



HAL
open science

Optimisation et apprentissage de modèles biologiques : application à l'irrigation [sic l'irrigation] de pomme de terre

Amaury Dubois

► **To cite this version:**

Amaury Dubois. Optimisation et apprentissage de modèles biologiques : application à l'irrigation [sic l'irrigation] de pomme de terre. Apprentissage [cs.LG]. Université du Littoral Côte d'Opale, 2020. Français. NNT : 2020DUNK0560 . tel-03144024

HAL Id: tel-03144024

<https://theses.hal.science/tel-03144024v1>

Submitted on 17 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ULCO

École doctorale ED Régionale SPI 72

Unité de recherche **Laboratoire d'Informatique Signal et Image de la Côte d'Opale (LISIC)**

Thèse présentée par **Amaury DUBOIS**

Soutenue le **15 décembre 2020**

En vue de l'obtention du grade de docteur de l'ULCO

Discipline **Sciences et Technologies de l'Information et de la Communication : Informatique**

Optimisation et apprentissage de modèles biologiques : application à l'irrigation de pomme de la terre

Thèse dirigée par Sébastien VEREL directeur
Fabien TEYTAUD co-encadrant

Composition du jury

<i>Rapporteurs</i>	Pierre PARREND Frédéric SAUBION	ICUBE, ECAM Strasbourg LERIA, Univ. Angers
<i>Examineurs</i>	Bilel DERBEL Évelyne LUTTON Jean-Noël AUBERTOT Virginie MARION-POTY	CRISAL, Univ. Lille/inria MIA, INRAe-AgroParisTech AGIR, INRAe-Auzeville LISIC, Univ. du Littoral Côte d'Opale
<i>Invités</i>	Éric RAMAT Laurent LELEU	LISIC, Univ. du Littoral Côte d'Opale WEENAT
<i>Directeurs de thèse</i>	Sébastien VEREL Fabien TEYTAUD	LISIC, Univ. du Littoral Côte d'Opale LISIC, Univ. du Littoral Côte d'Opale

COLOPHON

Mémoire de thèse intitulé « Optimisation et apprentissage de modèles biologiques : application à l'irrigation de pomme de la terre », écrit par Amaury DUBOIS, achevé le 4 janvier 2021, composé au moyen du système de préparation de document L^AT_EX et de la classe yathesis dédiée aux thèses préparées en France.

ULCO

École doctorale ED Régionale SPI 72

Unité de recherche LISIC

Thèse présentée par **Amaury DUBOIS**

Soutenue le **15 décembre 2020**

En vue de l'obtention du grade de docteur de l'ULCO

Discipline **Sciences et Technologies de l'Information et de la
Communication : Informatique**

Optimisation et apprentissage de modèles biologiques : application à l'irrigation de pomme de la terre

Thèse dirigée par Sébastien VEREL directeur
Fabien TEYTAUD co-encadrant

Composition du jury

<i>Rapporteurs</i>	Pierre PARREND Frédéric SAUBION	ICUBE, ECAM Strasbourg LERIA, Univ. Angers
<i>Examineurs</i>	Bilel DERBEL Évelyne LUTTON Jean-Noël AUBERTOT Virginie MARION-POTY	CRISAL, Univ. Lille/inria MIA, INRAe-AgroParisTech AGIR, INRAe-Auzeville LISIC, Univ. du Littoral Côte d'Opale
<i>Invités</i>	Éric RAMAT Laurent LELEU	LISIC, Univ. du Littoral Côte d'Opale WEENAT
<i>Directeurs de thèse</i>	Sébastien VEREL Fabien TEYTAUD	LISIC, Univ. du Littoral Côte d'Opale LISIC, Univ. du Littoral Côte d'Opale

ULCO

Doctoral School ED Régionale SPI 72

University Department LISIC

Thesis defended by **Amaury DUBOIS**

Defended on **15th December, 2020**

In order to become Doctor from ULCO

Academic Field IT

Optimization and learning of organic models: application to potato irrigation

Thesis supervised by Sébastien VEREL Supervisor
Fabien TEYTAUD Co-Monitor

Committee members

<i>Referees</i>	Pierre PARREND	ICUBE, ECAM Strasbourg
	Frédéric SAUBION	LERIA, Univ. Angers
<i>Examiners</i>	Bilel DERBEL	CRISTAL, Univ. Lille/inria
	Évelyne LUTTON	MIA, INRAe-AgroParisTech
	Jean-Noël AUBERTOT	AGIR, INRAe-Auzeville
	Virginie MARION-POTY	LISIC, Univ. du Littoral Côte d'Opale
<i>Guests</i>	Éric RAMAT	LISIC, Univ. du Littoral Côte d'Opale
	Laurent LELEU	WEENAT
<i>Supervisors</i>	Sébastien VEREL	LISIC, Univ. du Littoral Côte d'Opale
	Fabien TEYTAUD	LISIC, Univ. du Littoral Côte d'Opale

Mots clés : optimisation continue, optimisation boîte noire, optimisation multimodale, recherche arborescente, mcts, ucb, apprentissage par renforcement, apprentissage supervisé, sélection de caractéristiques

Keywords: continuous optimization , black box optimization, multimodal optimization, tree based search, mcts, ucb, reinforcement learning, supervised learning, feature selection

Cette thèse a été préparée au

LISIC

Maison de la Recherche Blaise Pascal
50, rue Ferdinand Buisson
CS 80699
62228 Calais Cedex
France

☎ (33)(0)3 21 46 36 53

✉ secretariat@lisic.univ-littoral.fr

Site <http://www-lisic.univ-littoral.fr/>



OPTIMISATION ET APPRENTISSAGE DE MODÈLES BIOLOGIQUES : APPLICATION À LIRRIGATION DE POMME DE LA TERRE**Résumé**

Le sujet de la thèse porte sur une des thématiques du LISIC : la modélisation et la simulation de systèmes complexes, ainsi que sur l'optimisation et l'apprentissage automatique pour l'agronomie. Les objectifs de la thèse sont de répondre aux questions de pilotage de l'irrigation de la culture de pomme de terre par le développement d'outils d'aide à la décision à destination des exploitants agricoles. Le choix de cette culture est motivé par sa part importante dans la région des Haut-de-France.

Le manuscrit s'articule en 3 parties. La première partie traite de l'optimisation continue multimodale dans un contexte boîte noire. Il en suit une présentation d'une méthodologie d'étalonnage automatique de paramètres de modèle biologique grâce à une reformulation en un problème d'optimisation continue mono-objectif multimodale de type boîte noire. La pertinence de l'utilisation de l'analyse inverse comme méthodologie de paramétrage automatique de modèles de grandes dimensions est ensuite démontrée. La deuxième partie présente 2 nouveaux algorithmes *UCB Random with Decreasing Step-size* et *UCT Random with Decreasing Step-size*. Ce sont des algorithmes d'optimisation continue multimodale boîte noire dont le choix de la position initiale des individus est assisté par un algorithme d'apprentissage par renforcement. Les résultats montrent que ces algorithmes possèdent de meilleures performances que les algorithmes état de l'art *Quasi Random with Decreasing Step-Size*. Enfin la dernière partie est focalisée sur les principes et méthodes d'apprentissage automatique (machine learning). Une reformulation du problème de la prédiction à une semaine de la teneur en eau dans le sol en un problème d'apprentissage supervisé a permis le développement d'un nouvel outil d'aide à la décision pour répondre à la problématique du pilotage des cultures.

Mots clés : optimisation continue, optimisation boîte noire, optimisation multimodale, recherche arborescente, mcts, ucb, apprentissage par renforcement, apprentissage supervisé, sélection de caractéristiques

LISIC

Maison de la Recherche Blaise Pascal – 50, rue Ferdinand Buisson – CS 80699
– 62228 Calais Cedex – France

OPTIMIZATION AND LEARNING OF ORGANIC MODELS: APPLICATION TO POTATO IRRIGATION**Abstract**

The subject of this PhD concerns one of the LISIC themes: modelling and simulation of complex systems, as well as optimisation and automatic learning for agronomy. The objectives of the thesis are to answer the questions of irrigation management of the potato crop and the development of decision support tools for farmers. The choice of this crop is motivated by its important share in the Haut-de-France region. The manuscript is divided into 3 parts. The first part deals with continuous multimodal optimisation in a black box context. This is followed by a presentation of a methodology for the automatic calibration of biological model parameters through reformulation into a black box multimodal optimisation problem. The relevance of the use of inverse analysis as a methodology for automatic parameterisation of large models is then demonstrated. The second part presents 2 new algorithms *UCB Random with Decreasing Step-size* and *UCT Random with Decreasing Step-size*. these algorithms are designed for continuous multimodal black-box optimization whose choice of the position of the initial local search is assisted by a reinforcement learning algorithms. The results show that these algorithms have better performance than *(Quasi) Random with Decreasing Step-Size* algorithms. Finally, the last part focuses on machine learning principles and methods. A reformulation of the problem of predicting soil water content at one-week intervals into a supervised learning problem has enabled the development of a new decision support tool to respond to the problem of crop management.

Keywords: continuous optimization , black box optimization, multimodal optimization, tree based search, mcts, ucb, reinforcement learning, supervised learning, feature selection

Remerciements

Je tiens à remercier la société weenat pour avoir accepté d'entreprendre cette collaboration ayant abouti à cette thèse.

Je tiens aussi à remercier aussi l'ANRT pour le financement de cette thèse.

Sommaire

Résumé	xi
Remerciements	xiii
Sommaire	xv
Liste des tableaux	xvii
Table des figures	xix
Introduction	1
I Problèmes d'optimisation et contributions dans la résolution de problème multimodal	35
1 Problèmes d'optimisation	37
2 Calibration de modèles paramétriques	59
II Problèmes d'optimisation assistée par renforcement et contributions dans la résolution de problèmes multimodaux	73
3 Problèmes d'apprentissage par renforcement	75
4 Problèmes d'optimisation assistée par renforcement	81
III Apprentissage supervisé de phénomènes biologiques et application à l'agriculture	107
5 Apprentissage automatique	109

6 Prédiction de phénomènes biologiques par apprentissage supervisé	125
Conclusion	157
Bibliographie	161
Table des matières	175

Liste des tableaux

1	Quantité d'eau nécessaire en litre pour produire 1 kilo de biens.	4
2	Table résumant les avantages et les inconvénients de l'utilisation de la sonde tensiométrique.	18
3	Table résumant les différents stades de développement d'un plant de tubercules pris en charge par le modèle PLANT.	25
2.1	Performances des jeux de paramètres pour la prédiction de l'évolution de la teneur en eau dans le sol en fonction de la méthode utilisée et de la parcelle de test. Les valeurs médianes en gras sont significativement meilleures selon le test Mann-Whitney à un niveau de confiance de 0,01. QRDS trouve toujours le meilleur jeu de paramètres, de plus la moyenne et la médiane des fitness sont meilleures (plus petite).	66
4.1	Nombre moyen d'évaluations nécessaires pour chaque algorithme pour découvrir tous les optima. Les valeurs en gras indiquent pour chaque expérience la meilleure performance. Un faible nombre d'évaluations indique un meilleur algorithme (avec l'intervalle de confiance à 95%).	89
4.2	Proportion d'optima découverts après 10^6 évaluations (avec l'intervalle de confiance à 95%).	90
4.3	Nombre moyen d'optima découverts par chaque algorithme après 10^6 évaluations sur 100 exécutions (un chiffre élevé indique une meilleure performance). Différents paramètres ont été testés, en particulier pour f_{Icop} , afin d'évaluer la robustesse de UCT-RDS.	102
6.1	Les 5 variables utilisées dans cette contribution.	132
6.2	Détails des performances à l'issue d'une leave-two-out cross validation avec un réseau de neurones selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.	143

6.3	Détails des performances à l'issue d'une leave-two-out cross validation avec une forêt d'arbres décisionnels selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.	144
6.4	Détails des performances à l'issue d'une leave-two-out cross validation avec un séparateur à vastes marges, selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.	144

Table des figures

1	Représentation des prélèvements en eau (en bleu pour les eaux de surface et en orange les eaux souterraines) uniquement pour les besoins de l'agriculture en 2013. (source : Banque Nationale des prélèvements quantitatifs en eau)	3
2	Représentation des déficits de précipitations entre les mois de juin 2018 à juillet 2019 par rapport à la mesure de référence sur la période 1991 - 2010. Une couleur de nuance rouge indique un fort déficit par rapport à la mesure de référence. (source : météo France)	5
3	Carte répertoriant les arrêtés de limitation des usages de l'eau au 29 août 2019. 87 départements ont pris des mesures de restriction et 7 étaient en vigilance. (source : Ministère du Développement durable)	6
4	Visuel de l'application. Focus sur le suivi d'une parcelle. Le graphique de droite affiche l'historique des mesures depuis le début de la culture. Les courbes rouges représentent les valeurs des tensiomètres exprimées en kPa. Une valeur faible de tensiomètre indique une forte quantité d'eau facilement utilisable pour la plante. Les différentes couleurs de fond sont explicitées sur la fig 5.	13
5	Visuel de l'application. Explications des couleurs de fond du suivi de culture.	14
6	Visuel de l'application. Page d'accueil d'un utilisateur de l'application. Toutes les parcelles suivies sont répertoriées sur la partie droite ainsi que les dernières valeurs des capteurs de cette parcelle. À gauche est indiqué le nom de l'exploitation suivie puis en dessous de gauche à droite : le nombre d'utilisateurs partageant ce compte, le nombre de parcelles et le nombre de capteurs implantés émettant leurs mesures.	14

7	Visuel de l'application. Prévisions de 2 OAD sur une parcelle. Sur la moitié haute sont présentées les météo expertes émettant un bulletin prévisionnel jusqu'à 10 jours. La partie basse est un outil de conseil sur la pulvérisation de produits phytosanitaires. En fonction du type d'intrant, un avis est émis toutes les heures sur les prochaines 24 h.	15
8	Ensemble des capteurs actuellement en service chez WEENAT.	15
9	Sonde tensiométrique. À noter la capsule de gypse englobant les électrodes (non visible)	16
10	cuillère à bascule servant à la mesure des précipitations (source Argalis)	17
11	Schéma du modèle WEEDRIQ avec ses différents sous-modèles. Les entrées et les interactions entre les modèles sont représentées par les flèches tandis que les éléments surlignés en vert indiquent les localisations dans les modèles où peuvent être renseignées les mesures capteurs.	19
12	Schéma du triangle de texture.	26
1.1	Fonctions convexes lisse (à gauche) et bruitée (à droite).	44
1.2	Schéma illustrant une fonction avec des optima locaux et un optimum global en minimisation.	45
1.3	Schéma illustrant la convergence de CMA-ES. La couleur blanche indique la zone optimale où converger, les cercles représentent les différentes lignes d'iso fitness. (source Wikipédia).	51
2.1	Schéma du principe de l'analyse inverse.	62
2.2	Simulation de la teneur en eau sur une campagne de culture. Sont représentées en noir les valeurs étalons issues des capteurs, en vert les valeurs du simulateur avec le paramétrage par défaut et en rouge celles avec le paramétrage optimisé.	65
2.3	Distribution des meilleurs optima trouvés à chaque optimisation pour chaque champ en fonction de l'algorithme d'optimisation utilisé. Une valeur faible indique une meilleure performance.	67
2.4	Distribution des meilleurs optima trouvés après 100 optimisations pour chaque champ en fonction de l'algorithme utilisé. Les points rouges représentent une valeur en dehors de l'intervalle de confiance. Une valeur faible indique une meilleure performance.	68
4.1	Fonctions multimodales utilisées pour les expériences (représentation dans un espace 2D).	87

4.2	Impact des paramètres de l'algorithme URDS avec $D = 2$. R est le paramètre régissant le compromis entre exploitation (faible valeur de R) et exploration (forte valeur de R). M est le nombre de subdivisions par dimension.	92
4.3	Illustration du partitionnement de UCT-RDS pour un espace en 2D et un partitionnement en $K = 3$ régions. L'algorithme découpe une région selon 1 dimension hiérarchiquement (a) et construit un arbre correspondant (b).	96
4.4	Fonctions multimodales utilisées dans les expériences (représentation en 2D).	99
4.5	Nombre d'optima découverts en fonction de la valeur de k_{UCT} (échelle en log), lorsque l'on partitionne une région en K sous-régions, sur f_{HumpSin} et en dimension 8.	103
5.1	Illustration d'une tâche de classification à gauche où l'on souhaite trouver une séparation entre les 2 formes et à droite d'une tâche de régression où l'on souhaite trouver une fonction qui explique au mieux tous les points.	112
5.2	Illustration d'un apprentissage sous optimal à gauche, équilibré et adapté au centre et d'un surapprentissage à droite.	112
5.3	Schéma simplifié d'un neurone, avec x_i les entrées, w_0 le biais, la fonction d'activation φ et une sortie y	114
5.4	Exemple d'un réseau de neurones "feed-forward fully connected" avec une couche cachée et un neurone dans la couche de sortie	115
6.1	Matrice de corrélation des 5 variables utilisées dans ce travail, sur la période de temps prise en considération (voir le texte).	134
6.2	Paramétrage des hyper paramètres des méthodes de sélection de variables avec l'utilisation de forêt d'arbres de décision comme algorithme d'apprentissage en fonction de plusieurs métriques. La valeur de k (un paramètre de la méthode RFE) correspond au nombre de variables conservées. Une valeur de seuil <i>Threshold</i> (paramètre de la méthode FM) élevée indique que seulement les paramètres avec un score de contribution élevé seront conservés. À l'exception de r^2 , une valeur faible indique une bonne performance.	136

6.3	Paramétrage des hyper paramètres des méthodes de sélection de variables avec l'utilisation de séparateurs à vastes marges comme algorithme d'apprentissage en fonction de plusieurs métriques. La valeur de k (un paramètre de RFE) correspond au nombre de variables conservées. Une valeur de seuil <i>Threshold</i> (paramètre de la méthode FM) élevée indique que seulement les paramètres avec un score de contribution élevé seront conservés. À l'exception de r^2 , une valeur faible indique une bonne performance.	137
6.4	Paramétrage des hyper paramètres des méthodes de sélection de variables avec l'utilisation d'un réseau de neurones comme algorithme d'apprentissage en fonction de plusieurs métriques. La valeur de k (un paramètre de RFE) correspond au nombre de variables conservées. Une valeur de seuil <i>Threshold</i> (paramètre de la méthode FM) élevée indique que seulement les paramètres avec un score de contribution élevé seront conservés. À l'exception de r^2 , une valeur faible indique une bonne performance.	138
6.5	Validation croisée de chaque méthode de sélection de variables. L'apprentissage est effectué en utilisant le meilleur paramétrage d'hyper paramètres pour chaque méthode. À l'exception de r^2 , une faible valeur indique une bonne performance.	145
6.6	Évolution des métriques en fonction du nombre de jours à prédire et des méthodes de sélection de variables (ou pas). À l'exception de r^2 , une faible valeur indique une bonne performance.	146
6.7	Évolution des métriques en fonction du nombre de jours à prédire et des méthodes de sélection de variables (ou pas). À l'exception de r^2 , une faible valeur indique une bonne performance.	147
6.8	Évolution des métriques en fonction du nombre de jours à prédire et des méthodes de sélection de variables (ou pas). À l'exception de r^2 , une faible valeur indique une bonne performance.	148
6.9	Simulation du 7e jour au cours d'une campagne de test. Cette campagne est caractérisée par de nombreux épisodes nuageux.	149
6.10	Simulation du 7e jour au cours d'une campagne de test. Cette campagne est caractérisée par de nombreux épisodes pluvieux.	149
6.11	Simulation du 7e jour au cours d'une campagne de test. Cette campagne est caractérisée par de nombreux épisodes de fortes chaleurs.	149

Introduction

Sommaire du présent chapitre

Contexte	2
Agriculture de précision	6
La maîtrise de l'irrigation	7
Solutions matérielles	8
Solutions logicielles	9
WEENAT	11
La société	11
Capteurs	13
WEEDRIQ	17
Architecture	18
Climat et rayonnement global	18
SWC	20
SOIL	21
WATER BALANCE	22
PLANT	23
Fonctionnement global	26
Objectifs	28

Contexte

Le rapport annuel du Conseil d'État de 2010 sur les données de consommation en eau en France [29] estime qu'en 2009, 33.4 milliards de m³ deau ont été prélevés en France métropolitaine. De cette quantité, 9% ont été utilisées pour les activités d'irrigation liées à l'agriculture. Cependant, cette part représente 48% de notre consommation, c'est-à-dire la quantité d'eau qui n'est pas directement restituée dans les réserves en eau après utilisation. Autrement dit, une part considérable de l'eau que nous gardons pendant une période étendue, et qui par conséquent n'est pas disponible pour notre environnement a pour origine nos activités d'irrigations.

En 2015; le rapport numéro 14061 [51] du ministère de l'Agriculture et de la forêt indique que les prélèvements en eau pour l'agriculture (toutes activités confondues, mais ne se limitant pas à l'irrigation) bien que modestes (environ 4 milliards de m³/an) par rapport à nos réserves annuelles globales (estimées à 160 milliards de m³/an) s'effectuent sur 3 mois coïncidant généralement avec létiage¹ des cours deau et rentrent alors en conflit avec les autres usages, parmi lesquels leau potable, définie comme un besoin prioritaire. Toujours d'après ce rapport les besoins vitaux en eau par jour et par personne sont estimés à 3 litres. De son côté, l'alimentation par l'agriculture en nécessite 1000 litres par jour et par personne nourrie. Fort heureusement, cette quantité est en grande partie fournie par les précipitations naturelles. Ainsi en France, l'irrigation soulève des controverses sociétales importantes surtout en période de sécheresse. Bien que la ressource en eau disponible en France soit abondante, sa répartition est inégale, et les prélèvements ne sont pas identiques d'une région à une autre (voir la carte 1).

Ce rapport souligne de plus l'augmentation des épisodes de sécheresses en conséquence du changement climatique et qui ont un impact direct sur les cultures pluviales, c'est-à-dire une agriculture dépendant entièrement des précipitations pour son approvisionnement en eau et devront elles aussi se tourner vers l'irrigation. En 2019; comme le montre la figure 2, les nappes phréatiques

1. Baisse périodique d'un cours d'eau vers son niveau le plus bas

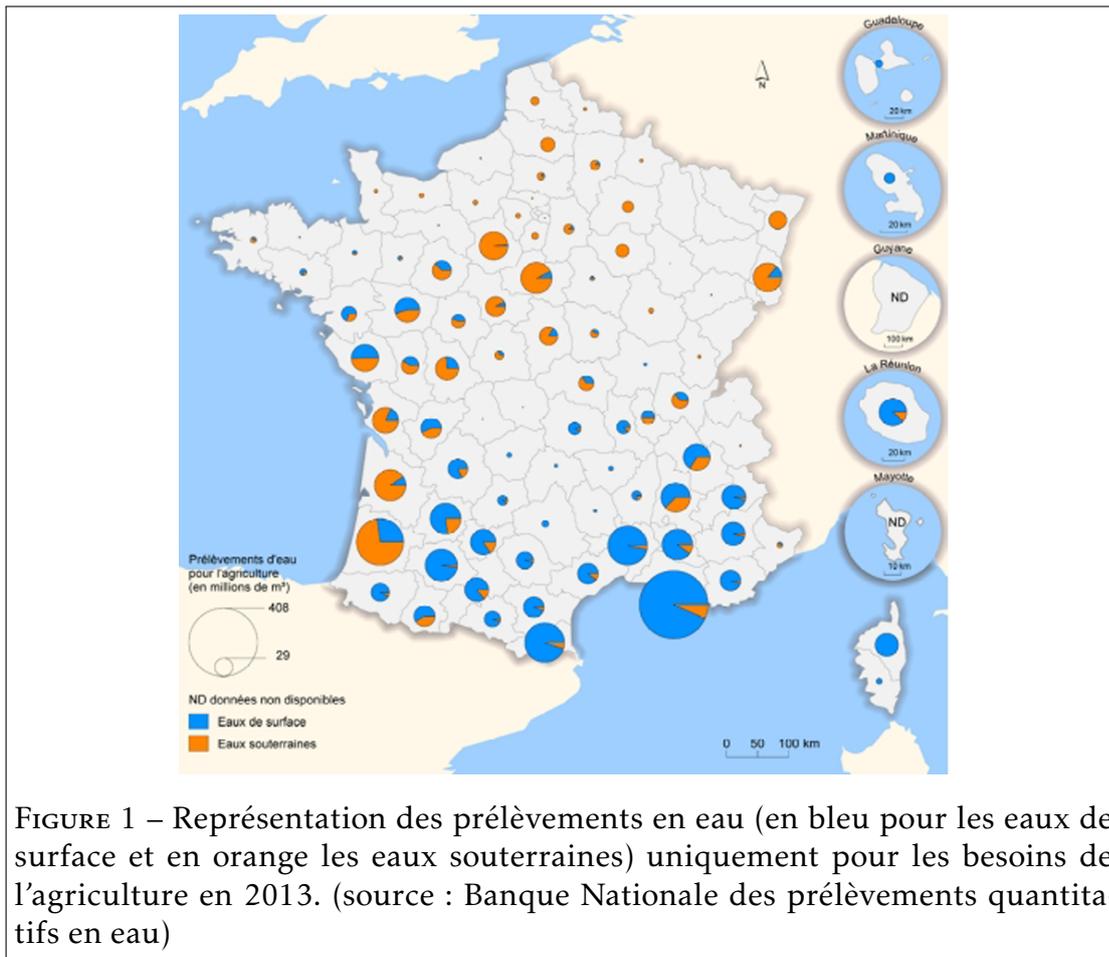


FIGURE 1 – Représentation des prélèvements en eau (en bleu pour les eaux de surface et en orange les eaux souterraines) uniquement pour les besoins de l'agriculture en 2013. (source : Banque Nationale des prélèvements quantitatifs en eau)

