tzanakou E. M. (2000). "Supervised and unsupervised pattern recognition: Feature extraction and computational intelligence." Taylor & Francis Group, London, UK.

Urbani D., Roussel-Ragot P., Personnaz L. et Dreyfus G. (1994). "The selection of neural models of nonlinear dynamical systems by statistical tests." *Neural Networks for Signal Processing [1994] IV. Proceedings of the 1994 IEEE Workshop*. 6-8 septembre 1994.

Wang C. R., Yuan H. J., Huang Z. Q., Zhang J. W. et Sun C. J. (2005). "A modified particle swarm optimization algorithm and its application in optimal power flow problem." *International Conference on Machine Learning and Cybernetics*, Guangzhou, China. 18-21 août 2005.

Xiong Z. Q., Tu X. R. et Tu G. Q. (2008). "Optimization of medium composition for actinomycin X2 production by *Streptomyces spp* JAU4234 using response surface methodology." *Journal of Industrial Microbiology & Biotechnology* **35**(7): 729-734.

Ye K., Jin S. et Shimizu K. (1998). "On the development of an intelligent control system for recombinant cell culture." *International Journal of Intelligent Systems* **13**(6): 539-560.

Yuan J. Q. et Vanrolleghem P. A. (1999). "Rolling learning-prediction of product formation in bioprocesses." *Journal of Biotechnology* **69**(1): 47-62.

Zafar M., Kumar S., Kumar S. et Dhiman A. K. (2012a). "Artificial intelligence based modeling and optimization of poly(3-hydroxybutyrate-co-3-hydroxyvalerate) production process by using *Azohydromonas lata* MTCC 2311 from cane molasses supplemented with volatile fatty acids: A genetic algorithm paradigm." *Bioresource Technology* **104**(0): 631-641.

Zafar M., Kumar S., Kumar S. et Dhiman A. K. (2012b). "Optimization of polyhydroxybutyrate (PHB) production by *Azohydromonas lata* MTCC 2311 by using genetic algorithm based on artificial neural network and response surface methodology." *Biocatalysis and Agricultural Biotechnology* **1**(1): 70-79.

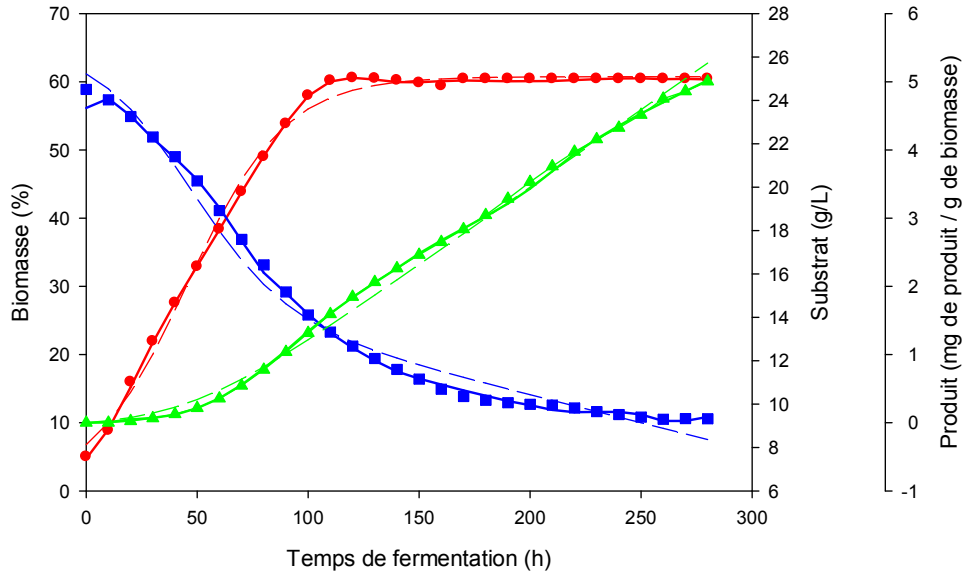
Zangirolami T. C., Johansen C. L., Nielsen J. et Jorgensen S. B. (1997). "Simulation of penicillin production in fed-batch cultivations using a morphologically structured model." *Biotechnology and Bioengineering* **56**(6): 593-604.

Zhang J. R., Zhang J., Lok T. M. et Lyu M. R. (2007). "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training." *Applied Mathematics and Computation* **185**(2): 1026-1037.

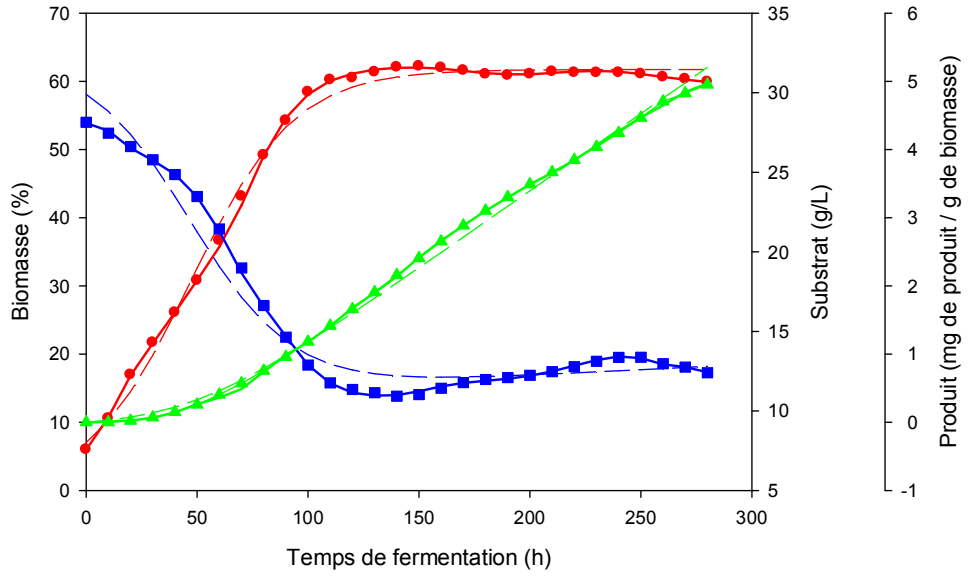
Zhang Y., Xu J., Yuan Z., Xu H. et Yu Q. (2010). "Artificial neural network-genetic algorithm based optimization for the immobilization of cellulase on the smart polymer Eudragit L-100." *Bioresource Technology* **101**(9): 3153-3158.

Zhu Y. H., Rajalahti T. et Linko S. (1996). "Application of neural networks to lysine production." *The Chemical Engineering Journal and the Biochemical Engineering Journal* **62**(3): 207-214.