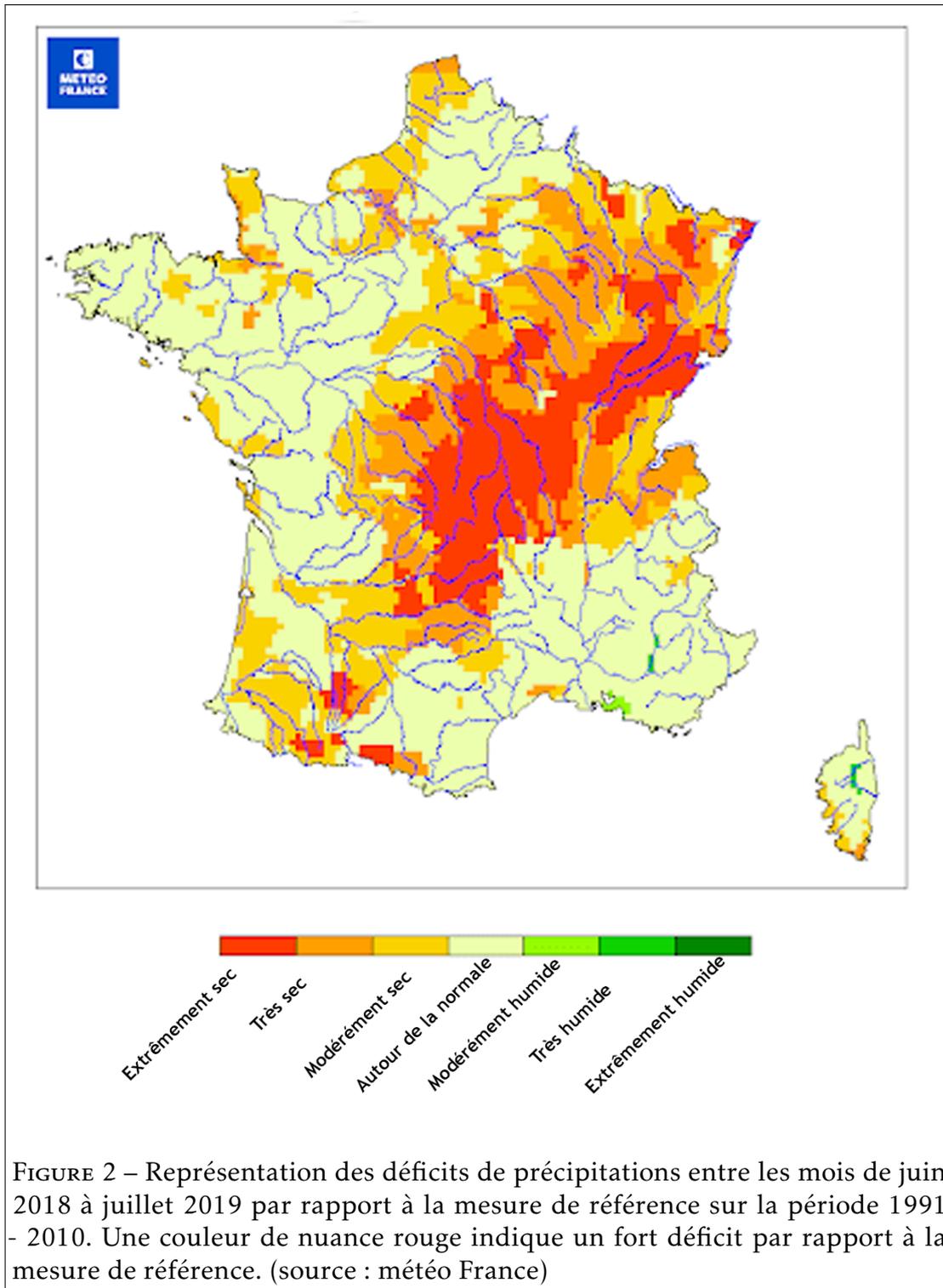
n'ont pas pu se recharger durant la période hivernale 2018-2019 [18], ce qui a conduit à des restrictions sévères quant à l'utilisation des ressources en eau (voir figure 3) en général et en particulier pour les exploitants agricoles[121].

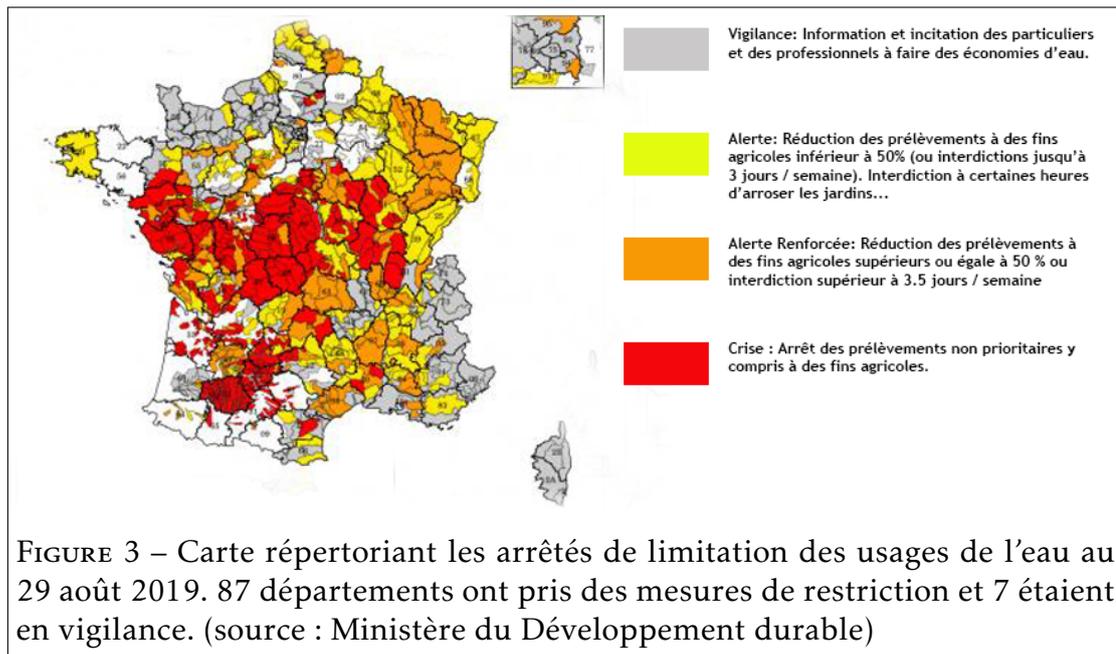
Si l'on prend comme exemple le Nord-Pas-de-Calais, en 2017 les prélèvements en eau représentaient 12 790 450 m³ [15] et la part prélevée dans les nappes phréatiques s'élevait à 84% [15]. Ces chiffres peuvent sembler importants pour les activités agricoles, cependant à titre de comparaison le tableau 1 reporte les quantités d'eau nécessaires pour produire un kilo de certains biens.

TABLEAU 1 – Quantité d'eau nécessaire en litre pour produire 1 kilo de biens.

Produit	Quantité d'eau $L.kg^{-1}$
Coton	5200
pomme de terre	590
blé	590
Maïs grain	454
Bière	25
salade	25

Dans la nouvelle région des Haut-de-France, 40 000 Hectares de terre sont consacrés à la culture de pomme de terre [1] faisant de cette région la première productrice de France avec presque 2/3 de la production nationale. La culture de pomme de terre est très sensible au stress hydrique[30] [89](c'est-à-dire un déficit d'eau par rapport aux besoins de la plante). Un stress hydrique important conduit à une forte dégradation des rendements de la plante. Ceci faisant écho à tout ce qui a été expliqué auparavant, la région Haut-de-France a donc un fort besoin de conseils pour les exploitants sur ce type culture. Dans ce contexte, les acteurs politiques, comme le ministère de l'Agriculture [97, 98] et les régions [87], promeuvent des techniques agricoles plus respectueuses de l'environnement et ayant un moindre impact sur les réserves en eau (techniques appelées "éco-responsable"). À cela on peut noter aussi que la Politique agricole commune (PAC) 2014-2020 [78], ainsi que ses prochaines versions, plaide pour une diminution et une meilleure gestion des ressources en eau. Dans ce but, une justification des pratiques et des apports en eau pourra être jugée nécessaire pour l'attribution des subventions de la PAC.





Agriculture de précision

Face aux enjeux actuels de productivité et de respect de l'environnement, une prise de conscience et une mutation vers des techniques agricoles eco-friendly" s'opèrent [99]. Ce qui se traduit concrètement par l'adoption depuis le début des années 2000 de *l'agriTech*² par les acteurs du monde agricole [66, 105]. Ce processus de mutation a vu l'émergence de *l'agriculture de précision* (partie intégrante de *l'agriTech*). Ce type d'agriculture nécessite l'emploi d'équipements de mesure afin de collecter des données sur les parcelles ou les cheptels avec pour objectifs de permettre aux exploitants agricoles de contrôler le déroulement des cultures, de comprendre les tendances globales et ainsi être conseillé sur la conduite à tenir sur leurs parcelles.

Jusqu'au début des années 2010, des équipements lourds embarqués (capteurs de rendement sur les moissonneuses-batteuses, robots de traite pour l'élevage, GPS, télédétection par satellite, ...) étaient principalement utilisés. Puis progressivement les capteurs autonomes, bien qu'ils ne soient pas récents [41],

2. Utilisation des technologies dans le but d'améliorer la qualité et les rendements des produits ou services issus de l'agriculture et de l'élevage

ont pris une place de plus en plus importante dans ce type d'agriculture. Ceci a été rendu possible par une baisse des coûts elle-même possible par le développement de l'offre dans ce secteur d'activités[28].

La maîtrise de l'irrigation

Comme expliqué précédemment, la maîtrise des ressources en eau potable liée à l'activité de l'irrigation est un enjeu majeur [50] puisqu'elle influe directement sur la qualité et les rendements des cultures [39]. Plusieurs techniques d'irrigation en plein champ existent (goutte à goutte, gravitaire, aspersion, ...), cependant l'irrigation par aspersion, plus communément appelée "cannons à eau", reste la méthode la plus utilisée [45]. Elle a pour avantage d'être une solution pérenne.³ et mobile dans le sens où elle peut être déplacée d'une parcelle à une autre en fonction des besoins. Malheureusement, ce moyen d'irrigation est chronophage et coûteux en terme financier s'il n'est pas géré de manière efficace. En effet, il monopolise du personnel ainsi que le temps d'installation sur une parcelle. De plus, l'irrigation en elle-même est soumise à une taxe sur le prélèvement d'eau. Cet apport sur la culture peut avoir un faible impact si la teneur en d'eau dans le sol était déjà suffisante ou si la quantité aspergée était trop faible. Ceci peut conduire à un potentiel gaspillage d'eau et d'argent.

Ainsi ces problématiques ont fait naître un besoin chez les exploitants agricoles souhaitant mieux comprendre la notion de bilan hydrique⁴ leur permettant d'être conseillés dans la prise de décision sur la gestion et l'ordonnement de l'irrigation des cultures. La donnée principale (voire critique) de cette prise de décision est la connaissance de la teneur en eau dans le sol [40]. Cette valeur peut être exprimée sous différentes formes en fonction des techniques de mesure (réserve utile, réserve facilement utilisable, potentiel hydrique, réserve potentielle en eau, ...). Toutes ces valeurs sont dynamiques dans le sens où une surveillance quotidienne et régulière est nécessaire pour analyser et appréhender les tendances globales. L'accumulation des données passées permet aux

3. L'ensemble du kit d'irrigation est conservé après la récolte

4. Évolution de la teneur en eau dans le sol

exploitants agricoles d'estimer grâce à la connaissance de leur métier les tendances futures de la teneur en eau. Mais être en mesure de proposer une solution pour estimer de manière fiable à partir des données passées les prochaines valeurs sur un horizon de l'ordre de la semaine serait un bénéfice majeur pour les fermiers puisqu'il leur serait alors plus aisé d'organiser de manière plus pertinente leurs plannings d'irrigation ("tour d'eau").

Solutions matérielles

Afin de répondre à ces besoins, des solutions "capteurs autonomes" ont été développées pour l'acquisition de la teneur en eau. Malheureusement ces capteurs nécessitent l'intervention de l'agriculteur ou d'un technicien pour récolter les données stockées dans la mémoire du capteur. Plus récemment, le développement de l'*IoT*⁵ et des moyens de télécommunication longue distance (Lora, SigFox, ...) ont permis l'apparition et la démocratisation de capteurs autonomes connectés capables de téléverser leurs données directement sur un serveur distant, capitalisant ainsi toutes les données des clients. À propos des capteurs eux-mêmes, 2 méthodes de mesure sont possibles :

- La méthode directe ou gravimétrique [41] consiste à prélever des échantillons régulièrement sur les parcelles suivies puis de mesurer précisément en laboratoire les valeurs souhaitées. Cette technique est à ce jour la seule technique ayant les résultats les plus fiables en ce qui concerne la collecte de données. Cependant, elle souffre de nombreuses limitations. La principale étant le nombre important de prélèvements destructifs à effectuer sur une parcelle pour le suivi ce qui implique un coût financier et humain important ainsi qu'un protocole strict à respecter pour éviter l'introduction d'erreurs de mesure. Cette méthode est à privilégier pour étalonner les instruments de mesure.
- La méthode indirecte (utilisation des capteurs). Cette approche est une bonne alternative à la méthode directe puisqu'elle est possible sans forcément avoir recours aux prélèvements destructifs [41]. De nombreux

5. Internet Of Thing

modèles de capteurs sont actuellement disponibles (tensiométrique, capacitif, résistif, hygrométrique, ...). Leurs méthodes de mesure diffèrent, mais le résultat permet d'obtenir une information sur la teneur en eau. La qualité de la donnée collectée est directement influencée par la qualité du capteur utilisé. En effet, ces derniers sont soumis aux conditions climatiques d'une culture en plein champ (pluie, température, vent, mauvaises manipulations de la part du personnel, ...) et ce pendant une période s'étalant sur plusieurs mois (3 à 4 mois pour la pomme de terre par exemple). L'exposition prolongée aux intempéries provoque une dérive des capteurs (plus ou moins importante en fonction de la qualité) et donc une dégradation de la précision et de la fiabilité de la mesure [141].

La mesure par tensiomètre est souvent préférée aux autres moyens de mesure du fait de son faible coût de construction, de sa simplicité d'utilisation, et aussi de la possibilité de mesure directe. De plus, ils sont moins sensibles à la température et au potentiel osmotique.⁶ du sol [99, 128]

Solutions logicielles

Conjointement avec le développement des solutions capteurs apparaît *l'agri-intelligence* (outils permettant d'agréger toutes les informations stratégiques utiles à l'exploitation pour leur donner, après interprétation, de la valeur ajoutée. C'est un secteur essentiellement immatériel, permettant d'utiliser les données précédemment collectées afin d'apporter une plus-value. Toujours pour répondre aux besoins liés à l'estimation de la teneur en eau, plusieurs approches co-existent :

- Les systèmes experts, datant des années 90 [42, 66], représentent la première exploration de la formalisation des règles de raisonnement basées sur des experts du domaine. Ces systèmes ont connu un succès grâce à l'automatisation de la conduite et des exécutions mécaniques simples par exemple pour le pilotage des serres, la traite des vaches, la pulvérisation d'intrants sur les parcelles via guidage satellite ... On peut citer comme

6. Composante du potentiel hydrique. Corresponds à une concentration en soluté dans le sol. L'eau se déplace d'une surface A vers B si le potentiel osmotique de B est plus faible que celui de A.

exemple logiciel (pas forcément pertinent pour la question de la teneur en eau), COMAX [83] pour la culture du coton, ALES [114] utilisant les recommandations de la FAO (1976), Water Balance [6] pour la conduite à tenir sur les apports en eau et estimation des bilans hydriques basés sur les recommandations de la FAO.

- Les modèles mécanistes ou biologiques. L'idée est de simuler les comportements et dynamiques de phénomènes. On peut citer SPUD-GROTHW [69] simulant le développement d'un plant de pommes de terre, STIC [19] pour le développement de culture, AQUACROP [110] pour simuler les rendements en fonction des apports en eau, WEEDRIQ [111] pour le bilan hydrique de pomme de terre, ... le fonctionnement de ces modèles repose sur des relations mathématiques définies par des designers humains en fonction de tests réalisés sur des campagnes de prélèvement et des conseils d'expert. Ceci leur permet de donner une bonne explication quant aux réponses simulées. Ils ont prouvé leur pertinence et leur robustesse et font de ces approches des valeurs sûres. Cependant, la plupart sont des modèles paramétriques, ce qui implique qu'un étalonnage des paramètres est nécessaire afin de pouvoir les utiliser de manière optimale [23, 124, 38].
- L'intelligence artificielle. Cette approche n'est pas récente comme en témoigne les travaux par exemple en 1996 de Pachepsky [104] ou en 1998 avec Schaap [118] et sont encore une solution pertinente de nos jours [2, 108] ou plus récemment les travaux de Yamacc sur l'estimation de l'évapotranspiration de la pomme de terre (2020) [138] pour résoudre ces problèmes. Généralement cette approche utilise le principe de l'apprentissage automatique afin de générer des modèles à partir des données collectées. Les résultats obtenus sont généralement satisfaisants, mais la nécessité d'avoir une quantité de données suffisamment grande et couvrant le maximum de possibilités en limite fortement son emploi à grande échelle.

WEENAT

Cette section présente la société afin de mieux comprendre sa vision, son niveau de maturité et donc d'expertise qu'elle possède. Une explication des outils (matériels et immatériels) actuellement utilisés permet d'avoir une meilleure compréhension générale des méthodes d'acquisition des données ainsi que des moyens mis en uvre durant le déroulement de cette thèse.

La société

Fondée en 2014 à Euratechnologie à Lille par Jérôme Leroy, cette start-up a été créée suite à un constat : la durée travail d'un agriculteur est estimée à 53 heures par semaine, contre 37 heures pour la moyenne française selon IIN-SEE [65]. La vision de WEENAT est alors claire : "aider les agriculteurs en leur donnant la possibilité d'accéder, directement depuis leur téléphone, à des informations fiables et précises sur le sol, les plantes, la météo, et ainsi faciliter la prise de décision." ⁷ Suite à quoi sa première station météo agricole 100% *made in France* a vu le jour. La volonté affichée de proposer une démocratisation des solutions capteurs a permis à WEENAT de rapidement se développer à travers la France. En 2018 le siège social a été transféré à Nantes, année aussi de l'obtention des financements de la première thèse Cifre soit 4 ans seulement après sa création. En 2020, les chiffres clés de WEENAT peuvent être résumés ainsi [86] :

- Un réseau de 4000 utilisateurs
- Une présence en France et dans 6 pays européens
- 60 Distributeurs et partenaires : coopératives, négoce, instituts et agro-industriels
- Un taux de réabonnement de 95%
- 22 collaborateurs

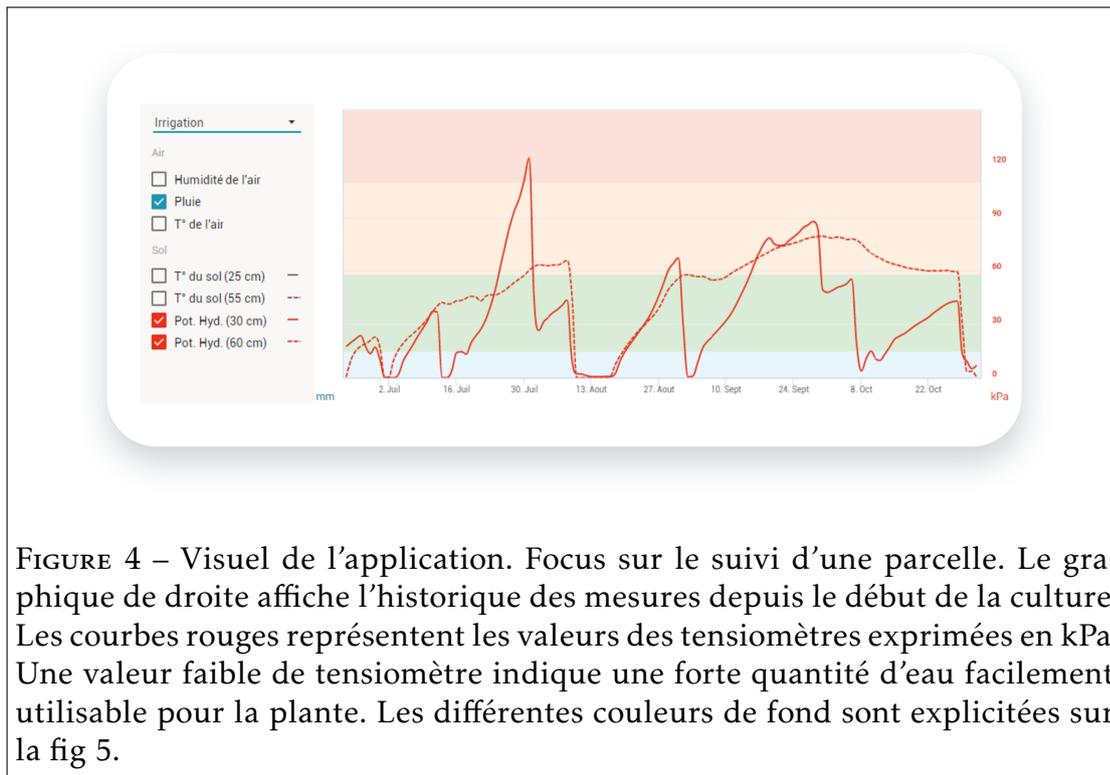
Ces chiffres démontrent la qualité et l'expertise que WEENAT a acquises aussi bien dans la conception de capteurs que dans la proposition de services. Au niveau opérationnel, la société se divise selon 3 couches :

7. Déclaration de Jérôme Leroy, fondateur de WEENAT

- **Acquisition** : cette couche englobe toute la partie capteur. Comme précisé avant, WEENAT fabrique ses propres capteurs autonomes et connectés, ce qui permet à ses ingénieurs de pouvoir modifier et améliorer les capteurs existants en fonction des retours d'expérience. Par exemple, au moment de la rédaction de cette thèse, le tensiomètre en vigueur en est à sa 4e version depuis 2014. D'autre part, de nouveaux besoins apparaissent. Ainsi la société est en mesure d'innover en proposant de nouveaux capteurs.
- **Mobilité** : derrière ce terme se cache toute l'infrastructure sur laquelle reposent les services de la société. Tous les capteurs de la couche précédente sont autonomes⁸ et connectés⁹. De ce fait les données sont téléversées depuis les parcelles vers l'application. Ce qui permet aux utilisateurs de l'application d'avoir un accès en temps réel aux informations de leurs parcelles (figure 4) à partir de n'importe quel navigateur internet (mobile ou PC). L'ambition de WEENAT est aussi de proposer un nouveau modèle collaboratif, inclusif et ouvert de la pratique agricole. La collaboration avec les négoce, coopératives agricoles et groupements de producteurs combinée avec le maillage des capteurs sur le territoire (par exemple, dans le Pas-de-Calais, la granularité du maillage est de 5 km entre 2 stations WEENAT) permet le développement de communautés dagriculteurs au niveau régional pouvant accéder aux données sans avoir nécessairement besoin dun capteur. Les figures 4,5,6,7 présentent des visuels de l'application dans sa version 2.0.
- **Outil d'aide à la décision (OAD)** : La dernière couche consacrée exclusivement aux services n'est possible que par la maîtrise des autres couches. Les données acquises servent à renseigner des OAD ou solutions logicielles 1. Actuellement les OAD disponibles sont :
 - Météo experte, basée sur un regroupement de plusieurs bulletins météo, permet une prévision à 10 jours actualisée 6 fois par jour.
 - Avizioó, Decitraitó, Mileosó, Movidaó, RIMpro, VitiMeteo, Xarvioó,

8. Ne nécessite pas l'intervention d'un opérateur humain tout au long d'une campagne de culture

9. Par l'intermédiaire d'un médium de télécommunication très longue distance comme Sig-Fox



sont des outils d’aide à la protection contre les maladies des cultures.

— IRRInov, Vintel, Irré-LIS, pour le pilotage de l’irrigation et des apports des cultures.

La figure 7 illustre certaines prévisions des OAD sur l’application.

Capteurs

Fort de son expérience(voir section 1), WEENAT propose aujourd’hui 6 types de capteurs (présentés sur la figure 8). Dans le cadre de cette thèse, les données ont été récoltées à l’aide de ces capteurs et en particulier grâce aux tensiomètres et aux pluviomètres. C’est sur ces 2 capteurs qu’une présentation du principe de fonctionnement va suivre.