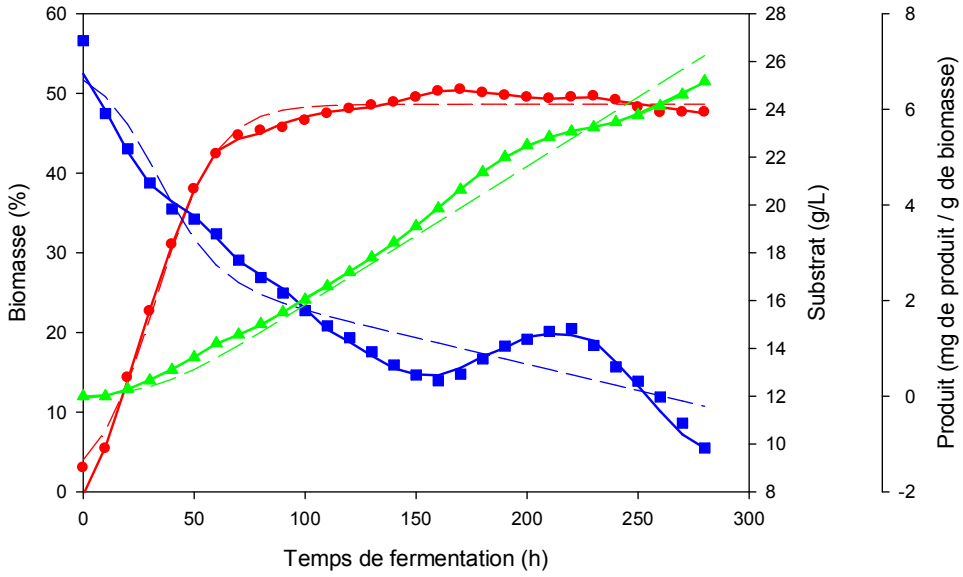
Annexe



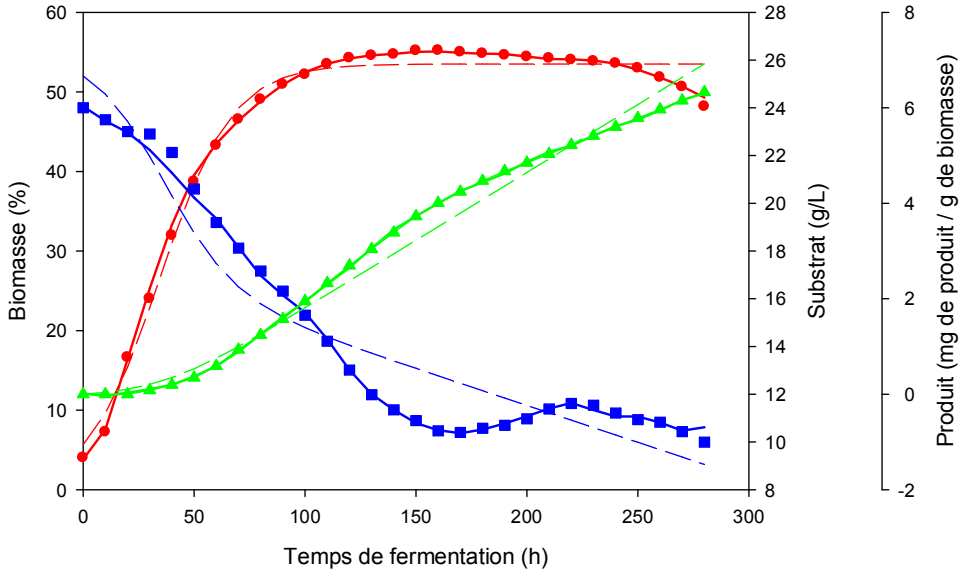
Lot N° 02



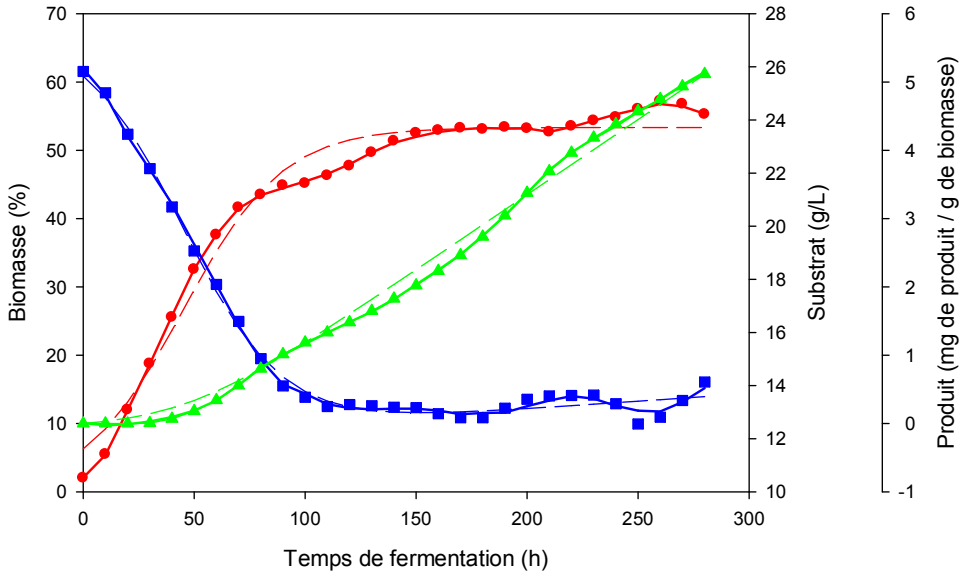
Lot N° 03



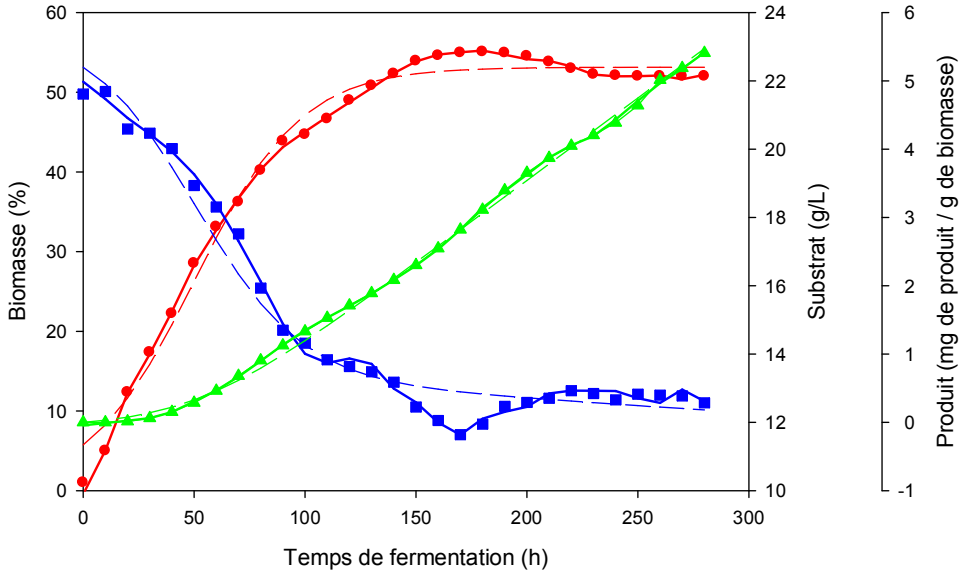
Lot N° 05



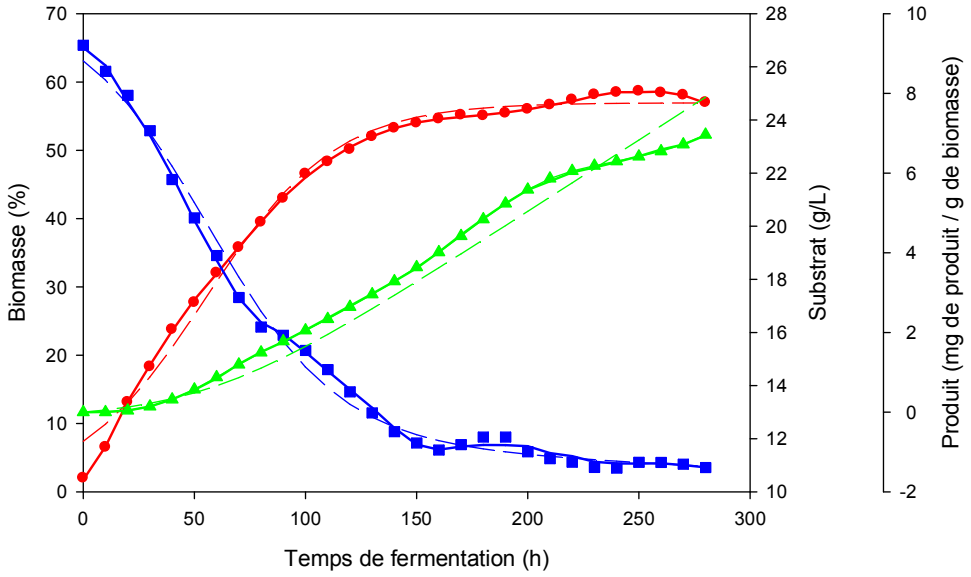
Lot N° 06



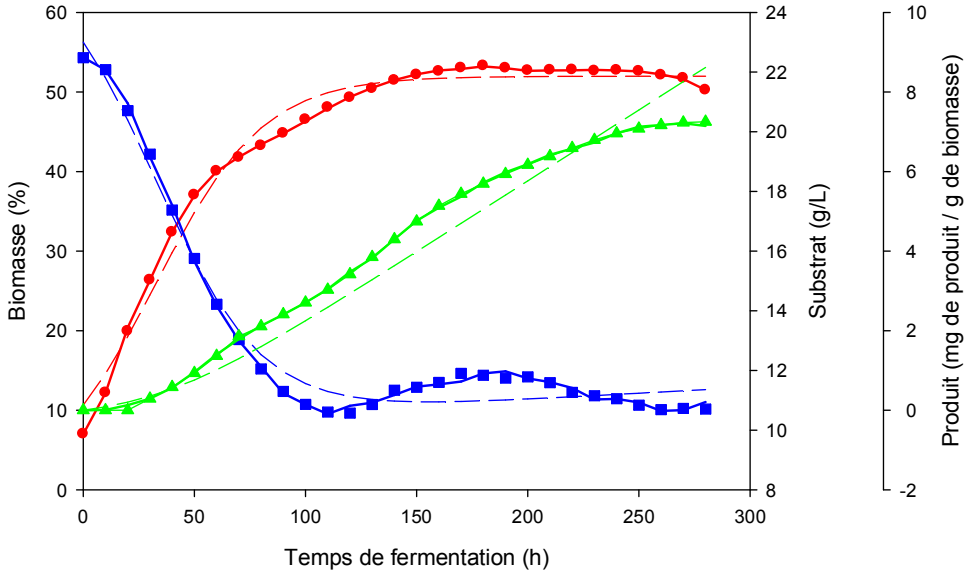
Lot N° 07



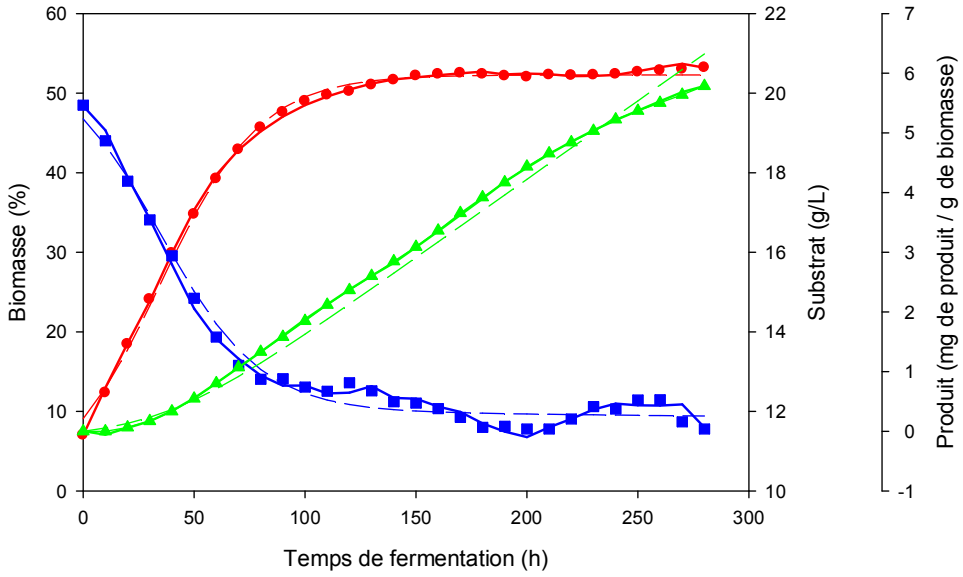