Tensiomètre

Utilisée depuis 1978 [26], la sonde tensiométrique visible sur la figure 9 est un capteur de résistance électrique à l’état solide utilisée pour mesurer la ten-

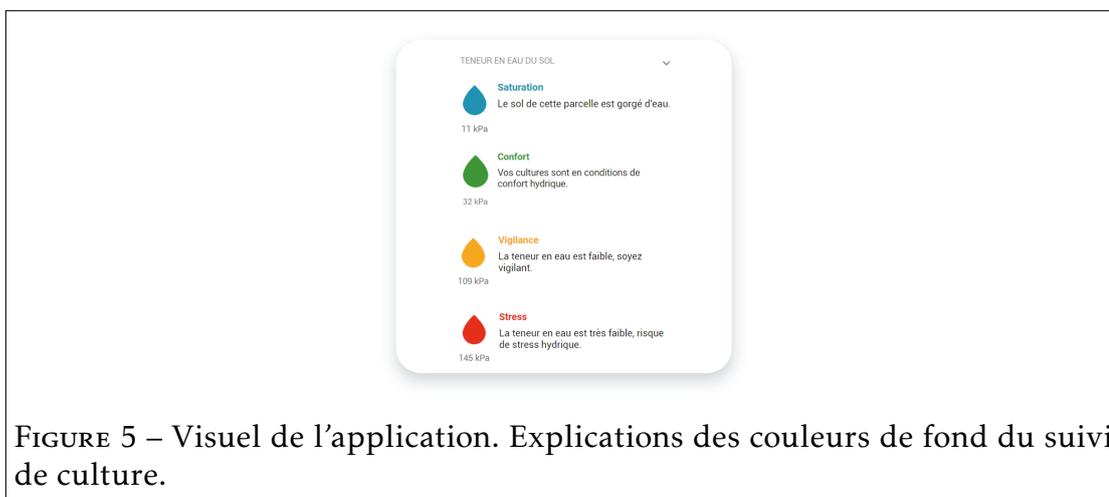


FIGURE 5 – Visuel de l'application. Explications des couleurs de fond du suivi de culture.



FIGURE 6 – Visuel de l'application. Page d'accueil d'un utilisateur de l'application. Toutes les parcelles suivies sont répertoriées sur la partie droite ainsi que les dernières valeurs des capteurs de cette parcelle. À gauche est indiqué le nom de l'exploitation suivie puis en dessous de gauche à droite : le nombre d'utilisateurs partageant ce compte, le nombre de parcelles et le nombre de capteurs implantés émettant leurs mesures.

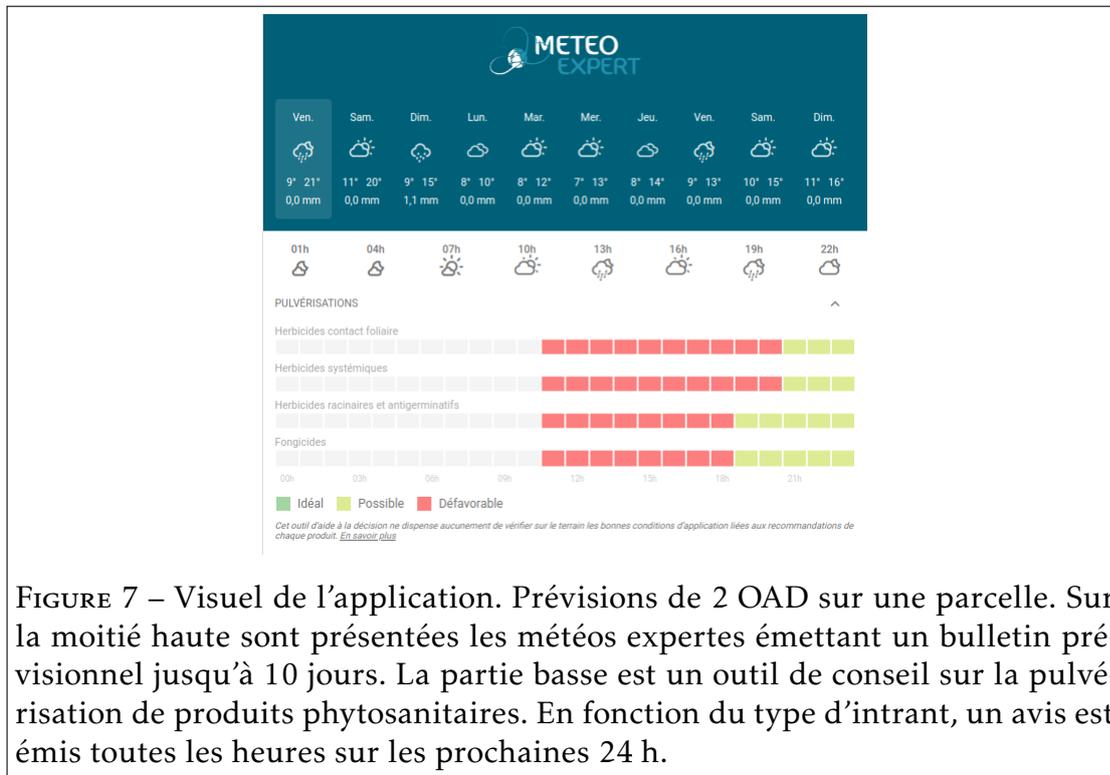


FIGURE 7 – Visuel de l’application. Prévisions de 2 OAD sur une parcelle. Sur la moitié haute sont présentées les météos expertes émettant un bulletin prévisionnel jusqu’à 10 jours. La partie basse est un outil de conseil sur la pulvérisation de produits phytosanitaires. En fonction du type d’intrant, un avis est émis toutes les heures sur les prochaines 24 h.

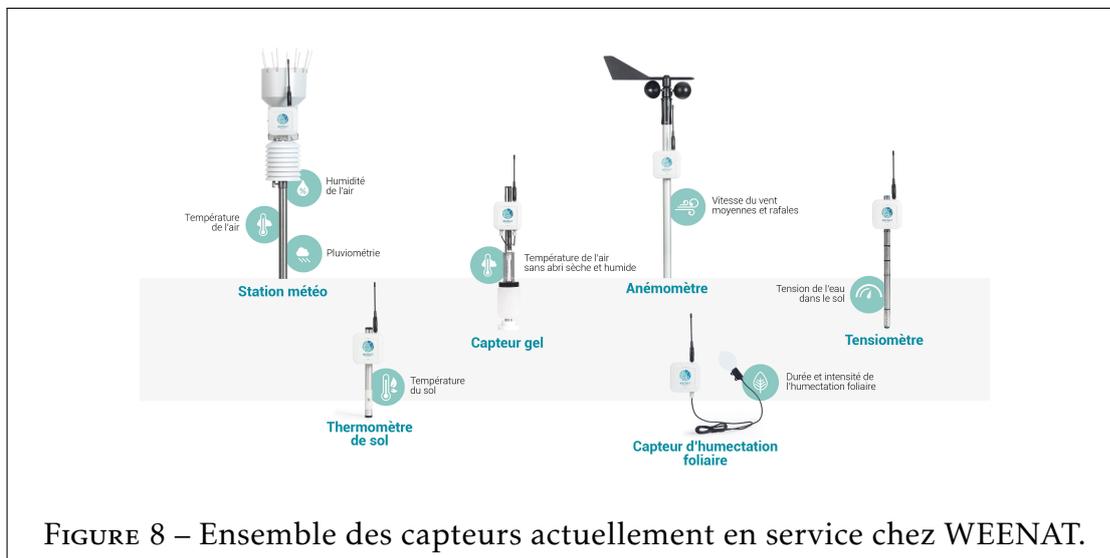


FIGURE 8 – Ensemble des capteurs actuellement en service chez WEENAT.



FIGURE 9 – Sonde tensiométrique. À noter la capsule de gypse englobant les électrodes (non visible)

sion d'eau du sol. Elle ne mesure pas directement la quantité d'eau présente dans le sol à l'instar des sondes capacitives, mais plutôt sa disponibilité pour la plante. Les sondes tensiométriques mesurent donc la force que la racine doit déployer pour extraire leau du sol. Cette force est exprimée en Centibars (CBARS) ou en kilopascal (kPa), l'unité utilisée chez WEENAT étant le kPa. Lorsque la teneur en eau change la résistance électrique change également et donc sa tension. C'est cette résistance que l'on mesure. Le capteur se compose d'une paire d'électrodes qui sont intégrées dans une matrice granulaire. Ces électrodes doivent être résistantes à la corrosion pour éviter les erreurs de mesure puisqu'elles vont être enterrées durant toute la période de mesure. Pour effectuer une mesure, un courant est appliqué entre ces électrodes afin d'obtenir une valeur de résistance. Le gypse installé à l'intérieur sert de tampon pour l'effet des niveaux de salinité que l'on trouve normalement dans les cultures et les paysages agricoles irrigués.

Les principaux avantages et inconvénients sont listés dans le tableau 2.

Pluviomètre

Afin de mesurer les précipitations en mm, les pluviomètres utilisés par WEENAT (inclus dans la station météo) utilisent le système de basculement à cuillère. Le principe de fonctionnement est simple, mais efficace :



FIGURE 10 – cuillère à bascule servant à la mesure des précipitations (source Argalis)

- Une cuillère tarée¹⁰ est retenue par un aimant en ferrite dure (fig 10)
- Lorsqu’il pleut, la cuillère se remplit
- Quand la cuillère est pleine, son poids est supérieur à la force d’attraction de l’aimant et la cuillère bascule en un bref moment (environ 300 mS).
- La cuillère se vide puis la force d’attraction de l’aimant remet en position initiale la cuillère.

La cuillère étant tarée, la quantité d’eau est alors facile à mesurer comme étant la somme des basculements de la cuillère. Ce système présente des avantages par rapport à d’autres systèmes comme le double auget. Le principal est la précision puisqu’il n’y aucune possibilité pour que de l’eau reste dans la cuillère lors d’un basculement. Dans le but de limiter les erreurs de mesure, l’ensemble du capteur est enfermé dans une enceinte.

WEEDRIQ

Au début de la collaboration entre le LISIC et WEENAT, un modèle agronomique de bilan hydrique à destination de la culture de pomme de terre (WEEDRIQ) a été développé par Ramat [111]. Ce modèle est capable de simuler et d’estimer la teneur en eau présente dans le sol ou *SWC* au cours du temps. Cette section présente ce modèle afin d’appréhender son architecture ainsi que les

10. 0.2 mm d’erreur pour les capteurs de WEENAT

TABLEAU 2 – Table résumant les avantages et les inconvénients de l'utilisation de la sonde tensiométrique.

Avantages	Inconvénients
- Peu coûteux	- Volume de mesure faible (quelques cm autour de la sonde)
- Seuils référencés	- Plusieurs tensiomètres nécessaires à différentes profondeurs
- Mesure en télétransmission possible	- Plage de valeurs limitée (0 à 200 kPa)
- Peu d'entretien	- Interprétation demandant de la technicité
	- Installation fastidieuse
	- Risque d'imprécisions

problématiques qui seront traitées par la suite.

Architecture

Le modèle global (dont une représentation schématisée est présentée sur la figure 11) est un automate temporel écrit dans un formalisme DEVS. Il est composé de plusieurs sous-modèles représentant des fonctions biologiques ainsi qu'une simulation du sol. Les modèles constituant WEEDRIQ sont mis en relation par l'intermédiaire de leurs entrées/sorties.

Climat et rayonnement global

Le point d'entrée du modèle global se situe au niveau du modèle simulant le climat ainsi que le rayonnement global (noté CLIMATE + RG). À partir des données météo provenant des capteurs, l'automate transforme ces données afin qu'elles puissent être utilisées par les autres modèles.

À noter que le système temporel utilisé pour la mise en forme est le système de date julienne afin de s'affranchir des problèmes inhérents au calendrier grégorien (jours différents dans les mois, années bissextiles, réformes calendaires,

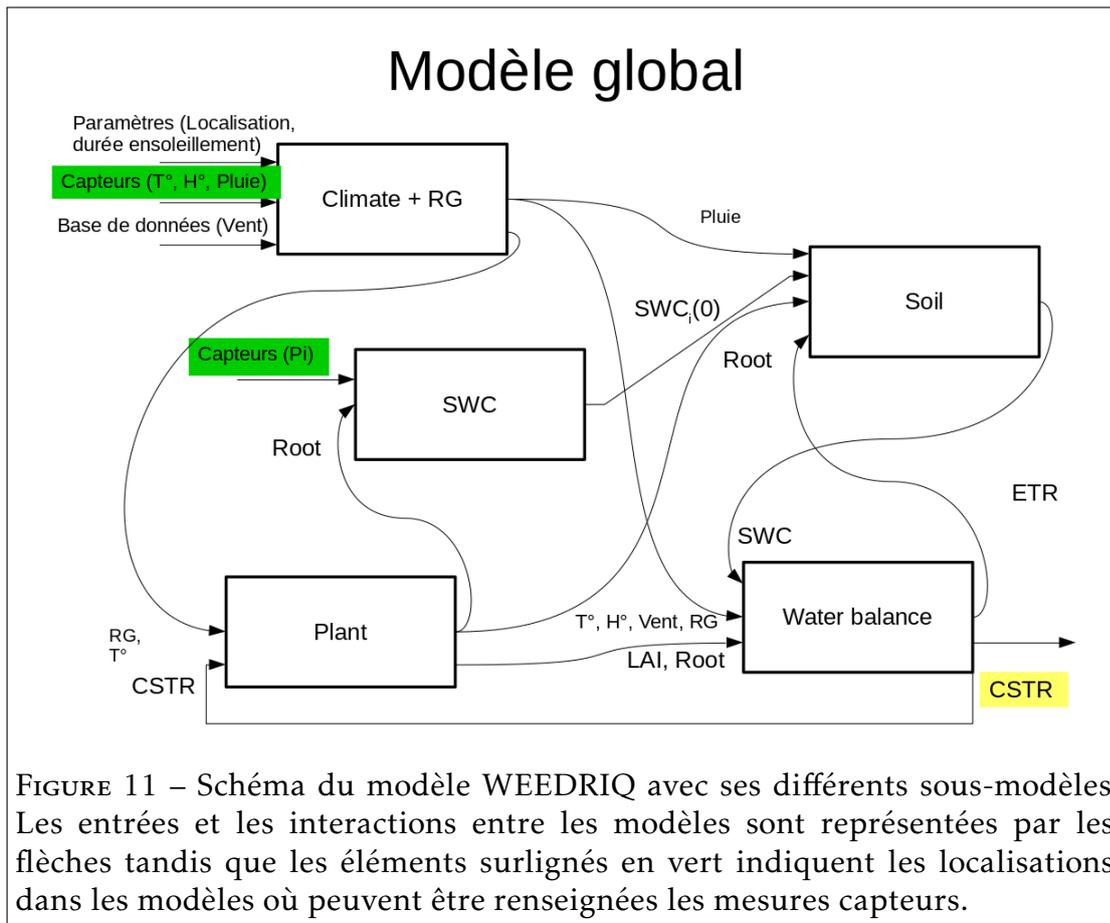


FIGURE 11 – Schéma du modèle WEEDRIQ avec ses différents sous-modèles. Les entrées et les interactions entre les modèles sont représentées par les flèches tandis que les éléments surlignés en vert indiquent les localisations dans les modèles où peuvent être renseignées les mesures capteurs.

...). L'avantage du calendrier Julien (version commençant au 24 4714 novembre avant Jésus-Christ) et de pouvoir mesurer la différence entre 2 dates par une simple soustraction (dans l'hypothèse de connaissance des dates juliennes pour ces 2 jours).

SWC

Au sein de l'automate WEEDRIQ, tous les modèles utilisant la teneur en eau pour leurs simulations ont besoin d'avoir cette donnée exprimée en mm. Cette valeur a été obtenue, durant les travaux de la thèse, par l'intermédiaire des capteurs tensiométriques de WEENAT. Malheureusement les capteurs tensiométriques renseignent sur le potentiel hydrique du sol (exprimé en kPa) et non sur sa teneur en eau (en mm). Le modèle SWC intervient ici pour résoudre ce problème. Il transforme les valeurs des potentiels hydriques des capteurs, c'est-à-dire les valeurs réellement mesurées, en une estimation de la teneur en eau (approximation). Cette transformation est rendue possible grâce à l'équation de Van Genuchten [132]. Cette formule est détaillée par l'équation 1. À partir d'un potentiel hydrique P (kPa), des paramètres de réglages, dont les valeurs, laissées à la discrétion de l'opérateur, sont $\alpha(L^{-1})$, n (supérieur à 1, sans unité), un volume résiduel d'eau $\theta_r(m^3m^{-3})$ et un volume maximal d'eau lorsque le sol est saturé $\theta_s(m^3m^{-3})$. L'équation de Van Genuchten permet d'obtenir une approximation du SWC. Les paramètres θ_s et θ_r sont fournis par la caractérisation du sol (voir le triangle de texture figure 12).

$$SWC(P) = \theta_r + \frac{(\theta_s - \theta_r)}{[1 + (\alpha \times |P|)^n]^{1-\frac{1}{n}}} \quad (1)$$

Cette grandeur est calculée pour chaque profondeur afin d'obtenir un volume global. Il est donc nécessaire d'obtenir plusieurs points de mesure à différentes profondeurs. Plus le nombre de points de mesure est élevé, meilleure sera l'approximation du volume global. Dans le contexte de cette thèse, 3 profondeurs (20, 30 et 40 cm) sont considérées par mesure. Ces profondeurs correspondent aux profondeurs racinaires des pommes de terre dans la région des Haut-de-France et donc des zones où les racines peuvent puiser l'eau.

Le calcul du volume global est donné par l'équation 2 à l'instant t :

$$SWC_t = \sum_{P=1}^3 SWC_t(P) \quad (2)$$

SOIL

Le modèle SOIL permet de représenter le système du sol ainsi que les réserves disponibles en eau de chaque couche à un instant t et enfin de répartir le volume initial d'eau entre les couches. La modélisation du sol est assimilée à une superposition de couches indépendantes. Afin d'alléger le modèle, une hypothèse forte est posée : toutes les couches possèdent la même composition en argile, en terreau, en limons et en sable. Cette composition est appelée texture et cette combinaison est reportée dans un triangle de texture [32]. De nombreuses représentations des triangles de texture ont été proposées souvent pour répondre à des problématiques régionales [48]. Concernant WEEDRIQ, le triangle de texture utilisé est celui de Mathieu [94], actuellement reconnu comme étant le plus pertinent pour une utilisation sur des sols de France métropolitaine (figure 12).

Le triangle de texture définit ainsi les caractéristiques intrinsèques du sol comme la capacité maximale d'eau qu'une couche peut contenir (capacité au champ), la vitesse de percolation (capacité de l'eau de s'infiltrer dans les couches inférieures), ... À l'initialisation de WEEDRIQ, le modèle SOIL a besoin d'avoir un volume d'eau représentant l'état hydrique initial. Pour que la simulation soit correctement réalisée, il faut que la valeur du volume global initial d'eau soit exactement la même que celle mesurée par les capteurs. Comme expliqué avant, les unités de mesures entre les valeurs capteurs et celles utilisées par les modèles de WEEDRIQ étant différentes, il faut employer le modèle SWC pour faire correspondre les grandeurs. La valeur initiale est appelée SWC_0 et est donc exprimée en mm. La valeur SWC_0 est ensuite équitablement distribuée selon les différentes couches. Au cours de la simulation, le modèle WATER BALANCE va agir sur les différentes couches du sol et modifier cette valeur de teneur en eau présente dans le sol. Cette dernière, appelée SWC_TNT_i (ex-

primée en mm) représente une estimation de l'évolution de SWC_0 au i ème jour de la simulation. Ainsi il devient aisé pour l'exploitant agricole d'avoir une information facilement interprétable sur les réserves hydriques du sol de ses parcelles à différentes profondeurs.

WATER BALANCE

Une fois la répartition de l'eau dans le sol effectuée, le modèle *WATTER BALANCE* règle les bilans hydriques s'opérant d'une part entre les couches observées et de l'autre avec la surface par remontée capillaire. Ce modèle calcule la quantité d'eau perdue par transpiration des plantes chaque jour suivant l'équation 3 donnée par la FAO, où Kc est un coefficient cultural donné par la *Food and Agriculture Organization* (FAO) [43], l'évapotranspiration (ETO) est donnée par l'équation de Penman-Montheith [44] suivant la méthode de la FAO [63].

$$Transpiration = Kc * ETO \quad (3)$$

Cette quantité d'eau est ensuite enlevée des couches en partant de la 1re couche, celle en contact avec la surface. Si cette quantité à prélever est supérieure à la couche actuelle, alors le prélèvement s'opère aussi sur la suivante et ainsi de suite jusqu'au moment où le déficit est comblé. La résolution du déficit est schématisée par l'algorithme 1.

WATTER BALANCE gère aussi les apports en eau. Ces derniers peuvent survenir avec 2 événements :

- Irrigations et précipitations. Ce cas de figure est le plus courant. L'eau est apportée dans le modèle à partir de la première couche.
- Remontées capillaires issues de remontées d'eau des nappes phréatiques. Plus rares que le premier cas de figure, mais prises en compte par le modèle. L'eau est apportée cette fois dans le modèle à partir de la dernière couche. Les apports, ayant été attribués dans leurs couches initiales, vont subir une translation dans les couches. En effet si la quantité d'eau est plus importante que *capacité au champ* (paramètre défini grâce au triangle des textures) alors l'eau va soit percoler et donc remplir les couches inférieures soit opérer une remontée capillaire depuis la nappe

```

Input :  $D$  : déficit en eau
 $c \leftarrow couche[0]$ ;
while  $D > 0$  do
  ;
   $disponible \leftarrow Disponibilite(c)$ ;
  if  $disponible > D$  then
    |  $disponible \leftarrow disponible - D$ ;
    |  $D = 0$ ;
  else
    |  $disponible \leftarrow 0$ ;
    |  $D \leftarrow D - disponible$ ;
    |  $c \leftarrow coucheSuiivante$ ;
  end
end

```

Algorithme 1 : Calcul du déficit d'eau

phréatique vers la surface. Un dernier cas de figure apparaît lorsque l'apport en eau par irrigation ou précipitation est supérieur à un seuil d'infiltration¹¹. Alors l'eau stagne et forme une flaque d'eau en surface. L'eau pourra donc mettre plusieurs jours à totalement pénétrer dans le sol. L'algorithme 2 explique le mécanisme de l'apport en eau dans les différentes couches du sol.

PLANT

Ce modèle simule le développement et les besoins d'un plant de tubercules depuis sa plantation jusqu'à la sénescence¹². Le modèle décrit un plant de pommes de terre comme étant un assemblage de 4 compartiments :

- Feuilles
- Tiges
- Racines
- Tubercules

11. Quantité d'eau journalière que la couche de surface peut faire pénétrer de l'eau dans le sol

12. Étape du développement d'un plant de tubercules, où la plante commence à mourir pour fournir toutes ses ressources aux tubercules.

Input : *AS* : Apport en eau depuis la surface, *AP* : Apport en eau depuis la nappe phréatique

Output :

coucheS \leftarrow *disponibilite_premiere_couche*(*Couche*);

coucheP \leftarrow *disponibilite_derniere_couche*(*Couche*);

flaque \leftarrow 0 **while** *AS* > 0 & *AP* > 0 **do**

 Creation de flaques d'eau;

if *AS* > (*capacite_max*(*coucheS*) - *coucheS*) **then**

flaque \leftarrow (*capacite_max*(*coucheS*) - *coucheS*)

AS \leftarrow *AS* - (*capacite_max*(*coucheS*) - *coucheS*)

end

if *AS* > 0 **then**

if *AS* < *coucheS* **then**

coucheS \leftarrow *coucheS* + *AS*;

AS \leftarrow 0;

else

coucheS \leftarrow *capacite_max*(*coucheS*);

AS \leftarrow *AS* - (*capacite_max*(*coucheS*) - *coucheS*);

coucheS \leftarrow *prochaine_couche*(*coucheS*);

end

end

if *AP* > 0 **then**

if *AP* < *coucheP* **then**

coucheP \leftarrow *coucheP* + *AS*;

AS \leftarrow 0;

else

coucheP \leftarrow *capacite_max*(*coucheP*);

AS \leftarrow *AS* - (*capacite_max*(*coucheP*) - *coucheP*);

coucheP \leftarrow *prochaine_couche*(*coucheP*);

end

end

end

Algorithme 2 : Algorithme du modèle WATTER BALANCE de l'apport en eau.

TABLEAU 3 – Table résumant les différents stades de développement d'un plant de tubercules pris en charge par le modèle PLANT.

Stade	Observations
Germination	Début de formation des racines
Levée	Début de formation de la tige et des feuilles
Tubérisation	Début de formation des tubercules
Sénescence	Mort de la plante au profit des tubercules

Les équations utilisées sont celles de Johnson [69] pour simuler les mécanismes mis en place par le tubercule pour acquérir de l'énergie puis l'utiliser ou la transformer en Matière sèche (MS). La MS est ensuite envoyée dans les différents compartiments en fonction de paramètres du modèle PLANT. L'allocation est aussi soumise au stade de développement de la plante. En effet, par exemple, un plant de tubercules ne génère de nouveaux tubercules qu'à partir de la tubérisation¹³ et ne pourra pas allouer de MS avant ce moment. Le stade de développement d'une plante est exclusivement contrôlé par son âge. Plus l'âge est avancé, plus le développement est avancé. Cependant les plantes ne "grandissent" pas de la même façon que les animaux. L'âge d'une plante est exprimé en $degre.Jour^{-1}$ et cette valeur est mesurée par rapport à une base [73]. Pour la pomme de terre cette base est de 6 degrés. L'âge d'une plante est donc la somme des températures supérieures à 6 degrés depuis la date de plantation. L'équation 4 explique le calcul où n est le nombre de jours écoulés depuis la date de plantation.

$$Age = \sum_{i=1}^n f(x_i) \quad (4)$$

$$f(x) = \begin{cases} Temp(i) & \text{if } Temp(i) > 6 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

L'âge calculé permet de connaître dans quelle phase de développement la plante se situe. Il existe de nombreux stades [95]. Le modèle plante en utilise les principaux qui sont reportés dans le tableau 3.

L'eau est le carburant nécessaire au bon fonctionnement de la mécanique de

13. Phase du développement d'un plant de tubercules

entrées/sorties respectives, l'implémentation du modèle se sert d'interfaces en C++. L'algorithme 3 présente de manière très simplifiée le déroulement d'une simulation ainsi que des valeurs de sortie accumulées chaque jour de la simulation. Toutes les valeurs ne sont pas représentées ici. En effet le modèle est capable de simuler plusieurs composantes comme la surface foliaire ou Leaf Area Index (LAI), le rendement du tubercule, le détail de la teneur en eau selon plusieurs profondeurs, ... Dans le cadre de cette thèse seule la sortie SWC (Soil Water Content) a été prise en compte puisqu'elle est, avec les travaux réalisés, la seule valeur qui peut être comparée avec les mesures capteurs par l'intermédiaire de la transformation par l'équation de van Genuchten [132].