Lot N° 08



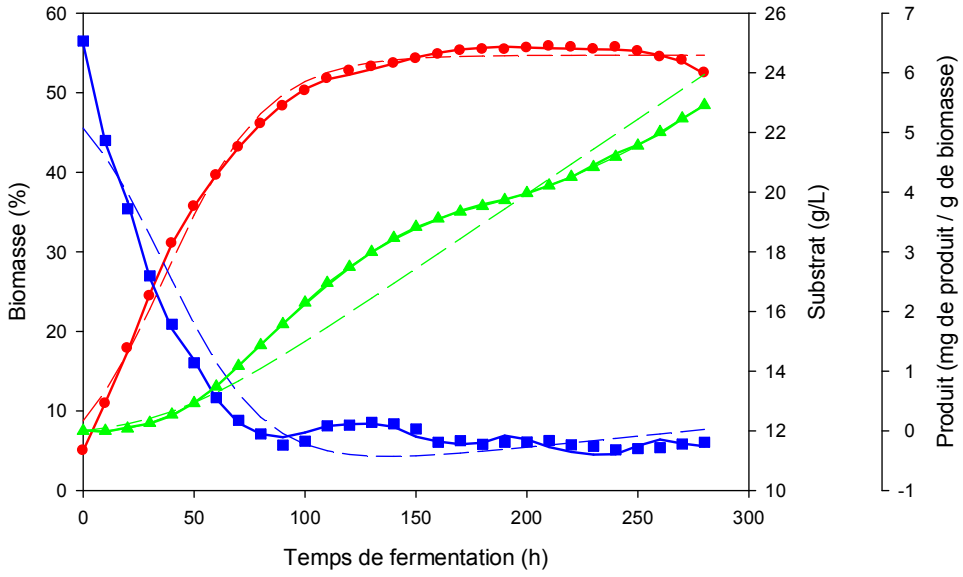
Lot N° 10



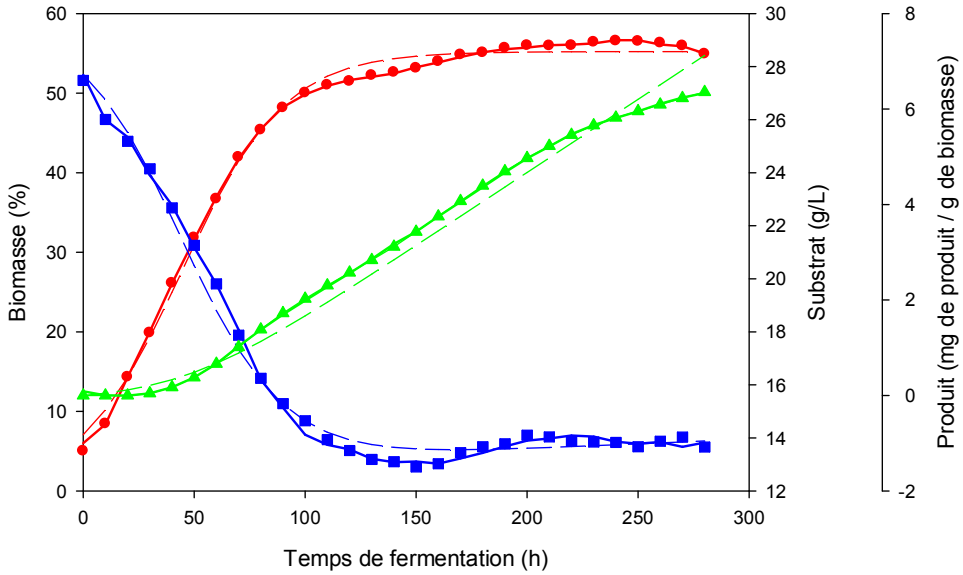
Lot N° 11



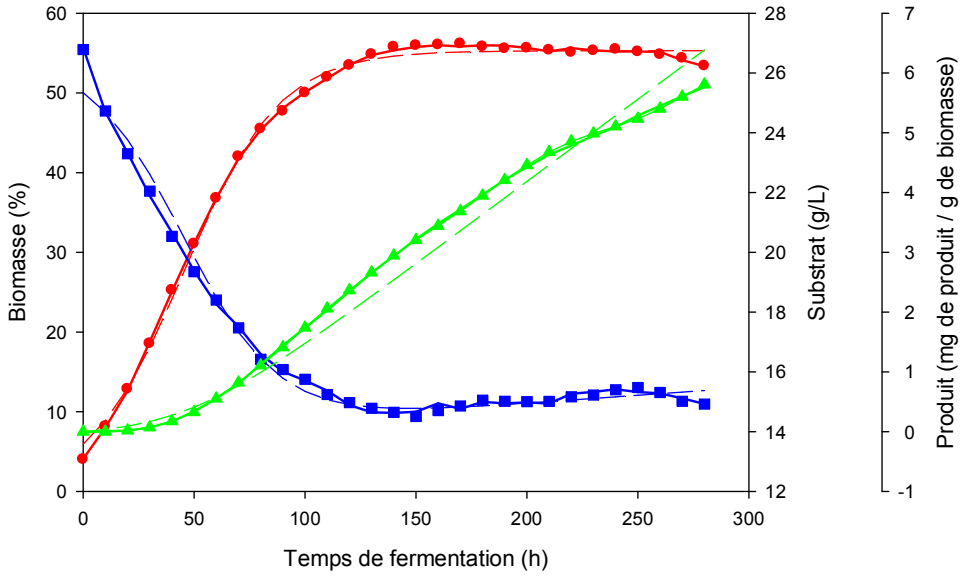