Input : T : durée de la simulation

Output : $CSTR$: tableau de T coefficients de stress hydrique
 SWC : tableau de T teneur en eau

Initialisation;

$CSTR \leftarrow \emptyset$;

$SWC \leftarrow \emptyset$;

$t \leftarrow 0$;

$meteo \leftarrow lire_meteo(T)$;

$SWC_0 \leftarrow calcul_SWC(t)$;

$couches \leftarrow repartition_couche(SWC_0)$;

Boucle principale;

while $t < T$ **do**

$z \leftarrow calcul_profondeur_racinaire(t)$;

$bilan_hydrique_t \leftarrow calcul_deficit(meteo_t, z)$;

$developpement_plante(couche(z))$;

$couches \leftarrow mise_a_jour_couches(bilan_hydrique_t)$;

$CSTR \leftarrow SWC \cup calcul_CSTR()$;

$SWC \leftarrow SWC \cup calcul_SWC(t)$;

$t \leftarrow t + 1$;

end

Algorithme 3 : Algorithme expliquant l'ordonnement des sous-modèles afin de simuler le développement d'un plant de tubercules ainsi que son environnement.

Objectifs

L'ambition de cette thèse est la création d'apports scientifiques aux problématiques rencontrées dans le monde réel. Pour ce faire une identification des différents verrous scientifiques a permis d'axer les recherches documentaires de la littérature afin d'apporter des contributions. Un autre aspect de cette thèse est la valorisation et la mise en production des fruits de cette recherche. Dans ce contexte, le cadre opérationnel étant vaste, les travaux de cette thèse se sont concentrés sur les problématiques de la culture de pomme de terre. Plus précisément, les travaux visent 3 objectifs :

- Concevoir une méthodologie d'étalonnage automatique des paramètres du modèle WEEDRIQ par transformation du choix de paramétrage du modèle en problème d'optimisation continue multimodale boîte noire afin de trouver un paramétrage plus performant.
- Développer un nouvel algorithme d'optimisation continue multimodale boîte noire et étudier ses performances face à d'autres algorithmes état de l'art. D'un point de vue de recherche fondamentale, le but est de proposer un nouvel algorithme d'optimisation continue mono-objectif multimodale capable de gérer des problèmes d'optimisation complexe de grandes dimensions dans un contexte boîte noire en utilisant la problématique du bandit manchot comme une stratégie de *restart*.
- Reformuler le problème de la prédiction de la teneur en eau en un problème d'apprentissage supervisé. À cette fin la modélisation de l'évolution de la teneur en eau est réalisée par l'utilisation de méthode d'apprentissage automatique. La problématique du choix des caractéristiques est aussi étudiée et l'application de différentes méthodes de sélection de caractéristiques est présentée.

Les travaux rapportés dans cette thèse sont organisés en 3 parties. La première partie présente un panorama de la littérature pour les problèmes d'optimisations ainsi que les contributions scientifiques dans la résolution de problèmes multimodaux. Le chapitre 1 définit les problèmes d'optimisation continue, présente leurs diverses typologies et introduit une variété de techniques d'analyse et de résolution. Le chapitre 2 présente une méthodologie d'étalonnage

de modèles biologiques complexes. La deuxième partie est centrée sur l'apprentissage par renforcement. Après un tour d'horizon de l'apprentissage par renforcement ainsi que de la présentation du compromis exploration/exploitation et de la méthode de l'Upper Confidence Bound (UCB) pour gérer ce compromis dans le chapitre 3, le chapitre 4 détaille 2 contributions portant sur un nouvel algorithme d'optimisation (et son amélioration) continue mono-objectif multimodale boîte noire dont le choix des points initiaux des recherches locales est assisté par un algorithme d'apprentissage par renforcement. Enfin la troisième partie est orientée sur l'apprentissage automatique supervisé. Le chapitre 5 rappelle les rudiments de l'apprentissage automatique dans un contexte supervisé ainsi que divers algorithmes utilisés au cours de cette thèse et le chapitre 6 expose une contribution scientifique portant sur une méthode de prédiction hebdomadaire de la teneur en eau dans le sol par des algorithmes d'apprentissage supervisé.

Bibliographie du présent chapitre

- [1] *Agriculture et Territoire* Chambre d'agriculture du Nord-Pas-de-Calais. <https://nord-pas-de-calais.chambre-agriculture.fr/pommes-de-terre/>. Accessed : 2020-05-6.
- [2] Sajjad AHMAD, Ajay KALRA et Haroon STEPHEN. « Estimating soil moisture using remote sensing data : A machine learning approach ». In : *Advances in Water Resources* 33.1 (2010), p. 69-80.
- [6] Richard G ALLEN et al. « Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56 ». In : *Fao, Rome* 300.9 (1998), p. D05109.
- [15] *bnpe eau de France*. <https://bnpe.eaufrance.fr/acces-donnees/codeDepartement/59/annee/2017/usage/2>. updated : 2020-01-16.
- [18] *brgm*. <https://www.brgm.fr/publication-presse/etat-nappes-eau-souterraine-1er-mai-2019>. published : 2019-05-01.
- [19] N BRISSON et al. « An overview of the crop model stics ». In : *European Journal of Agronomy* 18.3 (2003). Modelling Cropping Systems : Science, Software and Applications, p. 309-332. ISSN : 1161-0301. DOI : [https://doi.org/10.1016/S1161-0301\(02\)00110-7](https://doi.org/10.1016/S1161-0301(02)00110-7). URL : <http://www.sciencedirect.com/science/article/pii/S1161030102001107>.
- [23] Dachun CHEN et Yingjie MA. « Optimized algorithm for estimating parameters by solving Van Genuchten equation based on stochastic particle swarm optimization ». In : *Nongye Gongcheng Xuebao(Transactions of the Chinese Society of Agricultural Engineering)* 22.12 (2006), p. 82-85.
- [26] U.S departement COMMERCE. *Official gazette of the United States patent and trademark office*. Sous la dir. d'Authority of CONGRESS. T. 103. 1981.
- [28] *Crop life 2019 Precision Agriculture Dealership Survey : More Moves Toward Decision Agriculture*. <https://www.croplife.com/management/2019-precision-agriculture-dealership-survey-more-moves-toward-decision-agriculture/>. published : 2019-08-21.
- [29] Conseil DÉTAT. *Rapport annuel 2010 du Conseil d'État (données consommations)*. 2010.
- [30] T.M. DARWISH et al. « Nitrogen and water use efficiency of fertigated processing potato ». In : *Agricultural Water Management* 85.1 (2006), p. 95-104. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2006.03.012>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377406001028>.

- [32] A DELAGE et H LAGATU. « Sur les résultats obtenus par l'observation des terres arables en plaques minces ». In : *Compt. Rend.* 140 (1905), p. 1555.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.
- [39] Michael D DUKES, Lincoln ZOTARELLI et Kelly T MORGAN. « Use of irrigation technologies for vegetable crops in Florida ». In : *HortTechnology* 20.1 (2010), p. 133-142.
- [40] EOS. *soil moisture control is an essential farming constituent*. 2019. URL : <https://eos.com/blog/soil-moisture-control-is-an-essential-farming-constituent/>.
- [41] Donald C ERBACH. « Measurement of soil bulk density and moisture ». In : *Transactions of the ASAE* 30.4 (1987), p. 922-931.
- [42] Michel Crétenet ERIC JALLAS. « Aide à la décision en agriculture et gestion des ressources naturelles : note introductive. Savanes africaines : des espaces en mutation, des acteurs face à de nouveaux défis. » In : *HAL* hal-00143508 (2003). URL : <https://hal.archives-ouvertes.fr/hal-00143508/document>.
- [43] FAO. *FAO kc*. Accessed : 2020-05-12. URL : <http://www.fao.org/3/x0490e/x0490e0b.htm>.
- [44] FAO. *FAO Penman-Monteith equation*. Accessed : 2020-05-12. URL : <http://www.fao.org/3/x0490e/x0490e06.htm#chapter%20%20%2020fao%20penman%20monteith%20equation>.
- [45] FAO. *Sprinkler irrigation*. URL : <http://www.fao.org/3/S8684E/s8684e06.htm>.
- [48] A Richer de FORGES et al. « Perdue dans le triangle des textures ». In : *Etudes et Gestion des Sols* 15.2 (2008), p. 97-111.
- [50] Montesano FRANCESCO et al. *Modern technologies, strategies and tools for sustainable irrigation management and governance in Mediterranean Agriculture*. Sept. 2015. ISBN : 2-85352-549-X.
- [51] Michel VALLANCE FRANÇOIS COLAS-BELCOUR Roland RENOULT. *Rapport n° 14061 : Synthèse Eau et Agriculture*. Rapp. tech. Ministère de l'agriculture de l'agroalimentaire et de la forêt, 2015.
- [63] Terry HOWELL et Steve EVETT. « The Penman-Monteith Method ». In : (jan. 2004).
- [65] INSEE. *Insee Références, édition 2016 - Fiches - Marché du travail*. 2016.

- [66] *itk Mission Intelligence Artificielle*. <http://www.itk.fr/wp-content/uploads/2018/04/ITK-Intelligence-Artificielle-Agriculture.pdf>. published : 2017-11-28.
- [69] K.B. JOHNSON, S.B. JOHNSON et P.S. TENG. « Development of a simple potato growth model for use in crop-pest management ». In : *Agricultural Systems* 19.3 (1986), p. 189-209. ISSN : 0308-521X. DOI : [https://doi.org/10.1016/0308-521X\(86\)90052-1](https://doi.org/10.1016/0308-521X(86)90052-1). URL : <http://www.sciencedirect.com/science/article/pii/0308521X86900521>.
- [73] Valentin KIENY. *Température & agriculture*. http://infoclimat.org/MeteoALecole/ressources_scientifiques/. Accessed : 2020.
- [78] Élise Deniel LAURENCE MENET Solenn Leplay. *Économiser leau en changeant les pratiques agricoles : retours d'expériences en Europe*. Rapp. tech. Ministère de l'agriculture et de l'alimentaire, 2018.
- [83] Hal LEMMON. « COMAX : An expert system for cotton crop management ». In : *Science* 233.4759 (1986), p. 29-33.
- [86] Jérôme LEROY. *site web weenat*. Accessed : 2020-05-12. URL : <https://weenat.com/>.
- [87] *Les aides Le site d'information des CCI sur les aides aux d'entreprises*. <https://les-aides.fr/fiche/a5Z1DH1GwPjMB3ZQ/conseil-regional/aide-aux-investissements-dans-les-infrastructures-liees-a-l-irrigation-agricole.html>. Accessed : 2020-05-06.
- [89] C. D. van LOON. « The effect of water stress on potato growth, development, and yield ». In : *American Potato Journal* 58 (1981). URL : <https://doi.org/10.1007/BF02855380>.
- [94] C MATHIEU et al. « Physical Analysis of Soils : Selected Methods ». In : *Lavoisier TEC & DOC, PARIS ed* (1998).
- [95] Uwe MEIER. *Entwicklungsstadien mono- und dikotyle Pflanzten : BBCH Monografie*. Quedlinburg : Open Agrar Repository, 2018. ISBN : 978-3-95547-070-8. DOI : 10.5073/20180906-075119. URL : https://www.openagrar.de/rsc/thumbnail/openagrar_mods_00042352.png.
- [97] *Ministere de l'Agriculture et de l'Alimentation Économiser leau pour l'irrigation par les changements de pratiques agricoles*. <https://agriculture.gouv.fr/economiser-leau-pour-lirrigation-par-les-changements-de-pratiques-agricoles>. published : 2018-07-03.

- [98] *Ministere de l'Agriculture et de l'Alimentation* <https://agriculture.gouv.fr/afclim-agriculture-foret-climat-vers-des-strategies-dadaptation-0>. <https://agriculture.gouv.fr/economiser-leau-pour-lirrigation-par-les-changements-de-pratiques-agricoles>. published : 2019-06-16.
- [99] FRANCESCO F. MONTESANO et al. « Tensiometer-Based Irrigation Management of Subirrigated Soilless Tomato : Effects of Substrate Matric Potential Control on Crop Performance ». In : *Frontiers in Plant Science* 6 (2015), p. 1150. ISSN : 1664-462X. DOI : 10.3389/fpls.2015.01150. URL : <https://www.frontiersin.org/article/10.3389/fpls.2015.01150>.
- [104] PACHEPSKY, YA. A., D. TIMLIN, AND G. VARALLYAY. « Artificial Neural Networks to Estimate Soil Water Retention from Easily Measurable Data. » In : *Soil Sci. Soc. Am. J.* 60 (1998), p. 727-733. DOI : 10.2136/sssaj1996.03615995006000030007.
- [105] Neelmadhab PADHY et Suresh Chandra SATAPATHY. « Digital Advancement in AgriTech by Using Evolutionary Techniques ». In : *Smart Intelligent Computing and Applications*. Springer, 2020, p. 361-368.
- [108] Khanh PHAM et al. « Analysis of neural network based pedotransfer function for predicting soil water characteristic curve ». In : *Geoderma* 351 (2019), p. 92-102. ISSN : 0016-7061. DOI : <https://doi.org/10.1016/j.geoderma.2019.05.013>. URL : <http://www.sciencedirect.com/science/article/pii/S0016706118315179>.
- [110] Dirk RAES et al. « AquaCrop—the FAO crop model to simulate yield response to water : II. Main algorithms and software description ». In : *Agronomy Journal* 101.3 (2009), p. 438-447.
- [111] Eric RAMAT et Bertrand VANDORNE. *Plant growth model for decision making support*. Rapp. tech. Université du Littoral Côte d'Opale, et ISA Lille, 2002.
- [114] David G ROSSITER. « ALES : a framework for land evaluation using a microcomputer ». In : *Soil use and management* 6.1 (1990), p. 7-20.
- [118] SCHAAP, M. G., F. J. LEIJ, AND M. TH. VAN GENUCHTEN. « Neural Network Analysis for Hierarchical Prediction of Soil Hydraulic Properties. » In : *Soil Sci. Soc. Am. J.* 62 (1998), p. 847-855. DOI : 10.2136/sssaj1998.03615995006200040001.
- [121] *Service Public Restrictions d'eau : quels sont les départements concernés*. <https://www.service-public.fr/particuliers/actualites/A13500>. published : 2019-09-18. 2019.

- [124] Soroosh SOROOSHIAN, Qingyun DUAN et Vijai Kumar GUPTA. « Calibration of rainfall-runoff models : Application of global optimization to the Sacramento Soil Moisture Accounting Model ». In : *Water Resources Research* 29.4 (1993), p. 1185-1194. DOI : 10.1029/92WR02617. eprint : <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/92WR02617>. URL : <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/92WR02617>.
- [128] Martin THALHEIMER. « Tensiometer modification for diminishing errors due to the fluctuating inner water column ». In : *Soil Science Society of America Journal* 67.3 (2003), p. 737-739.
- [132] M Th VAN GENUCHTEN. « A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1 ». In : *Soil science society of America journal* 44.5 (1980), p. 892-898.
- [138] Sevim Seda YAMAÇ et Mladen TODOROVIC. « Estimation of daily potato crop evapotranspiration using three different machine learning algorithms and four scenarios of available meteorological data ». In : *Agricultural Water Management* 228 (2020), p. 105875.
- [141] Fedro S ZAZUETA et Jiannong XIN. « Soil moisture sensors ». In : *Soil Sci* 73 (1994), p. 391-401.

Première partie

Problèmes d'optimisation et contributions dans la résolution de problème multimodal

Problèmes d'optimisation

Sommaire du présent chapitre

1.1 Problèmes d'optimisation	38
1.2 Typologies d'optimisation	39
1.2.1 Classes d'optimisation	40
1.2.2 Optimisation hors ligne	41
1.2.3 Optimisation dynamique	42
1.2.4 Dérivabilité de la fonction objectif	42
1.2.5 Connaissance analytique du problème	44
1.2.6 Fonction multimodale	44
1.3 Méta heuristique	45
1.3.1 Stratégies d'évolution	46

Introduction Ce chapitre introduit les problèmes d'optimisation. Il se divise en trois parties. La première partie définit formellement les problèmes d'optimisation. La seconde présente différentes typologies associées aux problèmes d'optimisation en se focalisant sur les problèmes numériques. La dernière partie détaille les méthodes de résolution de problèmes d'optimisation à l'aide des méta heuristiques. Après une courte définition, un panorama d'algorithmes de la littérature pour la résolution de problème d'optimisation numérique boîte

noire utilisant une stratégie d'évolution utilisant une stratégie de *restart* est présenté.

1.1 Problèmes d'optimisation

Résoudre un sudoku, ranger ses affaires dans sa valise, trouver les proportions idéales pour réaliser son pain, définir le chemin le plus court entre deux villes ... Voilà bien quelques exemples de problèmes du quotidien dont la résolution est non triviale. Afin de les résoudre, un formalisme possible consiste à reformuler mathématiquement le problème rencontré sous forme d'un modèle. Ainsi l'ensemble des choix possibles (par exemple : l'ensemble des combinaisons de rangement possible des affaires dans le sac, les quantités d'ingrédients de la recette du pain ...) forment *l'espace de recherche*. Cependant toutes les solutions ne se valent pas forcément, certaines étant plus pertinentes que d'autres en fonction d'un critère d'évaluation. Ce critère défini par la personne voulant résoudre ce problème est appelé *fonction objectif*¹ et attribue une note pour chaque solution de l'espace de recherche. Cette note est arbitraire et reflète la pertinence d'une solution. Lorsque la fonction objective consiste à répondre au(x) critère(s) d'un seul objectif, la solution satisfaisant le mieux l'objectif est alors appelée solution optimale ou *optimum*. Il est possible que les solutions ne puissent pas être comparées, dans ce cas la réponse au problème est un ensemble de solutions. En prenant l'exemple de la recette du pain, on s'aperçoit qu'une solution fait intervenir différents ingrédients tels que l'eau, la farine, le sel, la levure, ... Ces éléments forment alors la *dimension* du problème. Une solution peut alors être exprimée comme un vecteur où chaque nombre réel correspond à une quantité d'un ingrédient. Cette modélisation a pour but d'opérer une abstraction de la réalité et donc de simplifier un problème parfois complexe à appréhender (réduction du bruit, des défauts, des biais ...). La conception d'un modèle adéquat repose sur une bonne connaissance experte du domaine considéré ainsi qu'une connaissance des méthodes de résolution. Dès lors, la définition du problème originel en un problème mathématique est faite, il est

1. Aussi appelé fonction de coût

alors possible de considérer sa résolution comme un problème d'optimisation. De manière formelle un problème d'optimisation peut être défini par :

- Un espace de recherche χ représentant l'ensemble des solutions possibles.
- Une fonction objective f étant le critère d'évaluation d'une solution tel que : $f : \chi \rightarrow \mathbb{R}$

Ainsi résoudre un problème d'optimisation consiste à trouver la meilleure (en minimisation ou maximisation) solution $x \in \chi$ en fonction de la fonction objective f telle que définie par l'équation 1.1 (ici en minimisation) où x est une solution appartenant à l'espace de recherche χ .

$$x^* = \operatorname{argmin}(f) \quad (1.1)$$

1.2 Typologies d'optimisation

Il existe de nombreuses applications utilisant les problèmes d'optimisation. Cependant il est possible de les classer en 2 grandes catégories :

- Mono-objectif. La fonction objective consiste à satisfaire 1 critère (ou bien une somme de plusieurs critères).
- Multi objectif. La fonction objective consiste à satisfaire plusieurs critères conjointement.

Chacune de ces catégories possède des typologies (optimisation combinatoire/numérique, boîte noire/ boîte blanche, optimisation dynamique ou non, multimodalité, ...) qui leur sont propres et influencent le choix des méthodes de résolution. Les travaux de cette thèse ont porté exclusivement sur les problèmes d'optimisation mono-objectif. La suite de la section présente donc les différentes caractéristiques des problèmes d'optimisation continue multimodale mono-objectif rencontrés au cours de la thèse dans un contexte hors ligne et dynamique.

1.2.1 Classes d'optimisation

Les problèmes d'optimisation peuvent être représentés comme une fonction faisant correspondre les solutions d'un espace de recherche avec les résultats de la fonction objectif correspondants. En fonction du type de valeur des variables composant les solutions, le problème peut être classé dans 2 classes d'optimisation. Les travaux de la thèse ainsi que ses contributions n'ont porté que sur des problèmes d'optimisation numérique.

Optimisation combinatoire

L'espace de recherche est composé de solutions ne contenant que des variables discrètes (appartenant aux entiers). Il est donc potentiellement fini est dénombrable (même si les problèmes combinatoires peuvent être de cardinalité infinie). À ce titre, toutes les solutions pouvant être listées, il suffirait de sélectionner la meilleure pour résoudre le problème. Cette assertion n'est réalisable que si l'espace de recherche est fini et de faible dimension ou bien si l'on dispose d'un budget de temps conséquent [4] (pour certains problèmes le temps de dénombrement de toutes les solutions peut être plus grand que le nombre d'années depuis le début de l'univers). La plupart des problèmes combinatoires étant NP-difficile [106], l'utilisation de méthodes de recherche efficaces comme les méthodes exactes ou d'autres méthodes au moins meilleures que l'énumération de toutes les solutions possibles devient nécessaire [142]. Comme exemples d'applications des problèmes d'optimisation combinatoire, on peut citer :

- La coloration de graphe [21]
- Le réglage de feux tricolores pour la mobilité urbaine [85]
- Le pilotage de réacteur nucléaire [100]

Optimisation numérique

Ce genre d'optimisation est aussi appelée optimisation continue. À l'instar de l'optimisation combinatoire, l'ensemble des solutions ainsi que les valeurs de la fonction de coût appartiennent aux réels. Dans ce cas de figure, le dénombrement des solutions est impossible puisque les variables à optimiser peuvent

prendre tout un continuum de valeurs. Comme expliqué dans la sous-section précédente, la contrainte de temps est aussi importante : on souhaite déterminer une solution satisfaisante avec un budget de temps raisonnable. Ainsi les méthodes de résolution d'optimisation continue relèvent toutes de l'analyse des fonctions objectif (lorsque cela est possible) ou bien à l'aide d'algorithmes construisant des solutions tendant vers un optimum[82]. L'espace de recherche étant infini, l'arrêt du processus d'optimisation est défini par un critère d'arrêt pouvant être par exemple :

- Un seuil d'acceptabilité (réponse approchant l'optimum)
- La meilleure solution trouvée après l'écoulement d'un budget d'itérations

Le champ d'application des problèmes d'optimisation continue est très vaste. De manière non exhaustive, on peut citer comme domaine d'application :

- l'étalonnage de modèles biologiques [38]
- l'ingénierie [88]
- la finance [91]

1.2.2 Optimisation hors ligne

L'optimisation hors ligne est un sous-ensemble de l'optimisation. Dans ce domaine les algorithmes d'optimisation ont une connaissance complète de l'information sur laquelle ils se reposent pour effectuer la tâche d'optimisation. Aucune nouvelle information ou observation postérieure à la base des connaissances actuelles ne doit pouvoir, en théorie, dégrader ou remettre en cause le(s) solution(s)/ paramétrage(s) déjà déterminées durant le processus d'optimisation. Ce type d'optimisation est pertinent, par exemple, lorsque l'optimisation des paramètres d'un modèle nécessite de nombreuses simulations. C'est notamment le cas des méthodes de "back-analyse" [68, 71, 117]. En effet, l'algorithme a accès à toutes les informations ainsi que les potentielles informations futures et cherche ainsi le meilleur paramétrage d'un modèle permettant d'expliquer les informations à sa disposition. L'un des objectifs de l'optimisation hors ligne est d'étalonner des modèles n'ayant pas besoin de se corriger rapidement (un nouveau processus d'optimisation pouvant s'opérer après une période d'accumula-

tion de nouvelles informations afin de renforcer le paramétrage du modèle). Les méthodes d'apprentissage supervisé (section 5.1) peuvent être assimilées à de l'optimisation hors ligne puisque le modèle généré a été appris en amont sur des données sur lesquelles il a toute l'information (caractéristiques et label).

1.2.3 Optimisation dynamique

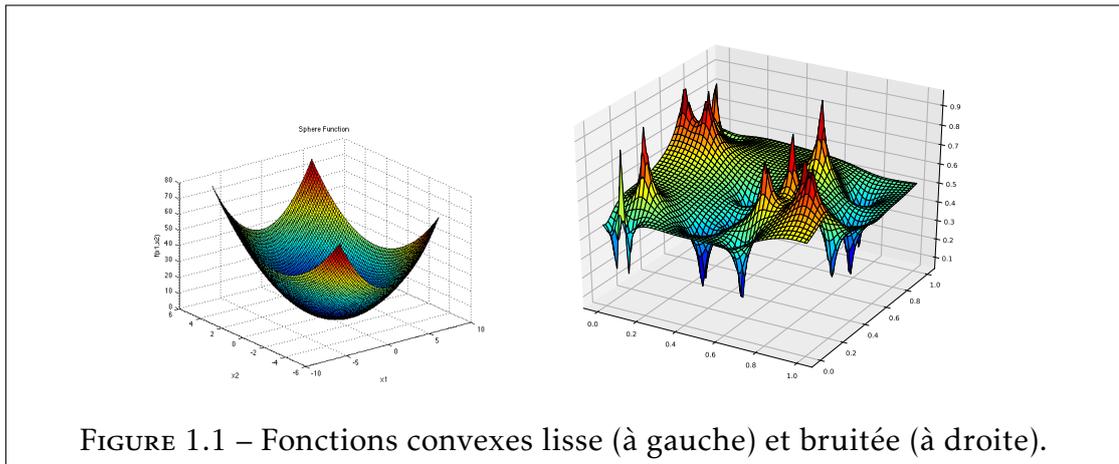
À l'opposé de cette approche se trouve l'optimisation dynamique ou en ligne. Cette approche part du postulat que l'algorithme n'ait qu'une connaissance partielle ou incomplète du système étudié. En effet, si le système étudié risque d'être (souvent) modifié par des événements inconnus, par exemple si la demande en énergie d'un pays est fortement affectée par un confinement général, alors le modèle doit pouvoir prendre des décisions à la volée faites sur les informations récoltées jusqu'à présent potentiellement combinées avec des hypothèses faites ou des connaissances partielles du futur (prévisions). On peut citer comme exemples, les modèles de contrôle prédictif [135, 67]. Dans ce cas l'optimisation ne doit pas excéder l'ordre de la minute [135]. Les modèles de locomotion des robots utilisent aussi ces algorithmes [27, 46]. L'apprentissage par renforcement (section 3.1) peut être assimilé à de l'apprentissage dynamique. En effet le modèle généré doit prendre une décision à partir d'informations accumulées et de connaissances partielles voir incomplètes pouvant être remises en question par des événements inconnus.

Ces 2 approches de l'optimisation sont complémentaires et répondent à des besoins spécifiques.

1.2.4 Dérivabilité de la fonction objectif

Une fonction est dérivable en un point a quand elle admet une dérivée finie en a , c'est-à-dire, quand elle peut être approchée de manière assez fine par une fonction affine au voisinage de a . Elle est dérivable sur un intervalle réel ouvert non vide si elle est dérivable en chaque point de cet intervalle. Plus formellement soit, I un intervalle non vide de \mathbb{R} , $f : I \rightarrow \mathbb{R}$ et $x_0 \in I$, on dit que f est dérivable en x_0 si $f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0+h) - f(x_0)}{h}$ est finie. Cette limite est appelée la dérivée de f .

Lorsque la fonction objective d'un problème d'optimisation est dérivable ce qui peut être possible dans le cadre de l'optimisation numérique et combinatoire, alors l'algorithme peut tirer parti de la connaissance du gradient (on parle alors de *gradient-based method*) pour trouver des solutions de meilleure qualité par rapport à la fonction de coût. En effet si l'on considère le cas d'une fonction de coût cherchant à minimiser l'erreur, un algorithme basé sur l'utilisation du gradient consiste à partir d'un point de départ (solution candidate) appartenant à l'espace de recherche de la fonction de coût, translater ce point dans la direction opposée au gradient afin de diminuer l'erreur et ainsi atteindre une solution de meilleure qualité à chaque itération sous réserve que le pas soit suffisamment petit. Cette technique est appelée descente de gradient. L'atout principal de cette technique est la possibilité d'optimiser de nombreux paramètres de la fonction objectif et donc de résoudre des problèmes de grande dimension tout en ayant recours à peu d'évaluations [134]. À noter qu'il existe d'autres méthodes d'optimisation de manière non exhaustive, on peut citer : adam, nadam, nesterov [115]. Malheureusement, l'avantage de la descente de gradient est aussi sa principale faiblesse. Lorsque la fonction à optimiser est multimodale, non dérivable, non convexe, ... (figure 1.2) la convergence vers une solution toujours de meilleures qualités peut l'obliger à converger vers une solution sous optimale (appelé optimum local). Il faut alors utiliser des techniques stochastiques (faisant appel à l'aléatoire) pouvant remédier à ce problème [16]. Les méta heuristiques (section 1.3) sont de bonnes candidates pour résoudre ces problèmes. Grâce à elles de nombreux problèmes issus du monde réel utilisant des fonctions de coût non dérivables, peuvent ainsi trouver une solution approximant la ou les solutions optimales. De manière non exhaustive, on peut citer le domaine de la chimie [34], biologie [55], de la médecine [93, 103], de l'ingénierie [58]. Comme il existe de très nombreux algorithmes d'optimisation, il est nécessaire de choisir le mieux adapté. Le critère principal pour le choix d'un algorithme dans la résolution d'un problème d'optimisation est le nombre d'évaluations nécessaires pour arriver à converger vers la solution optimale.



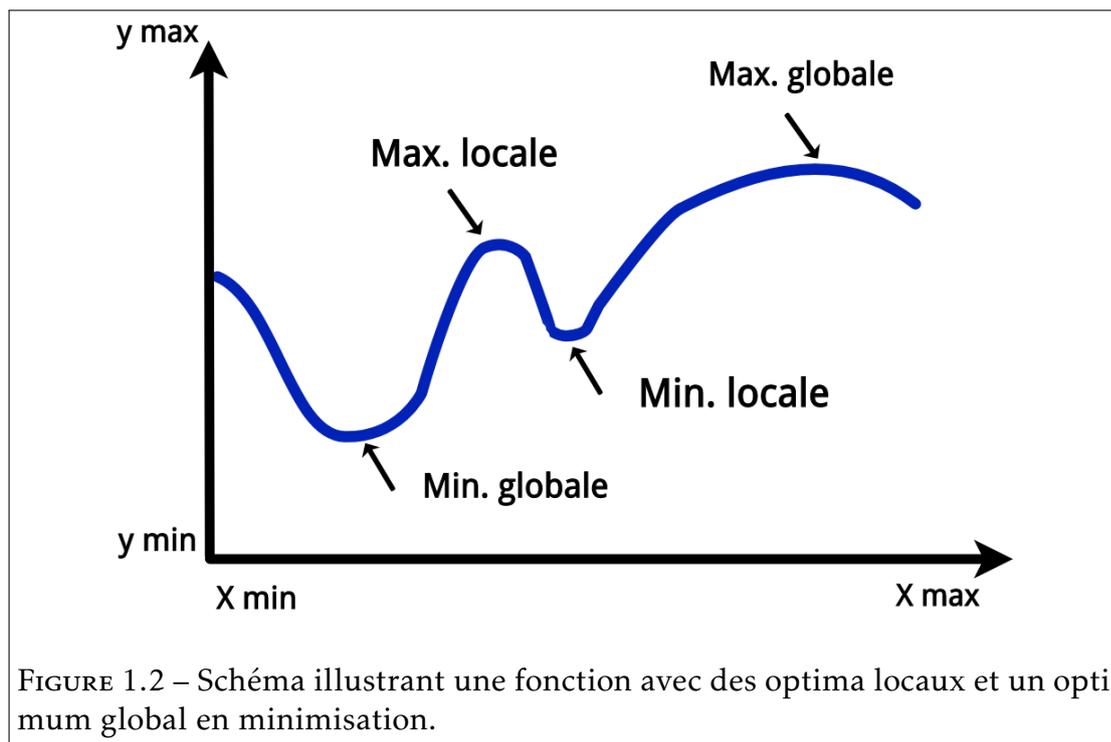
1.2.5 Connaissance analytique du problème

Quand la définition analytique de la fonction de coût est connue alors on parle d'optimisation *boîte blanche*. Dans ce contexte, la définition devient alors une information supplémentaire pour l'algorithme, il est alors possible d'utiliser ses caractéristiques algébriques afin de le résoudre plus efficacement (descente de gradient ...). L'optimisation *boîte noire* quant à elle, désigne toutes les fonctions objectives qui n'ont pas accès à cette définition formelle ou bien lorsque l'algorithme d'optimisation n'a pas nécessairement besoin de la connaître pour la résoudre. Ainsi la qualité d'une solution x n'est accessible qu'à travers la réponse d'un oracle $f(x)$. Ce dernier pouvant représenter le résultat d'une simulation, ou d'un calcul. L'algorithme n'a donc que la possibilité de connaître. $(x_0, f(x_0), (x_1, f(x_1))) \dots$

1.2.6 Fonction multimodale

Lorsqu'une fonction n'est pas convexe, elle possède un ou plusieurs optima locaux. Ces solutions correspondent à une qualité de réponse localement meilleure que les autres, mais potentiellement sous-optimale par rapport à la solution optimale (appelée aussi globale) (voir la figure 1.1). On parle de fonction multimodale dans 2 cas de figure :

- La fonction possède un optimum global, mais de nombreux optima locaux (dont certains peuvent avoir une qualité très proche de celle opti-



male). Dans ce cas il est nécessaire de découvrir le plus grand nombre afin de trouver la meilleure.

- la fonction possède plusieurs optima globaux, c'est-à-dire que plusieurs solutions ayant des paramètres plus ou moins différents répondent au mieux aux critères de la fonction objective.

Les problèmes provenant du monde réel sont souvent multimodaux [119].

1.3 Méta heuristique

Une heuristique est un algorithme de résolution dont la conception repose sur l'expérience de son concepteur. Une heuristique est un compromis entre une performance efficace et une possibilité d'utilisation. En effet, elle ne garantit pas d'obtenir une réponse optimale à un problème, cependant elle doit pouvoir le plus souvent fournir une réponse approximant la solution optimale tout en fournissant le moins souvent une solution de piètre qualité² et enfin,

2. Il faut comprendre ici que l'heuristique doit être le plus généralisable possible.

être facile à implémenter³. De plus leur temps d'exécution doivent être raisonnable. Puisqu'il est peu probable qu'une seule heuristique réponde à tous les problèmes d'optimisation, une combinaison de plusieurs heuristiques appelées méta heuristique a été proposée. Ces algorithmes peuvent ainsi résoudre des problèmes difficiles pour une simple heuristique. C'est le cas des optimisations continues boîte noire et/ou multimodale.

On distingue 2 grandes classes de méta heuristique : les algorithmes gérant 1 solution et les algorithmes gérant une population de solutions. Le reste de la section présente les différentes méta heuristiques utilisées durant cette thèse.

1.3.1 Stratégies d'évolution

Ces algorithmes sont inspirés de la théorie darwinienne de l'évolution. Selon ce principe, la pression exercée par un milieu permet aux solutions les plus adaptées de pouvoir survivre et de ce fait se reproduire. Dans son implémentation pour les algorithmes d'optimisation, la fonction objective sert d'environnement. La survie des solutions les mieux adaptées est déterminée par un opérateur de sélection. Enfin la reproduction est effectuée sur les solutions susceptibles de se reproduire par une méthode de variation aléatoire qui peut comprendre une étape dite de mutation et une étape dite de croisement. Les Evolution Strategy (ES) sont des méthodes stochastiques, puisqu'il y a au moins un de ses opérateurs qui utilise des processus aléatoires. Le fonctionnement peut être résumé ainsi : à partir d'une population de solutions, généralement tirée aléatoirement, un ensemble des meilleures (mieux adaptés) solutions est sélectionné. Ces solutions seront les parents de la nouvelle génération de la population grâce à l'opérateur de mutation (ou croisement). La notation générale des stratégies d'évolution est $(\mu + \lambda)$ -ES avec μ parents et λ enfants. Le remplacement des μ meilleurs est effectué parmi les $\mu + \lambda$ solutions. Il existe un cas particulier où la taille de la population est de 1 (noté (1+1)-ES) ce qui signifie que le meilleur individu entre le parent et son enfant est conservé.

Itérativement, et en accord avec le principe darwinien, l'ensemble de la population va tendre à devenir plus adapté aux critères de la fonction objective.

3. Dans sa version initiale tous du moins

L'algorithme 4 schématise le principe général régissant les algorithmes évolutionnaires. Les travaux de cette thèse se sont centrés sur l'étude des stratégies d'évolution pour répondre au problème de calibration du modèle WEEDRIQ dans un contexte boîte noire nous interdisant la connaissance du gradient. Le reste de la section présente 2 algorithmes état de l'art utilisés durant la thèse dans un contexte continu multimodal boîte noire. Le premier est un algorithme à population unique tandis que le second est un algorithme à population.

```

begin
   $P \leftarrow \text{initialisation\_alatoire}()$ 
  while critere d'arret non satisfait do
     $\text{parents} \leftarrow \text{SelectionMeilleur}(P)$ 
     $\text{enfants} \leftarrow \text{mutation}(\text{parents})$ 
     $P \leftarrow \text{remplacer}(P \cup \text{enfants})$ 
  end
end

```

Algorithme 4 : Algorithme de la stratégie d'évolution

(Quasi) Random with Decreasing Step size

Dans le contexte des problèmes d'optimisation multimodale boîte noire, la fonction objectif possède beaucoup d'optima globaux ou bien de nombreux optima locaux proches de l'optimum global. Le but consiste donc à découvrir le plus d'optima possible afin de sélectionner le ou les meilleurs a posteriori. Du fait de l'hypothèse de la boîte noire, l'utilisation de techniques basées sur le gradient est impossible. Les algorithmes Random with Decreasing Step size (RDS) [119] et Quasi Random with Decreasing Step size (QRDS) [127] ont été proposés afin de répondre à ce problème. Ce sont des algorithmes d'optimisation proposés pour répondre aux problèmes d'optimisation multimodale boîte noire. Ils utilisent le principe de la recherche locale à solution unique contrôlée par une stratégie de relance ou *restart*. La recherche locale peut être effectuée avec n'importe quel algorithme à stratégie d'évolution [12]. Dans sa version initiale, l'algorithme utilisé est un (1+1)-ES avec le step-size respectant la règle du "one-fifth rule success" [9]. Cette règle implémente l'idée que le step-size doit

augmenter si les sélections de la recherche locale se font sur les solutions candidates par rapport à la solution courante. Ceci suggère que la taille du voisinage est trop petite et doit donc être augmentée. Inversement, si la solution courante est plus souvent gardée que la solution candidate, alors l'échantillonnage des solutions candidates est trop large. Il faut donc réduire le step-size. De manière optimale, la probabilité d'échantillonner correctement une solution meilleure tend vers $1/5$ [113, 120].

L'algorithme 5 présente le principe de fonctionnement de la recherche locale SearchDS (utilisé dans les algorithmes RDS et QRDS). Cet algorithme fait évoluer itérativement les solutions candidates : il génère une nouvelle solution par mutation en utilisant une distribution normale avec un écart-type σ . Si le nouveau point est meilleur alors la valeur de σ est augmentée, car cela signifie que le point actuel est loin de la solution, ainsi la recherche doit être étendue à un voisinage plus grand. Dans le cas contraire, σ est diminué parce que cela signifie que le point actuel est proche de la solution et donc la recherche doit se focaliser sur un voisinage restreint. La meilleure solution (par rapport à la fonction de coût) entre le nouveau et l'ancien est conservée. La recherche est terminée si la solution converge vers les coordonnées d'un optimum déjà découvert ou bien lorsque la recherche converge à une distance ϵ d'un optimum (dans ce cas, les coordonnées de cette solution ou le point sont mémorisés). En utilisant cet algorithme de recherche local, une stratégie de restart est très simple à implémenter, comme le montre l'algorithme 6. À chaque itération, un point est sélectionné à travers l'espace de recherche. Ce dernier est utilisé comme point de départ d'une recherche locale. Quand la recherche locale est terminée, une nouvelle recherche locale commence si tous les optima n'ont pas été découverts et si le budget d'évaluation n'est pas écoulé. Schoenauer et al. [119] ont comparé les 2 algorithmes : RDS et QRDS. Ils ont démontré que l'échantillonnage du point de départ suivant une séquence quasi aléatoire (QRDS) plutôt qu'une loi uniforme (RDS) était plus efficace. Une caractéristique importante de QRDS est le "murder operator". Dans le but d'éviter de converger sur des optima déjà découverts, l'algorithme vérifie à chaque évaluation si la solution courante est à une distance (euclidienne) supérieure à un seuil ϵ_x de chaque optimum déjà mémorisé et donc découvert. Si c'est le cas, la recherche est arrêtée prématu-

rément sans sauvegarder la solution (l'idée est de ne pas gaspiller le budget d'évaluation pour des optima déjà découverts).

Covariance Matrix Adaptation Evolution Strategy

CMA-ES [59] (signifiant stratégie dévolution à matrice de covariance adaptative) est une stratégie d'évolution pour l'optimisation continue à destination des problèmes non linéaires ou non convexes. Contrairement aux méthodes à solution unique, CMA-ES utilise une population de solutions, ainsi le risque de convergence prématurée sur des optima locaux de CMA-ES est réduit si la taille de la population est suffisamment grande [131]. CMA-ES suppose la distribution de la population de solution comme une loi normale multivariée dont la matrice de covariance est un paramètre appris au cours de l'optimisation à l'aide d'une distribution de recherche normale[61]. L'algorithme 7 présente le principe de fonctionnement de l'algorithme. Une propriété importante de cet algorithme est son invariance par transformations linéaires de l'espace de recherche. CMA-ES est efficace pour trouver la solution de fonction unimodale et est supérieure à d'autres algorithmes état de l'art lorsque le problème est mal posé et non séparable. La figure 1.3 schématise itérativement le processus de convergence de CMA-ES. Itérativement l'algorithme génère des solutions candidates à partir de la moyenne des μ meilleures solutions courantes dans un voisinage σ et selon une loi normale définie dont les paramètres C sont donnés par la matrice de covariance. La sélection des meilleures solutions candidates permet de mettre à jour le vecteur moyen ainsi que la matrice de covariance et enfin d'adapter la taille du pas de la recherche. L'opération est répétée jusqu'à ce que la recherche converge ou qu'un critère d'arrêt ne soit pas atteint.

Covariance Matrix Adaptation Evolution Strategy with Increasing POPulation

Dans le contexte des problèmes d'optimisation multimodale, [60] montre que l'augmentation de la taille de la population peut améliorer les performances de CMA-ES. Auger et al. [10] propose une version de CMA-ES utilisant une stratégie de redémarrage : à chaque redémarrage (chaque fois que le critère

Input : f : fonction à optimiser σ_0 : step-size ϵ_σ : seuil du step-size y^* : valeur maximale de la fonction objectif ϵ_y : seuil de la valeur de la fitness \mathbf{x} : point de départ de la recherche ϵ_x : seuil de la valeur de la position de la recherche $\hat{\mathbf{X}}$: ensemble des optima déjà découverts**Output :** $\hat{\mathbf{X}}$: ensemble mise à jour des optima locaux y : valeur de l'optimum**begin** $y \leftarrow f(\mathbf{x})$ $\sigma \leftarrow \sigma_0$ **repeat**

mutation

 $\mathbf{x}' \leftarrow \mathcal{N}(\mathbf{x}, \sigma)$ $y' \leftarrow f(\mathbf{x}')$

selection selon le 1/5e adaptation

if $y' < y$ **then**| $\mathbf{x} \leftarrow \mathbf{x}'$ | $\sigma \leftarrow 2\sigma$ **end****else**| $\sigma \leftarrow 2^{-1/4}\sigma$ **end**

arreter la recherche si déjà connue

if $\exists \hat{\mathbf{x}} \in \hat{\mathbf{X}}, \|\mathbf{x} - \hat{\mathbf{x}}\| < \epsilon_x$ **then**| **break****end**

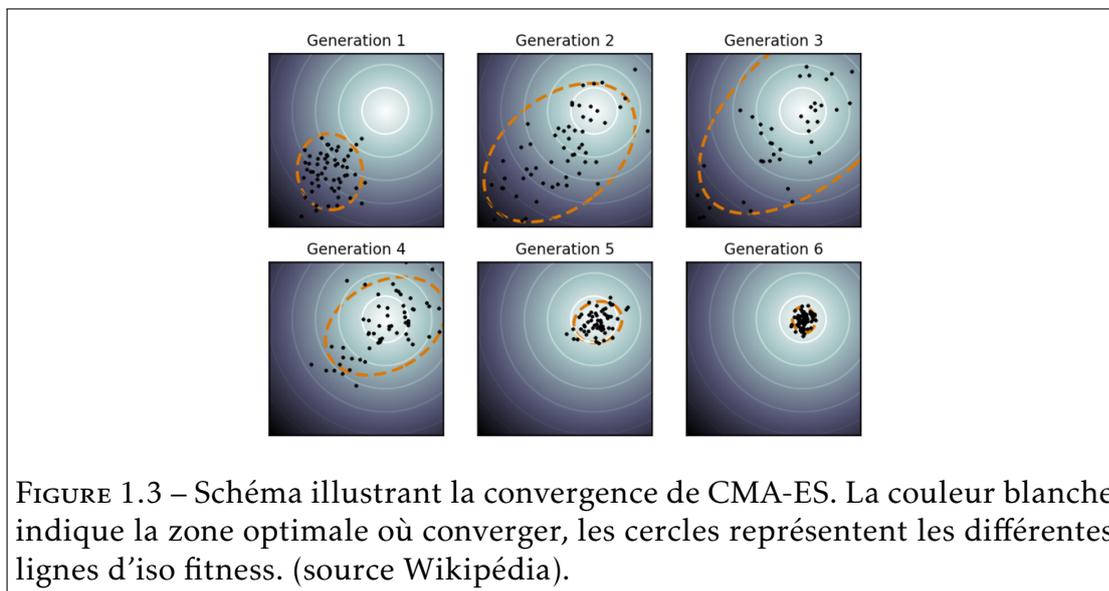
mémoriser l'optimum découvert

if $\|y - y^*\| < \epsilon_y$ **then**| $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} \cup \{\mathbf{x}\}$ | **break****end****until** $\sigma < \epsilon_\sigma$ **end**

Algorithme 5 : Recherche d'un optimum en utilisant un step size décroissant, SearchDS (minimisation)

```
Input :  $f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x$   
Output :  $\hat{\mathbf{X}}$   
begin  
   $\hat{\mathbf{X}} \leftarrow \emptyset$   
  while tous les optima n'ont pas été découverts do  
     $\mathbf{x} \leftarrow \text{SampleQR}()$   
     $\hat{\mathbf{X}}, y \leftarrow \text{SearchDS}(f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x, \hat{\mathbf{X}})$   
  end  
end
```

Algorithme 6 : Quasi-random Restarts with Decreasing Step-size



d'arrêt est rempli) la taille de la population est doublée comme expliqué par l'algorithme 8. En augmentant la taille de la population, la recherche locale devient alors de plus en plus globale après chaque redémarrage. Les résultats donnés dans [10] montrent que cette amélioration apporte une bonne performance dans le contexte d'une boîte noire multimodale. L'algorithme 8 présente le principe de fonctionnement de CMA-ES IPOP afin de gérer les problèmes multimodaux.

Input :

f : fonction à optimiser
 λ : nombre d'échantillons par itération
 x : vecteur de population
 s : vecteur des fitness

Output : x_1 : best optima find so far

begin

```

while stop when criterion met do
  for  $i$  in  $1 \dots \lambda$  do
     $x_i \leftarrow \mathcal{N}(\mu, C, \sigma)$ 
     $s_i \leftarrow f(i)$ 
  end
  Sort( $x_{1 \dots \lambda}$ )
  Update_mean_to_better_solution( $\mu, C, \sigma$ )
  Update_isotropic_evolution_path( $\mu, C, \sigma$ )
  Update_anisotropic_evolution_path( $\mu, C, \sigma$ )
  Update_covariance_matrix( $\mu, C, \sigma$ )
  Update_step-size( $\mu, C, \sigma$ )
end

```

end

Algorithme 7 : Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Compromis exploration/exploitation

À l'origine du développement et du choix de design de n'importe quel algorithme d'optimisation se trouve la question du compromis exploration/exploitation. En effet jusqu'ici la présentation du problème d'optimisation n'a mis en lumière que la partie exploitation. Cependant, comme expliqué dans la section 1.2.4, les algorithmes peuvent converger prématurément vers des optima

Input :

f : function to optimize
 λ : number of sample per iteration
 λ_0 : size of the initial population
 n : number of restart
 X : set of solution

Output : X or X_1 : set of optima or the best one

begin

```

   $n \leftarrow 0$  while number of evaluations not reach do
    |    $n \leftarrow n + 1$ 
    |    $\lambda \leftarrow \lambda_0 + 2 \times n$ 
    |    $X \leftarrow \text{cmaes}(\lambda, f)$ 
  end

```

end

Algorithme 8 : Covariance Matrix Adaptation Evolution Strategy with Increasing POPulation CMA-ES-IPOP

locaux. De ce fait avoir la possibilité d'explorer l'espace de recherche est intéressant afin d'éviter ce comportement. Le compromis apparaît donc comme suit : un algorithme privilégiant une exploration ne pourra converger tandis qu'un algorithme privilégiant l'exploitation risque de converger sur un optimum local. Les algorithmes doivent donc prendre en compte ce compromis afin de proposer une architecture capable de traiter avec ce compromis.

Synthèse Ce chapitre orienté sur les problèmes d'optimisation se divise en trois parties. La première partie, après une courte introduction, donne une définition d'un problème d'optimisation. La deuxième partie présente différentes typologies et caractéristiques intrinsèques des problèmes d'optimisation. Ce point est important à prendre en compte lors du choix de la méthode de résolution du problème. La dernière partie aborde la résolution des problèmes d'optimisation en se concentrant sur les problèmes d'optimisation numérique continue mono-objectif dans un contexte boîte noire. L'intérêt des méta heuristiques comme méthode de résolution y est mis en avant. S'ensuit une présentation détaillée issue de la littérature d'un échantillon d'algorithmes d'optimisation continue mono-objectif boîte noire multimodale. Ces derniers ont été choisis puisque la classe des problèmes d'optimisations traités dans la suite

du manuscrit sont continue, multimodale boîte noire. Les algorithmes RDS et QRDS utilisent une population à solution unique tandis que les algorithmes CMAES et une de ses variantes, CMAES-IPOP, maintiennent une population de plusieurs solutions. Ce chapitre pose les bases des travaux en optimisation qui seront utilisés dans les chapitres suivants.