Lot N° 12



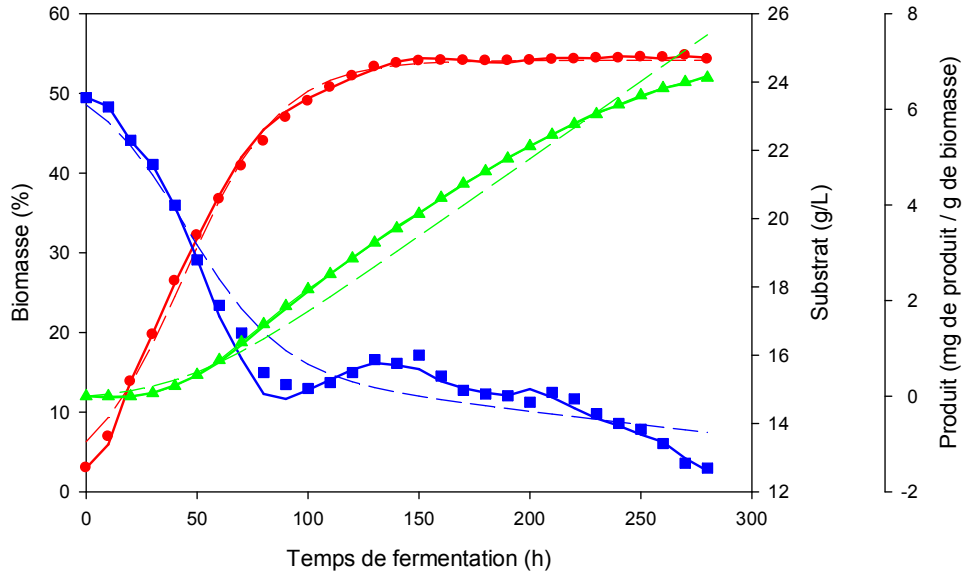
Lot N° 13



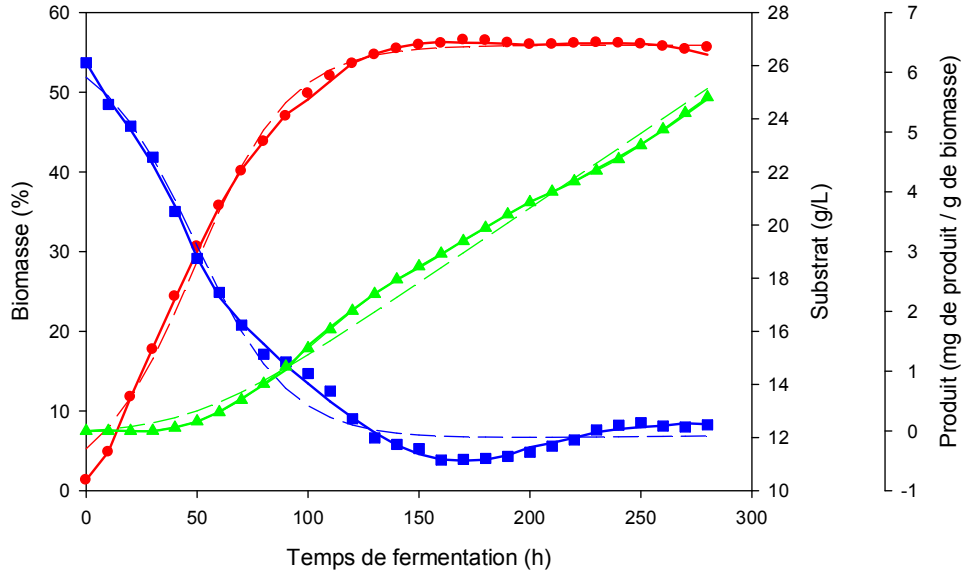
Lot N° 14



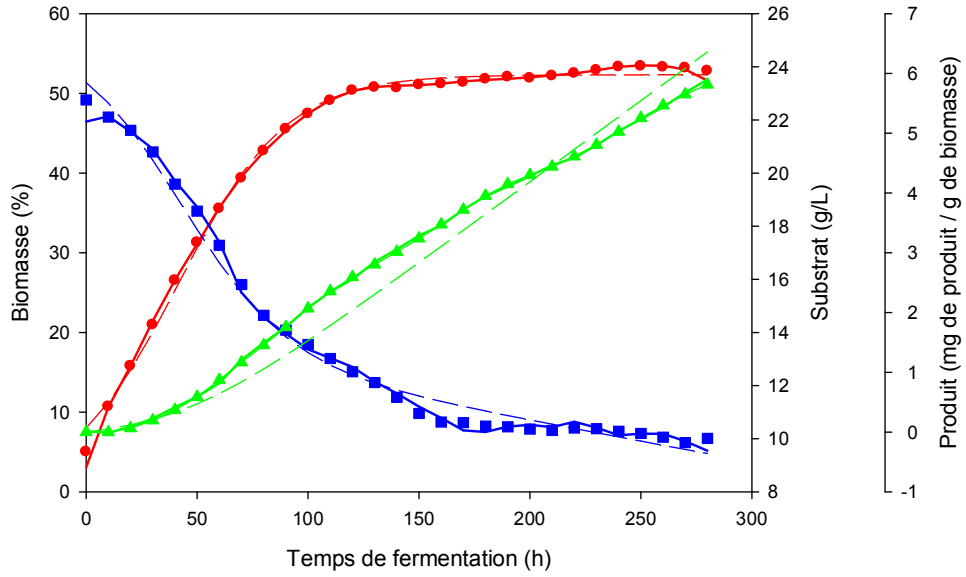
Lot N° 15



Lot N° 16



Lot N° 18



Lot N° 19

Figure 1. Résultats de modélisation.

(Les données expérimentales (symboles) pour la biomasse (●), glucose (■) et pleuromutiline (▲); Les résultats de la modélisation par RNFF sont représentés par des traits pleins et les résultats de la modélisation cinétique sont présentés par des lignes en pointillés).