Bibliographie du présent chapitre

- [4] Francisco Chicano et AL.. « Optimizing one million variable NK landscapes by hybridizing deterministic recombination and local search ». In : *GECCO* (2017), p. 753-760.
- [9] Anne AUGER. « Benchmarking the (1+ 1) evolution strategy with one-fifth success rule on the BBOB-2009 function testbed ». In : *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference : Late Breaking Papers*. 2009, p. 2447-2452.
- [10] Anne AUGER et Nikolaus HANSEN. « A restart CMA evolution strategy with increasing population size ». In : *2005 IEEE congress on evolutionary computation*. T. 2. IEEE. 2005, p. 1769-1776.
- [12] Thomas BACK, Frank HOFFMEISTER et Hans-Paul SCHWEFEL. « A survey of evolution strategies ». In : *Proceedings of the fourth international conference on genetic algorithms*. T. 2. 9. Morgan Kaufmann Publishers San Mateo, CA. 1991.
- [16] Léon BOTTOU. « Stochastic gradient descent tricks ». In : *Neural networks : Tricks of the trade*. Springer, 2012, p. 421-436.
- [21] Yan CAO et al. « Graph edge coloring : A survey ». In : *Graphs and Combinatorics* 35.1 (2019), p. 33-66.
- [27] Alessandro CRESPI et Auke Jan IJSPEERT. « Online optimization of swimming and crawling in an amphibious snake robot ». In : *IEEE Transactions on Robotics* 24.1 (2008), p. 75-87.
- [34] Stanley N DEMING, Lloyd R PARKER JR et M BONNER DENTON. « A review of simplex optimization in analytical chemistry ». In : *CRC Critical Reviews in Analytical Chemistry* 7.3 (1978), p. 187-202.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.
- [46] Siyuan FENG et al. « 3D walking based on online optimization ». In : *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2013, p. 21-27.
- [55] Genetha Anne GRAY et al. « Optimizing an empirical scoring function for transmembrane protein structure determination ». In : *INFORMS Journal on Computing* 16.4 (2004), p. 406-418.

- [58] Jeongwoo HAN, Michael KOKKOLARAS et Panos Y PAPALAMBROS. « Optimal design of hybrid fuel cell vehicles ». In : *Journal of fuel cell science and technology* 5.4 (2008).
- [59] Nikolaus HANSEN. « The CMA Evolution Strategy : A Comparing Review ». In : t. 192. Juin 2007, p. 75-102. DOI : 10.1007/3-540-32494-1_4.
- [60] Nikolaus HANSEN et Stefan KERN. « Evaluating the CMA evolution strategy on multimodal test functions ». In : *International Conference on Parallel Problem Solving from Nature*. Springer. 2004, p. 282-291.
- [61] Nikolaus HANSEN, Sibylle D MÜLLER et Petros KOUMOUTSAKOS. « Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES) ». In : *Evolutionary computation* 11.1 (2003), p. 1-18.
- [67] J. L. JEREZ et al. « Embedded Online Optimization for Model Predictive Control at Megahertz Rates ». In : *IEEE Transactions on Automatic Control* 59.12 (2014), p. 3238-3251.
- [68] Yufeng JIA et Shichun CHI. « Back-analysis of soil parameters of the Malutang II concrete face rockfill dam using parallel mutation particle swarm optimization ». In : *Computers and Geotechnics* 65 (2015), p. 87-96. ISSN : 0266-352X. DOI : <https://doi.org/10.1016/j.compgeo.2014.11.013>. URL : <http://www.sciencedirect.com/science/article/pii/S0266352X14002183>.
- [71] Ali KAMRAN et Liang GUOZHU. « Design and Optimization of 3D Radial Slot Grain Configuration ». In : *Chinese Journal of Aeronautics* 23.4 (2010), p. 409-414. ISSN : 1000-9361. DOI : [https://doi.org/10.1016/S1000-9361\(09\)60235-1](https://doi.org/10.1016/S1000-9361(09)60235-1). URL : <http://www.sciencedirect.com/science/article/pii/S1000936109602351>.
- [82] Claude LEMARÉCHAL. *Optimisation continue*. published : 2002-03-10. URL : <https://www.techniques-ingenieur.fr/base-documentaire/automatique-robotique-th16/automatique-sequentielle-42395210/optimisation-continue-s7210/>.
- [85] Florian LEPRÊTRE et al. « Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on SIALAC benchmark ». In : *Applied Soft Computing* 85 (2019), p. 105869.

- [88] T. Warren LIAO. « Two hybrid differential evolution algorithms for engineering design optimization ». In : *Applied Soft Computing* 10.4 (2010). Optimisation Methods Applications in Decision-Making Processes, p. 1188-1199. ISSN : 1568-4946. DOI : <https://doi.org/10.1016/j.asoc.2010.05.007>. URL : <http://www.sciencedirect.com/science/article/pii/S1568494610001080>.
- [91] Manuel LOZANO, Daniel MOLINA et Francisco HERRERA. « Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems ». In : *Soft Computing* 15.11 (2011), p. 2085-2087.
- [93] Alison L MARSDEN, Jeffrey A FEINSTEIN et Charles A TAYLOR. « A computational framework for derivative-free optimization of cardiovascular geometries ». In : *Computer methods in applied mechanics and engineering* 197.21-24 (2008), p. 1890-1905.
- [100] Mathieu MUNIGLIA et al. « Massive asynchronous master-worker EA for nuclear reactor optimization : a fitness landscape perspective ». In : *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2017, p. 295-296.
- [103] Rodrigue OEUVRAY. *Trust-region methods based on radial basis functions with application to biomedical imaging*. Rapp. tech. EPFL, 2005.
- [106] Christos H PAPADIMITRIOU et Kenneth STEIGLITZ. *Combinatorial optimization : algorithms and complexity*. Courier Corporation, 1998.
- [113] Ingo RECHENBERG. « Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution ». In : (1973).
- [115] Sebastian RUDER. « An overview of gradient descent optimization algorithms ». In : *arXiv preprint arXiv:1609.04747* (2016).
- [117] Reem SALIM et al. « The parameter identification of the Nexa 1 2 kW PEMFC’s model using particle swarm optimization ». In : *Renewable Energy* 82 (2015). International Conference on Renewable Energy : Generation and Applications (ICREGA 2014), p. 26-34. ISSN : 0960-1481. DOI : <https://doi.org/10.1016/j.renene.2014.10.012>. URL : <http://www.sciencedirect.com/science/article/pii/S0960148114006399>.
- [119] Marc SCHOENAUER, Fabien TEYTAUD et Olivier TEYTAUD. « A Rigorous Runtime Analysis for Quasi-Random Restarts and Decreasing Stepsize ». In : *Artificial Evolution*. Angers, France, 2011.
- [120] MA SCHUMER et Kenneth STEIGLITZ. « Adaptive step size random search ». In : *IEEE Transactions on Automatic Control* 13.3 (1968), p. 270-276.

- [127] Fabien TEYTAUD et Olivier TEYTAUD. « QR Mutations Improve Many Evolution Strategies : A Lot On Highly Multimodal Problems ». In : *Proc. of the 2016 GECCO conference*. ACM. 2016, p. 35-36.
- [131] Frans VAN DEN BERGH et Andries Petrus ENGELBRECHT. « Cooperative learning in neural networks using particle swarm optimizers ». In : *South African Computer Journal* 2000.26 (2000), p. 84-90.
- [134] Gerhard VENTER. « Review of optimization techniques ». In : *Encyclopedia of aerospace engineering* (2010).
- [135] Y. WANG et S. BOYD. « Fast Model Predictive Control Using Online Optimization ». In : *IEEE Transactions on Control Systems Technology* 18.2 (2010), p. 267-278.
- [142] Hai-xiang ZHAO et Frédéric MAGOULÈS. « Feature selection for support vector regression in the application of building energy prediction ». In : *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE. 2011, p. 219-223.

Calibration de modèles paramétriques

Sommaire du présent chapitre

2.1	Étalonnage automatique de modèle paramétrique	60
2.1.1	Contexte	60
2.1.2	Travaux similaires	61
2.1.3	Définition du problème d'optimisation	63
2.1.4	Protocole expérimental	64
2.1.5	Expériences et résultats	64
2.1.6	Synthèse	69

Introduction Ce chapitre présente une contribution scientifique. Le problème de calibration y est présenté comme un problème d'optimisation numérique boîte noire. La contribution majeure [38] propose une nouvelle méthodologie d'étalonnage automatique des paramètres de modèles biologiques grâce à l'utilisation de l'analyse inverse [20]. Les travaux portent en premier lieu sur une comparaison de performance entre les jeux de paramètres optimisés et celui des experts dans l'estimation de la teneur en eau dans le sol au cours d'une campagne. La performance d'un jeu de paramètres correspond à l'erreur de si-

mulation avec ce paramétrage par rapport à des valeurs étalons. S'ensuit une comparaison de deux algorithmes état de l'art (2019) pour ce type de problème utilisant des stratégies d'évolution différentes. Le critère de comparaison utilisé est la fitness associée à un jeu de paramètre découvert après qu'un budget d'évaluations soit écoulé. Enfin la fiabilité et le non-surapprentissage d'un jeu de paramètres optimisés du modèle WEEDRIQ par rapport au paramétrage expert sont aussi à l'étude via une validation croisée sur les différentes instances de test. L'expérience est effectuée avec le modèle de bilan hydrique WEEDRIQ [111]. Les performances de la méthodologie sont établies au travers de différentes instances dont les données sont issues de mesures réelles de campagne de cultures historiques.

2.1 Étalonnage automatique de modèle paramétrique

2.1.1 Contexte

Comme expliqué dans la section 1 de la partie introduction, en agriculture, de nouvelles techniques de prise de décision font appel à des capteurs hydriques pour mesurer l'état des plantes et du sol afin d'estimer la teneur en eau du sol. Couplé à un modèle de croissance de plante il est possible d'estimer l'état hydrique de la plante ainsi que la quantité d'eau disponible pour celle-ci, car dépendant de la taille (profondeur) des racines et des caractéristiques du sol. Bien qu'une telle approche numérique puisse permettre d'estimer avec précision l'état de la culture et que la précision des capteurs soit bonne, un inconvénient majeur est le réglage des (très) nombreux paramètres des modèles agronomiques. En effet, les modèles combinent souvent différents sous-modèles basés sur des équations différentielles, des transitions d'états finis, etc ... Ceci requiert le réglage des paramètres numériques, biologiques ou bien géologiques. Même si les experts peuvent mesurer, estimer ou donner des bornes à certaines valeurs de paramètres¹, la plupart du temps, leur valeur n'est pas connue pour une culture de plein champ, car dépendante d'un sol spécifique et d'une es-

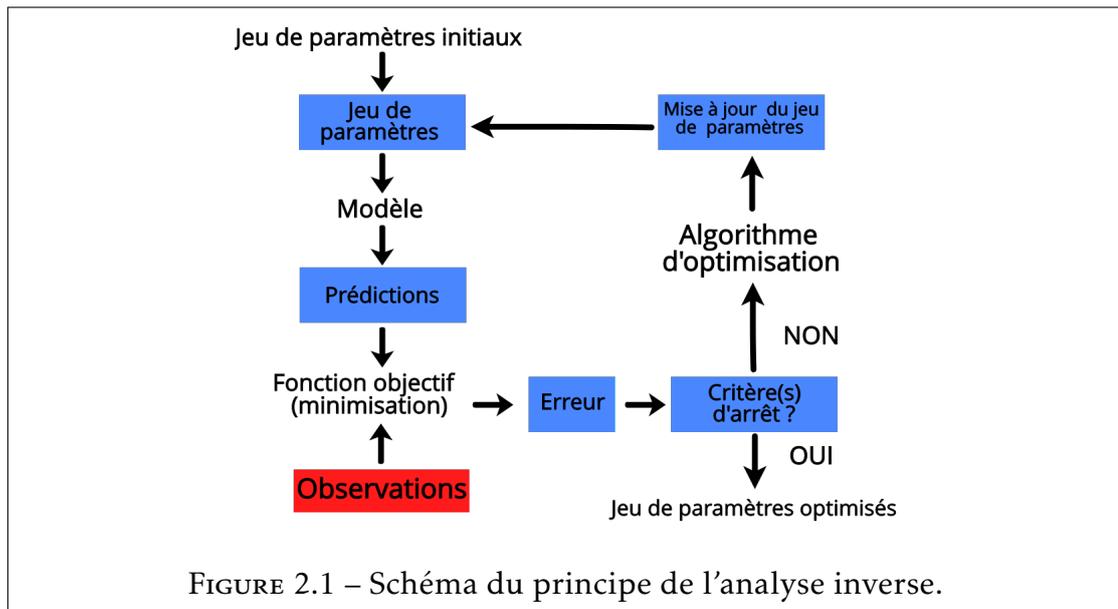
1. Certains paramètres numériques peuvent n'avoir aucune signification réelle ou biologique et donc n'ont en théorie aucune borne

pèce/variété de plantes. Cette contribution explique comment il est possible de définir plus précisément les paramètres du modèle de croissance des plants de pommes de terre WEEDRIQ [111] grâce à l'acquisition de données des capteurs hydriques (fournis par la société *WEENAT*) et d'un algorithme d'optimisation pertinent qui minimise l'erreur entre les valeurs prédites (calculées par le modèle), et celles fournies par les capteurs. WEEDRIQ étant une collection de modèles biologiques dont la définition analytique des équations est complexe, elle peut être vue comme une boîte noire. Dans le domaine évolutionnaire, ce problème boîte noire peut être vu comme un problème d'étalonnage ("calibration problem") [126, 56]. Le paramétrage du modèle de croissance du plant de pommes de terre montre des difficultés spécifiques. Ses paramètres dépendent des spécificités locales telles que le sol, la variété de pommes de terre, l'exposition aux intempéries, etc. ... De plus les données sont difficiles à recueillir et rares. En effet, une campagne d'acquisition de données avec des capteurs hydriques pour un champ de pommes de terre dure en moyenne 4 mois, et ne peut être effectuée sur le même champ que tous les 5 ans (et parfois 3 ans) en raison de la rotation des cultures. En outre, le nombre de paramètres des modèles est élevé : plusieurs dizaines pour des modèles représentatifs tels que STICS [137], AquaCrop [110] ou Weedric [111].

Au début de la thèse, ce problème avait été traité comme un problème modal. Cependant, chaque résultat d'exécution d'algorithme d'optimisation (en utilisant l'algorithme CMA-ES) ne donne pas, à un delta près, des solutions proches (en considérant la distance euclidienne) les unes des autres bien que les fitness associées soient proches. L'hypothèse forte de la multimodalité du problème a été posée.

2.1.2 Travaux similaires

Le problème lié à l'étalonnage automatique de paramètres de modèle est souvent traité en utilisant la méthode d'analyse inverse [20]. La figure 2.1 schématise le principe de l'analyse inverse. À partir d'un jeu de paramètres initiaux, itérativement, un jeu de paramètres est passé au modèle paramétrique. L'erreur entre les valeurs de sorties (prédiction) et des observations (valeurs étalons) est



mesurée par une métrique (fonction objective). Si un critère d'arrêt n'est pas atteint alors l'erreur est transmise à l'algorithme d'optimisation. Cette erreur correspond à la fitness associée à la solution courante (jeu de paramètres). L'algorithme d'optimisation détermine alors un nouveau jeu de paramètres dans l'espace de recherche. Ce processus est itéré jusqu'à ce que les valeurs de sortie correspondent à des données observables (valeurs étalons).

Concernant les problèmes d'optimisation multimodale ou Multi-Modal Optimization (MMO) en anglais, de nombreux algorithmes ont été proposés, certains utilisant la dérivabilité du gradient (voir section 1.2.4). Malheureusement, dans un contexte boîte noire, ces algorithmes ne sont pas directement applicables. Les autres méthodes, n'utilisant pas le gradient (gradient free), utilisent généralement les stratégies évolutionnaires (ES). Ces dernières ont montré leur robustesse et leur efficacité dans la résolution de problème académique d'optimisation continue mono-objectif boîte noire [13, 112, 113]. Les ES consistent à trouver itérativement de meilleures solutions à partir d'un point de départ. Dans le contexte des problèmes d'optimisation multimodaux, les stratégies évolutionnaires sont généralement combinées avec soit des techniques de "niching" soit des "restart strategy" afin de trouver tous les optima.

2.1.3 Définition du problème d'optimisation

Le problème d'optimisation de l'étalonnage du modèle d'irrigation WEEDRIQ 1 en utilisant les observations des données récoltées par les capteurs peut être défini comme suit. Le modèle possède des paramètres ($d = 36$ pour WEEDRIQ) réels pour lesquels aucune connaissance a priori sur leur implication dans les équations, simulations ou interactions dans les différents modèles n'est disponible. La borne de chaque paramètre est donnée par des experts (valeur de la FAO). Ainsi on peut le considérer comme une fonction boîte noire avec un espace de dimension $d > 1 : [0, 1]^d$, la valeur de chaque paramètre est normalisée pour éviter l'effet d'échelle. Le processus de normalisation est effectué grâce à l'équation 6.1 où x est un paramètre du modèle WEEDRIQ.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

Comme expliqué précédemment, l'hypothèse de la forte multimodalité du problème avec énormément d'optima locaux proches de l'optimum est posée. Il est donc nécessaire de trouver le maximum d'optima afin de découvrir le meilleur. Les capteurs étant en mesure de collecter les informations sur la teneur en eau présente dans le sol (par l'intermédiaire de la transformation du potentiel hydrique par l'équation de Van Genuchten [132]), leurs valeurs permettent évaluer l'erreur de la simulation du modèle. On peut donc définir l'erreur comme étant la somme au carré des distances entre les valeurs simulées et les valeurs capteurs, ceci étant résumé par l'équation 2.2, où x correspond à un vecteur de solution tel que $x = (x_1, \dots, x_{36})$, n le nombre de jours de simulation, SWC_t la valeur capteur à l'instant t transformée par l'équation de Van Genuchten en teneur en eau et $SWC_TNT_t(x)$ la teneur en eau estimée à l'instant t par le modèle WEEDRIQ en utilisant le paramétrage x .

$$f(x) = \sqrt{\sum_{t=1}^n \frac{1}{n} (SWC_t - SWC_TNT_t(x))^2} \quad (2.2)$$

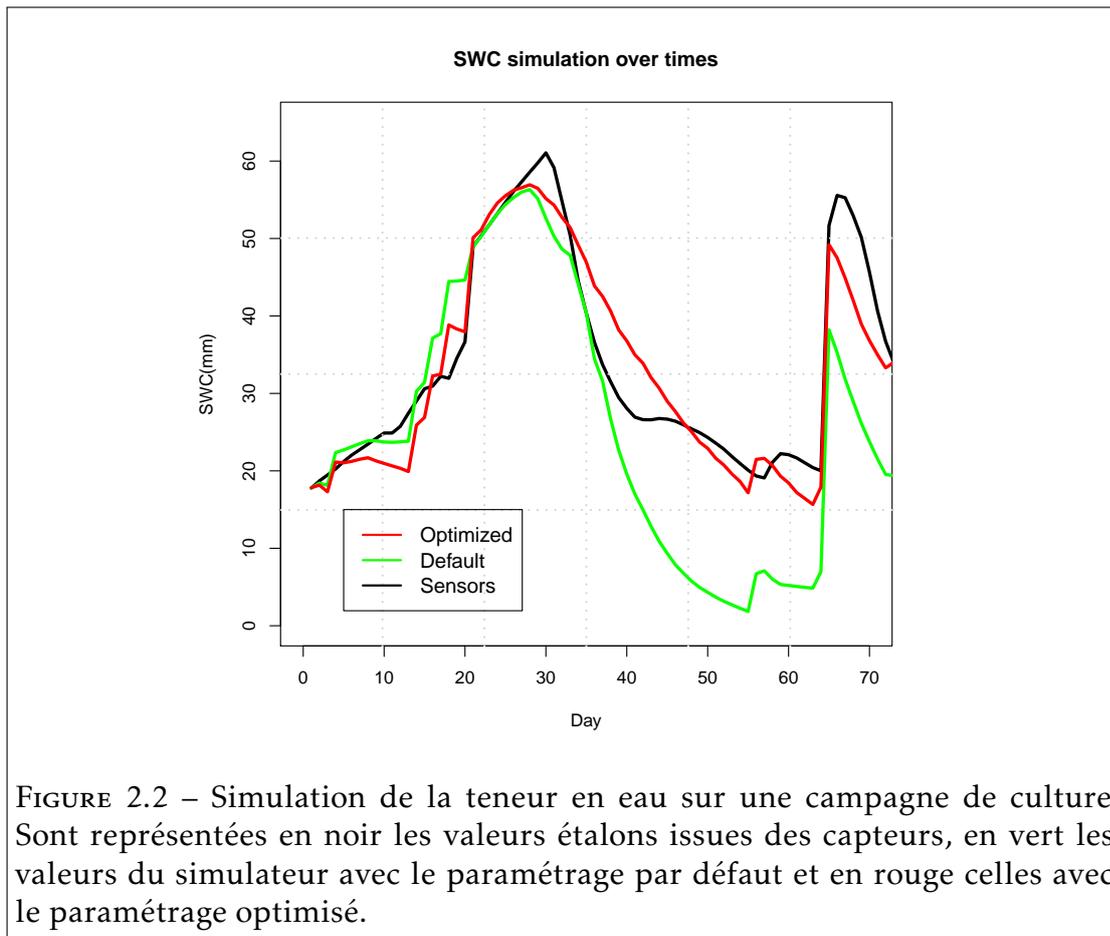
2.1.4 Protocole expérimental

Initialement, le modèle WEEDRIQ possède des valeurs de paramètres par défaut établis par des experts (valeurs de la FAO). L'étude porte en premier lieu sur la pertinence de l'étalonnage automatique de WEEDRIQ en utilisant la méthode de l'analyse inverse [20] grâce aux algorithmes d'optimisation par stratégie de dévolution multimodale (voir section 1.3.1) par rapport à l'utilisation du paramétrage expert. La méthode de l'analyse inverse postule qu'il est possible d'établir la performance d'un jeu de paramètres d'un modèle paramétrique comme étant une mesure d'écart entre la prédiction du modèle avec ce paramétrage et des valeurs étalons. On remarque qu'un problème de minimisation apparaît puisqu'il faut effectivement diminuer cette erreur pour obtenir un jeu de paramètres pertinent. L'optimisation se fera hors ligne, puisqu'il est nécessaire de connaître toutes les valeurs capteurs afin d'utiliser l'analyse inverse. D'autre part cette étude porte aussi sur une comparaison de performance entre 2 algorithmes ayant des stratégies différentes :

- QRDS, algorithme d'optimisation à solution unique combinée avec une stratégie de redémarrage. L'intérêt réside en sa convergence rapide et sa capacité d'exploration de l'espace de recherche grâce au séquençage du point de départ suivant une suite de nombres quasi aléatoires (suite de sobol).
- CMAES-IPOP, est une variante de CMAES pour une utilisation dans un contexte MMO. Sa matrice de covariance et l'augmentation de la population à chaque redémarrage doivent empêcher l'algorithme de rester bloqué dans des optima locaux.

2.1.5 Expériences et résultats

Les données qui ont pu être récoltées correspondent à 5 campagnes de cultures. Ces campagnes ont toutes eu lieu dans des zones géographiques proches et selon les experts avec un type de sol ne variant pas ou peu. Seule la variété de pommes de terre cultivée est différente. Comme l'optimisation doit pouvoir être exécutée en temps raisonnable en vue d'une mise en production, le budget alloué à chaque algorithme est de 10^5 évaluations (le temps d'exécution est de



l'ordre de la dixième de seconde). Afin de pouvoir comparer les performances sur la quantité peu élevée de campagnes de culture, 100 optimisations sont réalisées pour chaque algorithme sur chaque champ. Les paramètres de l'algorithme QRDS sont de 10^{-4} pour ϵ_{σ} , 0.5 pour ϵ_x et $5 * 10^{-3}$ pour ϵ_y . le paramétrage de CMAES-IPOP est celui par défaut.

Discussion la figure 2.2 présente un exemple de campagne de culture parmi les 5 disponibles. La valeur réelle (capteur) est représentée par la courbe noire, celle du simulateur avec le paramétrage par défaut (expert) en vert et celle avec le paramétrage suite à une optimisation en rouge. Visuellement, les résultats indiquent clairement une amélioration de la simulation suite à l'optimisation des paramètres. Numériquement, les valeurs pour chaque champ sont explicitées

TABLEAU 2.1 – Performances des jeux de paramètres pour la prédiction de l'évolution de la teneur en eau dans le sol en fonction de la méthode utilisée et de la parcelle de test. Les valeurs médianes en gras sont significativement meilleures selon le test Mann-Whitney à un niveau de confiance de 0,01. QRDS trouve toujours le meilleur jeu de paramètres, de plus la moyenne et la médiane des fitness sont meilleures (plus petite).

Crop	Algorithm	Best	Mean	Median
Crop 1	Default	80.83	/	/
	QRDS	15.8	16.3 ± 0.03	16.3
	CMA-ES IPOPOP	16.4	18.5 ± 0.12	18.3
Crop 2	Default	57.81	/	/
	QRDS	16	16.7 ± 0.03	16.7
	CMA-ES IPOPOP	17	19 ± 0.08	19
Crop 3	Default	63.65	/	/
	QRDS	17.5	18 ± 0.03	17.9
	CMA-ES IPOPOP	19	21.5 ± 0.12	21.6
Crop 4	Default	47.85	/	/
	QRDS	14.2	15.8 ± 0.06	16
	CMA-ES IPOPOP	16.7	20 ± 0.25	19.4
Crop 5	Default	50.74	/	/
	QRDS	15.1	15.3 ± 0.02	15.3
	CMA-ES IPOPOP	15.3	16.6 ± 0.07	16.5

ci-après.

Le tableau 2.1 présente les Mean Square Error (MSE) avec leurs intervalles de confiance à 95% de chaque algorithme ainsi que la performance du paramétrage par défaut. Une erreur faible indique une bonne prédiction. Quel que soit l'algorithme utilisé, la performance est toujours meilleure que le paramétrage par défaut. Ceci montre la pertinence de la méthode comme moyen de résolution de ce problème. Concernant la comparaison des algorithmes, la figure 2.3 présente la dispersion des optima trouvés après 100 optimisations pour chaque champ et pour chaque algorithme. Autrement dit chaque algorithme a opéré 100 optimisations avec un budget d 10^6 évaluations à chaque optimisation pour chaque champ. L'optimum à chaque optimisation est ensuite conservé.

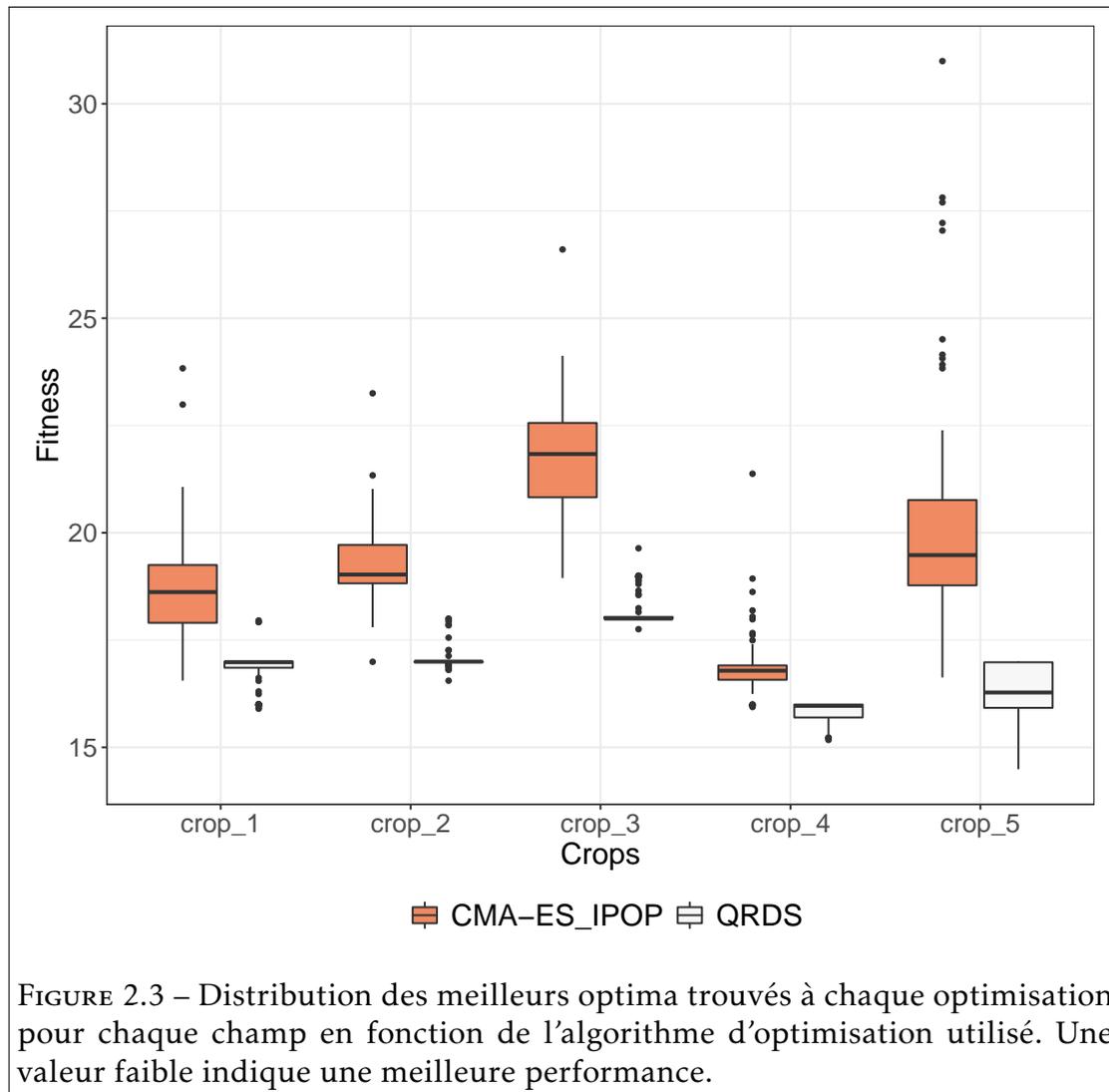


FIGURE 2.3 – Distribution des meilleurs optima trouvés à chaque optimisation pour chaque champ en fonction de l’algorithme d’optimisation utilisé. Une valeur faible indique une meilleure performance.

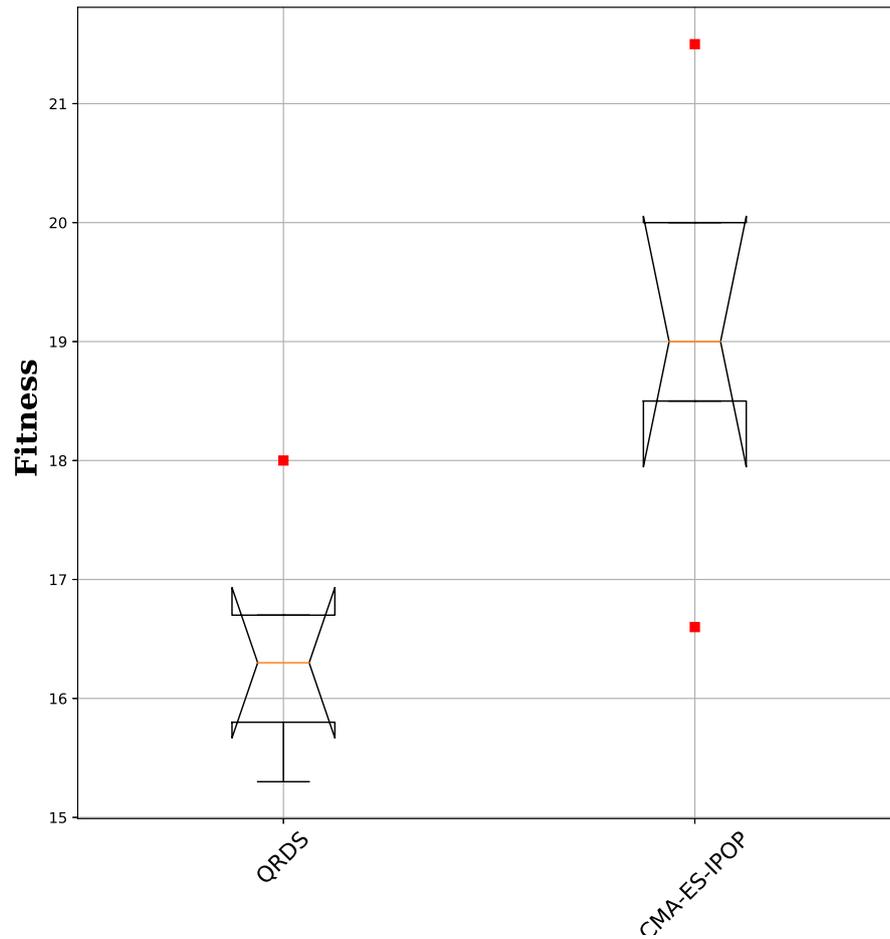


FIGURE 2.4 – Distribution des meilleurs optima trouvés après 100 optimisations pour chaque champ en fonction de l'algorithme utilisé. Les points rouges représentent une valeur en dehors de l'intervalle de confiance. Une valeur faible indique une meilleure performance.

Ceci représente donc finalement 500 valeurs par algorithme. D'après les résultats, QRDS est toujours statistiquement meilleur que CMAES-IPOP au vu des performances. De plus, QRDS est plus fiable, car la dispersion de ses optima est plus faible, au contraire de CMAES-IPOP ayant une dispersion plus importante. Si l'on considère à présent la meilleure solution trouvée pour chaque algorithme après les 100 optimisations pour chaque champ (comme le montre la figure 2.4), QRDS reste statistiquement plus performante que CMAES-IPOP. La raison de cette différence peut être expliquée par la topologie du problème. En effet, le problème étant très fortement multimodal, une stratégie favorisant une exploration semble plus efficace et QRDS grâce à son "murder operator" ne concentre son budget que sur des zones où un optimum n'a pas encore été découvert. Une autre hypothèse peut être la forme du paysage qui ne doit pas être globalement convexe, ce qui n'aide pas la matrice de covariance de CMA à converger.

2.1.6 Synthèse

À partir de la reformulation d'un problème de paramétrage d'un modèle biologique en un problème d'optimisation multimodale boîte noire, résultats des expériences démontrent que l'utilisation de l'analyse inverse comme méthodologie afin de paramétrer automatiquement le modèle par rapport à l'utilisation du paramétrage expert était pertinente puisque les jeux de paramètres issus de notre méthodologie sont toujours meilleurs pour prédire l'évolution de la teneur en eau dans le sol. Les tests ont été effectués sur 5 parcelles de cultures différentes sur plusieurs années. Le gain moyen est de 23% grâce à l'utilisation de cette méthode par rapport au paramétrage de référence. Concernant la comparaison de l'algorithme QRDS et de l'algorithme CMAES-IPOP, 100 exécutions de chaque algorithme ont été opérées sur chaque champ de test. Il en résulte que l'algorithme QRDS est statistiquement meilleur comme méthode de résolution. On peut émettre l'hypothèse qu'il est nécessaire d'explorer plus fréquemment l'espace de recherche et/ou que la fonction objective doit être localement simple (fonction lisse, localement convexe, ...) ce qui avantage l'algorithme QRDS par rapport à CMA-ES qui doit utiliser une partie de ses évaluations pour l'apprentissage de sa matrice de covariance. Enfin la question de la fiabilité et de la

robustesse des jeux de paramètres optimisés dans le contexte de la culture de pommes de terre a été étudiée. Expérimentalement, les performances de validation d'un jeu de paramètres optimisé sur les autres parcelles de tests (sans nouveau processus d'optimisation) apportent un gain statistiquement significatif par rapport au paramétrage de référence. Ceci permet de valider l'approche et la méthodologie employée.

Bibliographie du présent chapitre

- [13] Hans-Georg BEYER. *The Theory of Evolution Strategies*. Springer Science & Business Media, 2001.
- [20] Michele CALVELLO et Richard J FINNO. « Selecting parameters to optimize in model calibration by inverse analysis ». In : *Computers and Geotechnics* 31.5 (2004), p. 410-424. ISSN : 0266-352X. DOI : <https://doi.org/10.1016/j.compgeo.2004.03.004>. URL : <http://www.sciencedirect.com/science/article/pii/S0266352X04000540>.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.
- [56] Hoshin V GUPTA, Keith J BEVEN et Thorsten WAGENER. « Model calibration and uncertainty estimation ». In : *Encyclopedia of hydrological sciences* (2006).
- [110] Dirk RAES et al. « AquaCrop—the FAO crop model to simulate yield response to water : II. Main algorithms and software description ». In : *Agronomy Journal* 101.3 (2009), p. 438-447.
- [111] Eric RAMAT et Bertrand VANDOORNE. *Plant growth model for decision making support*. Rapp. tech. Université du Littoral Côte d’Opale, et ISA Lille, 2002.
- [112] J. RAPIN et O. TEYTAUD. *Nevergrad - A gradient-free optimization platform*. <https://GitHub.com/FacebookResearch/Nevergrad>. 2018.
- [113] Ingo RECHENBERG. « Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution ». In : (1973).
- [126] Y TANG, P REED et Thibaut WAGENER. « How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? » In : (2006).
- [132] M Th VAN GENUCHTEN. « A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1 ». In : *Soil science society of America journal* 44.5 (1980), p. 892-898.
- [137] FD WHISLER et al. « Crop simulation models in agronomic systems ». In : *Adv. Agron* 40.1 (1986), p. 41-208.

Deuxième partie

**Problèmes d'optimisation assistée
par renforcement et contributions
dans la résolution de problèmes
multimodaux**

Problèmes d'apprentissage par renforcement

Sommaire du présent chapitre

3.1 Apprentissage par renforcement	76
3.2 Problème du bandit manchot	77
3.2.1 Upper Confidence Bound	77
3.2.2 Upper Confidence Tree	78

Introduction Ce chapitre introduit les problèmes d'apprentissage par renforcement. Dans un premier temps, le problème d'apprentissage par renforcement est formellement défini. S'ensuit une présentation du problème du bandit manchot avec l'explication du compromis exploration/exploitation. La présentation de ces éléments est nécessaire pour appréhender les travaux dans la suite du manuscrit puisque les travaux qui en découlent utilisent le problème du bandit manchot comme une stratégie de restart. La méthode de l'Upper Confidence Bound est ensuite introduite comme méthode pour gérer le compromis exploration/exploitation. Enfin l'algorithme de l'Upper Confidence Tree est détaillé.

3.1 Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de recherche de l'apprentissage automatique. Cet apprentissage considère comment un agent doit prendre des décisions dans un environnement. L'idée principale est de récompenser chaque action de l'agent. Ce processus peut être modélisé comme un processus de décision markovien $M = (\mathcal{S}, \mathcal{A}, P, R)$ où :

- \mathcal{S} est l'ensemble des états
- \mathcal{A} l'ensemble des actions
- $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$ est la probabilité d'une transition d'un état s vers s' grâce à l'action a (à un pas de temps discret t)
- $R_a(s, s')$ la récompense après la transition de l'état s vers l'état s' avec l'action a

La prise de décision de l'agent est modélisée par une politique $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. Cette politique donne une probabilité de faire une action a quand l'agent est dans un état s , c'est-à-dire $\pi(s, a) = P(a_t = a | s_t = s)$. Le résultat R est la somme cumulée des récompenses r_k jusqu'à l'épisode actuel depuis le temps t jusqu'à un état final atteint au moment T [52].

$$R_t = \sum_{k=t+1}^T r_k \quad (3.1)$$

Les méthodes d'apprentissage par renforcement basées sur la récompense essaient de trouver la meilleure politique, c'est-à-dire celle qui maximise l'espérance de la récompense en calculant une fonction de valeur d'action intermédiaire.

$$Q^\pi(s, a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a] \quad (3.2)$$

En d'autres termes, cette fonction estime le résultat quand l'agent fait l'action a dans l'état s et utilise la politique π pour effectuer toutes les actions consécutives. La valeur de l'action est généralement estimée en utilisant un algorithme itératif : une estimation de la fonction peut être utilisée pour améliorer la politique et ainsi calculer une nouvelle estimation de la fonction. À chaque

étape de ce processus, l'algorithme doit choisir une action à considérer, c'est-à-dire le compromis entre l'exploitation (sélectionner l'action ayant la valeur maximale) et l'exploration (sélectionner une action aléatoire). Ce compromis est un problème classique des problèmes de bandit manchot [7].

3.2 Problème du bandit manchot

Le problème du bandit manchot ou en anglais Multi-Armed Bandit (MAB) [77] est un problème classique d'apprentissage où l'agent, devant un environnement inconnu (le bandit manchot) doit successivement sélectionner un bras du manchot (une action), avec une récompense inconnue, avec pour objectif de maximiser les gains cumulés totaux. À titre d'exemple, on peut citer la conduite autonome, la robotique, ... Plus précisément, N bras du bandit avec des probabilités sur les gains inconnus p_n sont disponibles. À chaque instant k , l'agent sélectionne un bras et reçoit un gain $r_k = 1$ avec la probabilité p_n ou $r_k = 0$ sinon. Le but est de maximiser le gain total accumulé sur les K instants. En d'autres termes, l'agent doit trouver une politique qui minimise la perte par rapport au meilleur bras. Ce regret peut être exprimé comme :

$$Kp^* - \sum_{k=1}^K r_k \quad (3.3)$$

où $p^* = \max\{p_1, \dots, p_N\}$ est la probabilité maximale de gain parmi les bras du bandit. Une technique efficace pour traiter le compromis exploitation/explo-ration dans les problèmes de bandit manchot est l'*Upper Confidence Bound* ou UCB.

3.2.1 Upper Confidence Bound

Proposé par Auer [8], c'est une méthode pour les problèmes faisant intervenir un compromis entre exploration et exploitation puisqu'il fournit une borne asymptotique optimale sur le regret en $\mathcal{O}(\ln(K))$. Le principe de cet algorithme est le suivant : Étant données k actions possibles, à chaque itération l'algorithme sélectionne le bras i qui possède le score UCB le plus élevé :

$$ucb_i = \hat{r}_i + R \cdot \sqrt{\frac{2 \ln S}{S_i}} \quad (3.4)$$

où \hat{r}_i est la récompense moyenne du bras i , S_i le nombre de fois que le bras i a été sélectionné durant les S itérations. On remarque que la partie gauche de l'équation tend à favoriser le bras ayant le gain moyen le plus élevé (exploitation) tandis que la partie droite favorise le bras ayant été le moins sélectionné (exploration). Le compromis est réglé par la constante R . Plus cette valeur est grande plus l'exploration aura de l'importance dans le choix du bras. Le paramétrage de cette constante est très souvent empirique bien que des méthodes de paramétrage automatique aient été proposées [7].

3.2.2 Upper Confidence Tree

L'algorithme *Upper Confidence Tree* (UCT) est un algorithme d'apprentissage par renforcement basé sur les récompenses [74, 122]. Cet algorithme est l'implémentation du *Monte-Carlo Tree Search* (MCTS). L'algorithme estime une fonction de valeur d'action $Q_{UCT}(s, a)$ en construisant un arbre de recherche où la racine (l'élément initial) correspond à l'état de départ et les branches aux états visités par la suite. Dit autrement, l'arbre $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{A}$ est un sous-ensemble de paires (états/actions). La valeur de chaque nud est estimée par simulation "Monte Carlo". La politique utilisée dans l'algorithme UCT est basée sur l'algorithme UCB [7], c'est-à-dire un algorithme de bandit manchot conçu pour résoudre les problèmes avec compromis exploration/exploitation. Pour un état courant s , si toutes les actions sont présentes dans l'arbre ($\forall a \in \mathcal{A}(s), (s, a) \in \mathcal{T}$) alors la politique arborescente est appliquée. C'est-à-dire qu'une phase de sélection successive des différents nuds de l'arbre est effectuée en considérant à chaque fois l'action maximisant le score UCB sur $Q_{UCT}(a, s)$:

$$\pi_{UCT}(s) = \operatorname{argmax}_a Q_{UCT}(s, a) + k_{UCT} \sqrt{\frac{\log N(s)}{N(s, a)}} \quad (3.5)$$

où $N(s, a)$ est le nombre de fois que l'action a a été sélectionnée dans l'état s , $N(s)$ est le nombre total de sélections de s ($N(s) = \sum_{a \in \mathcal{A}(s)} N(s, a)$). k_{UCT} est un para-

mètre de l'algorithme. Si toutes les actions de l'état courant ne sont pas dans l'arbre, c'est-à-dire que $\bar{\mathcal{A}}(s) = \{a | (s, a) \notin \mathcal{T}\}$ n'est pas vide alors une action est sélectionnée aléatoirement dans $\bar{\mathcal{A}}(s)$ et un nouveau nud est ajouté à l'arbre. Enfin, à la fin de l'épisode, chaque nud visité est mis à jour en utilisant le résultat de l'épisode :

$$\begin{aligned} N(s_t, a_t) &\leftarrow N(s_t, a_t) + 1 \\ Q_{UCT}(s_t, a_t) &\leftarrow Q_{UCT}(s_t, a_t) + \frac{R_t - Q_{UCT}(s_t, a_t)}{N(s_t, a_t)} \end{aligned} \quad (3.6)$$

Les nouveaux nuds sont simplement initialisés avec la récompense R_t . Ainsi, après plusieurs épisodes, on obtient un arbre non équilibré où les nuds les plus prometteurs ont été fréquemment visités (donc plus estimés).

Synthèse Ce chapitre introduit les problèmes d'apprentissage par renforcement. Dans un premier temps, l'apprentissage est défini comme un processus markovien. S'ensuit une présentation du problème du bandit manchot ainsi que du compromis d'exploration/exploitation inhérent aux problèmes d'apprentissage par renforcement. La méthode UCB est ensuite expliquée comme une méthode permettant de maintenir le compromis exploration/exploitation des différentes actions possibles durant le processus d'apprentissage. Enfin, l'algorithme UCT est présenté. Ce dernier implémente un Monte-Carlo Tree Search dont la politique de choix d'une action est motivée par son score UCB permettant de maintenir le compromis exploration/exploitation. Ces deux méthodes sont à la base des travaux qui ont conduit à la conception des nouveaux algorithmes utilisés dans le chapitre suivant.

Bibliographie du présent chapitre

- [7] Peter AUER, Nicolo CESA-BIANCHI et Paul FISCHER. « Finite-time analysis of the multiarmed bandit problem ». In : *Machine learning* 47.2-3 (2002), p. 235-256.
- [8] Peter AUER, Nicolò CESA-BIANCHI et Paul FISCHER. « Finite-time Analysis of the Multiarmed Bandit Problem ». In : *Mach. Learn.* 47.2-3 (2002), p. 235-256. ISSN : 0885-6125. DOI : 10.1023/A:1013689704352.

- [52] Sylvain GELLY et David SILVER. « Combining Online and Offline Knowledge in UCT ». In : *International Conference of Machine Learning*. 2007.
- [74] L. KOCSIS et C. SZEPESVÁRI. « Bandit based Monte-Carlo planning ». In : *Machine Learning : ECML 2006*. Springer, 2006, p. 282-293.
- [77] Tze Leung LAI et Herbert ROBBINS. « Asymptotically efficient adaptive allocation rules ». In : *Advances in applied mathematics* 6.1 (1985), p. 4-22.
- [122] David SILVER. « Reinforcement Learning and Simulation-Based Search ». Thèse de doct. University of Alberta, 2009.

Problèmes d'optimisation assistée par renforcement

Sommaire du présent chapitre

4.1 Stratégie de restart assistée par UCB	82
4.1.1 Motivations	82
4.1.2 Travaux similaires	83
4.1.3 UCB Random-restarts with Decreasing Step-size (URDS)	84
4.1.4 Fonctions de test	86
4.1.5 Analyse expérimentale	89
4.1.6 Synthèse	91
4.2 Stratégie de restart assistée par recherche arborescente	94
4.2.1 Motivations	94
4.2.2 Upper Confidence Tree for Random restarts with De- creasing Step-size	95
4.2.3 Analyse expérimentale	97
4.2.4 Fonctions de test	97
4.2.5 Résultats	100
4.2.6 Synthèse	104

Introduction Ce chapitre présente deux nouvelles contributions en recherche. Deux nouveaux algorithmes d'optimisation numérique multimodale, dans un contexte boîte noire utilisant une stratégie d'évolution avec une stratégie de *restart* assistés par un algorithme d'apprentissage par renforcement, sont expliqués. Ces algorithmes proposent de considérer le choix des positions des recherches locales comme un problème de bandit manchot. Ils suggèrent de découper l'espace de recherche en sous-parties et de considérer chaque sous-ensemble comme le bras du bandit. Le choix d'une zone pour la recherche locale est alors motivé par un score donné par la méthode de l'Upper Confidence Bound. Le premier algorithme, Ucb Random with Decreasing Step-size (URDS) [36], considère un découpage fixe de l'espace de recherche tandis que le second, Upper confidence Tree Random Decreasing Step-size (UCT-RDS), [37] considère un découpage hiérarchique de l'espace de recherche. Les performances des algorithmes sont comparées avec celles de deux autres algorithmes, QRDS et RDS sur plusieurs problèmes d'optimisation multimodale issus de la littérature.

4.1 Stratégie de restart assistée par UCB

4.1.1 Motivations

L'optimisation multimodale 1.2.6 est un outil très important pour de nombreux problèmes en ingénierie ou en apprentissage (section 1.2.6). Les résultats des travaux précédents (section 2.1) tendent à prouver l'hypothèse de la multimodalité du problème d'optimisation des paramètres du modèle WEEDRIQ. Ainsi, proposer un nouvel algorithme utilisant l'algorithme Quasi Random with Decreasing Step-size (QRDS) combiné avec un autre algorithme capable de choisir plus judicieusement la position des recherches locales permettrait d'améliorer les performances de ce dernier. Pour étayer ces propos, il faut comparer les performances avec des fonctions multimodales de tests de la littérature. Cette contribution a donc pour intérêt de proposer un nouvel algorithme d'optimisation multimodale capable de trouver tous les optima globaux (dans un contexte de maximisation) d'une fonction donnée dont la définition analytique est in-

connue. Plus précisément, étant donnée une fonction boîte noire continue multimodale $f : [0, 1]^D \rightarrow \mathbb{R}$, le but étant de trouver tous les points $x \in [0, 1]^D$ tels que $f(x) = y^*$ où y^* est la valeur (aussi appelée fitness) maximale de f . Ce genre d'optimisation est associé à l'optimisation mono modale classique lorsque les optima locaux sont très proches (mais de moins bonne qualité) de l'optimum global. Dans ce cas, il est nécessaire de trouver tous les optima lorsque cela est possible¹ ou bien le maximum d'entre eux afin de déterminer l'optimum parmi ces solutions. Pour analyser les performances d'un algorithme d'optimisation multimodale, l'une des métriques couramment prises en considération est le nombre d'évaluations de f [119, 84, 60]. Un bon algorithme trouve tous les optima globaux avec le moins d'évaluations possible.

4.1.2 Travaux similaires

De très nombreux algorithmes ont été proposés pour répondre aux problèmes d'optimisation, mais beaucoup d'entre eux utilisent la dérivée de la fonction ("gradient-based methods"), ce qui n'est pas directement applicable dans un contexte d'optimisation boîte noire. Ainsi, les méthodes n'utilisant pas la dérivée du gradient (gradient-free methods) sont généralement basées sur des stratégies évolutionnaires (ES) [113, 14] (voir section 1.3.1). Dans le contexte des optimisations multimodales, cette stratégie est combinée avec soit une technique de "niching" soit une technique de "restart" pour trouver tous les optima. Parmi les différentes techniques de "niching", la technique de partage ou "sharing" [53] indique que tous les individus de la même niche (c'est-à-dire un groupe d'individus proches les uns des autres) partagent la même fitness. Ceci force l'algorithme à explorer de nouvelles zones de l'espace de recherche ne partageant pas la même niche. La technique de "crowding" [31] (incluant la version déterministe [92] et probabiliste [96]) consiste à remplacer des individus de la population actuelle (crowd) par des individus candidats similaires d'une nouvelle population. La technique de "clustering" [139] analyse et regroupe les individus similaires. Enfin, on peut citer aussi la technique de "clearing" [107] et plus récemment le "modified clearing" [123]. Ce sont des variantes de la tech-

1. Le nombre de solutions peut être exponentiel avec la dimension

nique de "sharing" où les meilleurs individus sont préservés (individus dominants) tandis que la fitness des autres est supprimée (individus dominés). Une autre technique classique des optimisations multimodales est la stratégie de relance ou "restart" [11, 70, 127, 3]. L'algorithme QRDS proposé par [119] (voir section 1.3.1) a de bonnes performances quand les optima sont relativement bien distribués à travers l'espace de recherche ou quand la fonction est localement simple, c'est-à-dire lorsque la fonction est, par exemple, lisse, non bruitée, localement convexe, ...

4.1.3 UCB Random-restarts with Decreasing Step-size (URDS)

Un nouvel algorithme plus flexible que les algorithmes utilisant une stratégie de restart détaillés ci-dessus est proposé où la stratégie de *restart* apprend quelle(s) zone(s) de l'espace de recherche est(sont) prometteur(euses). Comme expliqué ci-dessus, les algorithmes comme QRDS sélectionnent un nouveau point de départ dans l'espace de recherche en utilisant une méthode d'échantillonnage (quasi) aléatoire. Cette approche convient bien pour les fonctions localement simples et les fonctions avec des optima distribués uniformément sur l'espace de recherche. Cependant, dans de nombreuses applications du monde réel, la fonction à optimiser est plus complexe, car pouvant avoir des zones localement complexes, des plateaux, ... C'est pourquoi certaines zones de l'espace de recherche sont plus à considérer que d'autres. Ce nouvel algorithme utilise la méthode de l'Upper Confidence Bound (UCB) pour sélectionner plus fréquemment les zones les plus prometteuses dans la recherche des optima. L'espace de recherche est partitionné en utilisant une grille régulière avec M subdivisions selon chaque dimension. Il en résulte $N = M^D$ zones. Ces zones sont ensuite considérées comme les bras d'un problème de bandit manchot (section 3.2). Le choix d'une zone peut être vu comme un dilemme entre une politique d'exploration équitable de toutes les zones à travers l'espace de recherche ou bien une exploitation des zones possédant une espérance élevée. La méthode UCB est une solution efficace pour gérer ce compromis. L'objectif est d'apprendre, au cours du processus d'optimisation, à sélectionner efficacement la zone du prochain point de départ. Cette approche a prouvé son efficacité pour les problèmes

d'intégration numérique [84]. Pour les optimisations multimodales, son implémentation est plus compliquée puisque URDS combine la stratégie de relance (grâce à l'UCB) pour gérer les multiples optima et une recherche locale pour estimer la récompense. L'algorithme proposé s'appelle UCB Random-restarts with Decreasing Step-size (URDS) et son principe de fonctionnement est expliqué par l'algorithme 9. L'algorithme URDS est basiquement composé de 2

Ucb Random-restarts with Decreasing Step-size

Input : $f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x$

Output : $\hat{\mathbf{X}}$

begin

$\hat{\mathbf{X}} \leftarrow \emptyset$

$k \leftarrow 0$

 { initialization step }

foreach j **do**

$k \leftarrow k + 1$

$\mathbf{x} \leftarrow \text{SampleArea}(j)$

$\hat{\mathbf{X}}, y \leftarrow \text{SearchDS}(f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x, \hat{\mathbf{X}})$

$S[j] \leftarrow \frac{1}{y^* - y + 0.1}$

$A[j] \leftarrow 1$

end

 { UCB step }

while *all optima not found* **do**

$k \leftarrow k + 1$

$j^* \leftarrow \text{argmax}_j \frac{S[j]}{A[j]} + R \sqrt{\frac{\ln k}{A[j]}}$

$\mathbf{x} \leftarrow \text{SampleArea}(j^*)$

$\hat{\mathbf{X}}', y \leftarrow \text{SearchDS}(f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x, \hat{\mathbf{X}})$

if $\hat{\mathbf{X}}' \neq \hat{\mathbf{X}}$ **then**

$\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}}'$

$S[j^*] \leftarrow S[j^*] + \frac{1}{y^* - y + 0.1}$

end

$A[j^*] \leftarrow A[j^*] + 1$

end

end

Algorithme 9 : Ucb Random-restarts with Decreasing Step-size

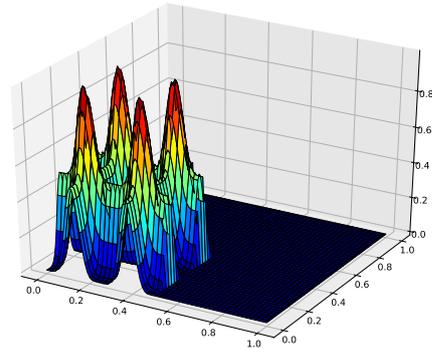
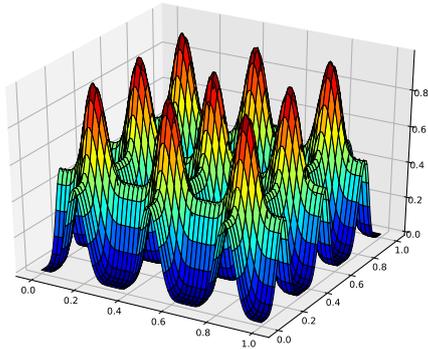
étapes : l'étape d'initialisation et l'étape de sélection UCB. La première étape ef-

fectue une recherche locale des N zones pour calculer une première estimation de la récompense de chaque zone. Les récompenses initiales sont stockées dans S et le nombre de sélections dans A . L'optimum trouvé durant cette étape est aussi mémorisé dans \hat{X} . Puis la seconde étape est la stratégie de restart. La zone la plus intéressante est sélectionnée par la stratégie UCB (section 3.2.1) et le point de départ de la recherche est échantillonné selon une séquence quasi aléatoire mise à l'échelle pour correspondre aux bornes du sous-espace considéré. La recherche locale est effectuée depuis ce point et à la fin une mise à jour de S et A est réalisée en fonction de la récompense. L'étape UCB se termine lorsque tous les optima ont été trouvés (\hat{X}). Si le nombre d'optima n'est pas connu à l'avance, un autre critère d'arrêt peut être employé. La récompense, terme favorisant l'exploitation dans la formule UCB (section 3.2.1), est définie comme : $\frac{1}{y^* - y + 0.1}$. L'idée sous-jacente de cette formule est de donner une récompense plus élevée au point proche du réel optimum y^* de la fonction. L'algorithme peut être modifié très facilement pour gérer les cas où y^* est inconnu. Par exemple, on peut définir la récompense de la fitness comme le nombre trouvé d'optima, c'est-à-dire remplacer dans l'algorithme 9 ligne 9 par $S[j] \leftarrow y$ et ligne 19 par $S[j^*] \leftarrow S[j^*] + y$. L'algorithme URDS requiert le paramétrage de 2 variables. Le paramètre de R est très classique dans les méthodes UCB; il règle le compromis exploration/exploitation 3.2.1. Une étude empirique permet de régler ce paramètre. Le paramètre M est le nombre de subdivisions par dimension et donne donc le nombre de zones à considérer par l'algorithme ($N = M^D$). Il doit être suffisamment élevé pour correspondre à la forme de la fonction, mais suffisamment petit pour limiter l'usage de la mémoire et éviter un nombre trop élevé d'évaluations pour l'initialisation de la récompense. Ainsi, le choix du paramètre M est empirique.

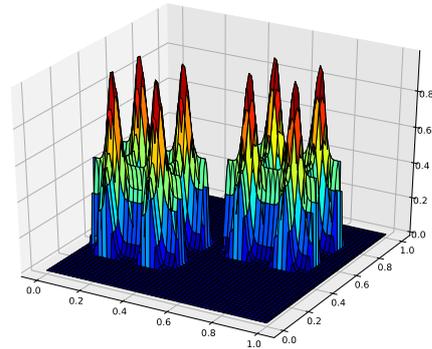
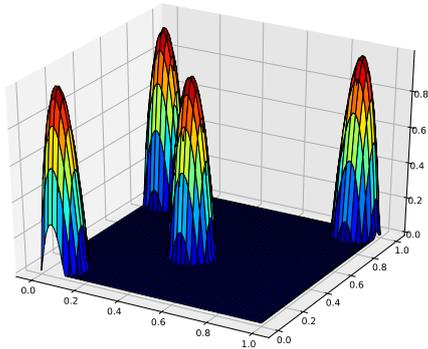
4.1.4 Fonctions de test

Dans le but d'évaluer l'algorithme proposé, plusieurs fonctions classiques ont été utilisées [123, 119]. Des fonctions modifiées modélisant des problèmes plus complexes ont été aussi utilisées (figures 4.1).

La fonction f_{sin} est une fonction sinus multidimensionnelle $[0, 1]^D \rightarrow [0, 1]$



(a) f_{Sin} avec $s = 3$ et $p = 3$ (voir [123] et équation 4.1). (b) f_{SinBasin} avec $s = 3$ et $p = 4$ (équation 4.2)



(c) f_{Hump} avec $\alpha = 2$, $Q = 4$ et $r = 0.1$ (voir [123] et équation 4.3). (d) f_{HumpSin} avec $s = 4$, $p = 2$, $z = 2$ et $r = 0.2$ (équation 4.4).

FIGURE 4.1 – Fonctions multimodales utilisées pour les expériences (représentation dans un espace 2D).

définie par :

$$f_{\text{Sin}}(\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D \sin^{2s}(p\pi\mathbf{x}_d). \quad (4.1)$$

2 paramètres permettent de régler la complexité de la fonction : p , le nombre de pics par dimension et s , définissant la largeur des pics. Cette fonction a été utilisée dans [123] et dans [119] avec $s = 3$ et $p = 5$. Elle possède p^D optima qui sont distribués équitablement (figure. 4.1a). La fonction f_{SinBasin} est similaire à, f_{Sin} mais avec un large plateau (voir Figure. 4.1b) :

$$f_{\text{SinBasin}}(\mathbf{x}) = \begin{cases} f_{\text{Sin}}(\mathbf{x}) & \text{if } \|\mathbf{x}\|_{\infty} < 0.5 \\ 0 & \text{sinon} \end{cases} \quad (4.2)$$

La fonction f_{Hump} est une version simplifiée de la fonction Hump présentée dans [123] où les pics générés aléatoirement à travers l'espace de recherche ont la même forme (figure 4.1c) :

$$f_{\text{Hump}}(x) = \max \left[0, 1 - \left(\frac{\min_q \|x - x_q\|}{r} \right)^{\alpha} \right] \quad (4.3)$$

α définit l'inverse de la netteté des pics, r le rayon des pics et Q le nombre de pics (et donc le nombre d'optima). Les points \mathbf{x}_q définissent les centres des pics et sont échantillonnés aléatoirement dans $[0, 1]^D$. Cette randomisation rend la fonction plus difficile à optimiser, mais les pics sont localement simples.

Enfin, la fonction f_{HumpSin} combine la randomisation de f_{Hump} avec la complexité locale de f_{Sin} (figure. 4.1d) :

$$f_{\text{HumpSin}}(\mathbf{x}) = \begin{cases} f_{\text{Sin}}\left(\frac{\mathbf{x} - \mathbf{x}_z + r}{2r}\right) & \text{if } \|\mathbf{x} - \mathbf{x}_z\|_{\infty} < r \\ 0 & \text{sinon} \end{cases} \quad (4.4)$$

z est le nombre de zones et r le rayon des zones. Les points \mathbf{x}_z définissent les centres des zones et sont aléatoirement distribués de façon à ne pas se chevaucher et sont évidemment dans l'espace de recherche. Le nombre d'optima est donc $z \times p^D$.

TABLEAU 4.1 – Nombre moyen d'évaluations nécessaires pour chaque algorithme pour découvrir tous les optima. Les valeurs en gras indiquent pour chaque expérience la meilleure performance. Un faible nombre d'évaluations indique un meilleur algorithme (avec l'intervalle de confiance à 95%).

Function	D	RDS	QRDS	URDS	# Optima
$f_{\text{Sin}} [s = 3, p = 5]$	2	8723 ± 972	5361 ± 396	5203 ± 438 [R = 100, M = 5]	25
$f_{\text{SinBasin}} [s = 3, p = 5]$		12251 ± 2306	5312 ± 639	4675 ± 589 [R = 1, M = 2]	9
$f_{\text{Hump}} [\alpha = 1, K = 5, r = 0.1]$		6296 ± 1399	4067 ± 485	4663 ± 903 [R = 50, M = 7]	5
$f_{\text{HumpSin}} [s = 4, p = 8, z = 2, r = 0.1]$		235579 ± 17139	164244 ± 6037	164340 ± 21293 [R = 2, M = 3]	128
$f_{\text{Sin}} [s = 3, p = 5]$	3	114298 ± 25678	78124 ± 16430	81771 ± 13131 [R = 100, M = 5]	125
$f_{\text{Sin}} [s = 3, p = 6]$		206403 ± 21129	143094 ± 12954	155076 ± 10033 [R = 100, M = 9]	216
$f_{\text{SinBasin}} [s = 3, p = 5]$		98940 ± 41683	59311 ± 12260	43959 ± 16772 [R = 0.1, M = 2]	27
$f_{\text{SinBasin}} [s = 3, p = 6]$		70355 ± 8959	57721 ± 7195	62957 ± 15797 [R = 5, M = 7]	27
$f_{\text{Hump}} [\alpha = 1, K = 5, r = 0.1]$		81840 ± 15005	71421 ± 14414	68762 ± 14225 [R = 100, M = 5]	5
$f_{\text{HumpSin}} [s = 4, p = 4, z = 2, r = 0.01]$		683312 ± 23255	561314 ± 21614	458303 ± 43231 [R = 2, M = 5]	128
$f_{\text{SinBasin}} [s = 3, p = 4]$	5	359394 ± 25454	277184 ± 24792	254704 ± 34479 [R = 2, M = 3]	32
$f_{\text{Hump}} [\alpha = 1, K = 5, r = 0.1]$		54612 ± 7034	50451 ± 7455	38566 ± 7390 [R = 2, M = 3]	5

4.1.5 Analyse expérimentale

En utilisant les fonctions précédentes, une comparaison de l'algorithme URDS avec d'autres algorithmes avec stratégie de relance tels que RDS et QRDS (section 1.3.1) est présentée. Chaque algorithme est exécuté 30 fois pour chaque expérience et les valeurs moyennes sont calculées ainsi que leurs intervalles de confiance à 95%. La première expérience consiste à mesurer le nombre d'évaluations nécessaires pour trouver tous les optima. Les résultats sont disponibles dans la table 4.1. D'après les résultats, en premier lieu, QRDS et URDS ont des performances toujours meilleures que RDS. Ceci confirme les résultats expérimentaux trouvés dans [119], à savoir la supériorité de QRDS sur RDS. Ceci permet valider le protocole expérimental. QRDS et URDS ont des performances similaires sur les problèmes en dimension 2, mais nettement meilleures que RDS. URDS surpasse QRDS lorsque la dimension augmente ($d = 3$ ou $d = 5$) sauf pour les fonctions *Sin* et *Sin_Basin* ceci pouvant être expliqué par la disposition régulière des optima à travers l'espace de recherche. Pour des fonctions plus complexes, en particulier f_{HumpSin} qui est la fonction la plus difficile expérimentée ici, URDS a des performances supérieures à QRDS notamment pour la fonction *Hump* en dimension 5 avec $\#Optima = 5$, sur cette fonction URDS a correctement estimé les zones de l'espace de recherche pour optimiser les positions initiales des recherches locales.

TABLEAU 4.2 – Proportion d'optima découverts après 10^6 évaluations (avec l'intervalle de confiance à 95%).

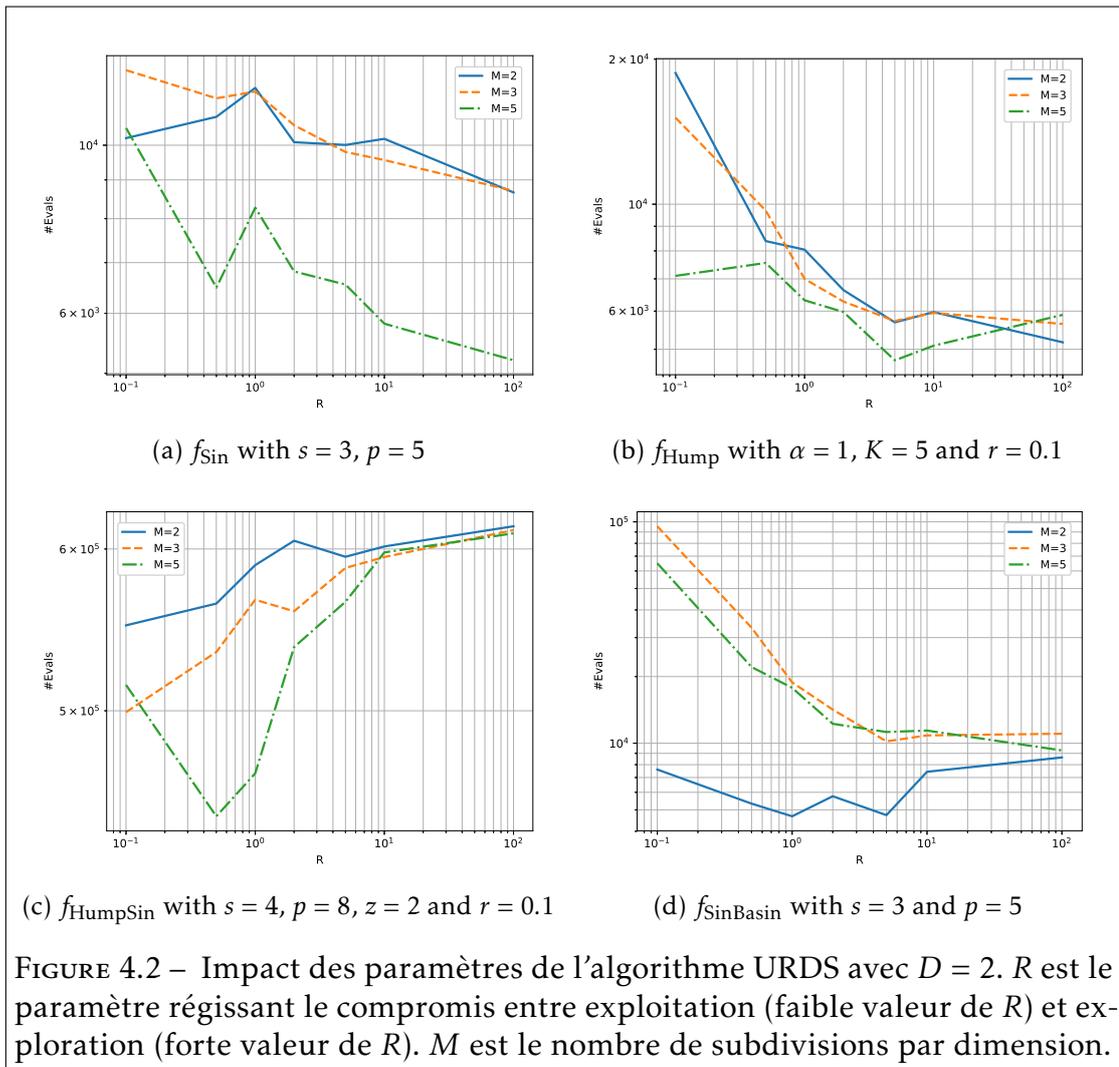
Function	D	RDS	QRDS	URDS	# Optima
$f_{\text{HumpSin}} [s = 4, p = 4, z = 2, r = 0.01]$	2	0.84%	95.31%	99.68% [$R = 0.1, M = 5$]	32
$f_{\text{HumpSin}} [s = 4, p = 8, z = 2, r = 0.1]$	3	66.7%	72%	89.7% [$R = 2, M = 5$]	1024
$f_{\text{Sin}} [s = 3, p = 4]$		91.6%	96.15%	92.49% [$R = 100, M = 3$]	1024
$f_{\text{HumpSin}} [s = 4, p = 8, z = 2, r = 0.1]$	5	$5.6 \cdot 10^{-4}\%$	$5.5 \cdot 10^{-4}\%$	$2.2 \cdot 10^{-2}\%$ [$R = 0.1, M = 3$]	65536
$f_{\text{HumpSin}} [s = 4, p = 4, z = 2, r = 0.1]$		1.74%	1.73%	28.9% [$R = 0.1, M = 3$]	2048

Dans la 2e expérience, des problèmes plus complexes sont considérés et le nombre d'optima trouvés par les 3 algorithmes, après qu'un budget de 10^6 évaluations soit écoulé, est mesuré (Table 4.2). Cette expérience est difficile, car le nombre d'optima à découvrir est important. Un bon compromis exploration/exploitation est crucial pour obtenir une performance élevée. D'après les résultats, hormis pour la fonction *Sin* en dimension 5, URDS est en mesure de trouver un nombre d'optima plus important que QRDS. L'écart devient important lorsque le nombre d'optima est conséquent, par exemple avec $f_{\text{HumpSin}}[s = 4, p = 8, z = 2, r = 0.1]$ le facteur de gain est de 40 et pour $f_{\text{HumpSin}}[s = 4, p = 4, z = 2, r = 0.1]$ le pourcentage moyen d'optima trouvé passe de 1.73% à environ 28.9%. Ces résultats démontrent la pertinence de la zone de sélection implémentée dans URDS. L'algorithme apprend les zones où les optima sont concentrés. Comme il était espéré, les 3 algorithmes ont des performances assez semblables pour f_{Sin} , parce que la fonction a distribué régulièrement les optima. Dans les tables 4.1 et 4.2, les résultats d'URDS sont les meilleurs résultats obtenus parmi les différentes instances des problèmes. Par exemple, pour la première expérience, le gain moyen d'URDS par rapport à QRDS est de 9.1%. Pour la deuxième expérience, pour la fonction $f_{\text{HumpSin}}[s = 4, p = 8, z = 2, r = 0.1]$ le facteur de gain est d'environ 40 et pour la fonction $f_{\text{HumpSin}}[s = 4, p = 4, z = 2, r = 0.1]$ le pourcentage d'optima découverts par URDS est de 29.8% contre 1.73% pour QRDS. Pour les autres instances de la 2e expérience, le gain est toujours statistiquement supérieur à celui de QRDS. Le paramètre R (compromis exploration/exploitation) est particulièrement intéressant à considérer puisqu'il influence clairement les performances. En effet, avec les fonctions où les optima sont régulièrement distribués (comme

pour f_{Sin} et f_{Hump} en petites dimensions) le meilleur résultat est obtenu avec une valeur de R importante ce qui signifie une politique favorisant l'exploration. Sur les fonctions où les optima sont concentrés dans de petites zones telles que f_{SinBasin} et f_{HumpSin} , les meilleurs résultats sont obtenus avec une valeur de R faible (favorisant l'exploitation). Pour illustrer l'impact des paramètres d'URDS (R et M) plus en profondeur, les jeux de paramètres de URDS sur différentes fonctions sont affichés (figure. 4.2). Ces graphes montrent que l'influence de R est liée à la régularité de la distribution des optima alors que M est lié à la forme de la distribution non régulière. Par exemple, la fonction f_{Sin} présentée sur la figure 4.2a a des optima régulièrement distribués. Ainsi le meilleur résultat est obtenu avec une valeur de R et de M importante (ce qui force la distribution des points de relance de manière équitable). De la même manière, la fonction f_{Hump} présentée sur la figure 4.2b a aussi des optima distribués régulièrement. Ainsi les meilleurs résultats sont obtenus avec une forte valeur de R et de M . La fonction f_{HumpSin} présentée dans la figure 4.2c a ses optima concentrés sur 2 zones. Le meilleur résultat est donc obtenu avec une valeur faible de R (exploitation) et une valeur forte de M (ce qui permet à l'algorithme de mieux exploiter ces zones). Enfin, la fonction f_{SinBasin} présentée sur la figure 4.2d a ses optima régulièrement distribués, mais seulement sur le premier quadrant de l'espace de recherche. Le paramètre le plus important est M . Avec $M = 2$ les zones d'intérêts sont parfaitement couvertes.

4.1.6 Synthèse

Dans cette contribution un nouvel algorithme d'optimisation multimodale utilisant la stratégie de relance a été proposé. Cet algorithme peut gérer des fonctions complexes, par exemple localement complexes, ou bien des fonctions avec les optima distribués de manière non uniforme à travers l'espace de recherche. Il partitionne l'espace de recherche et modélise les successives sélections des zones de redémarrage des recherches locales comme un problème de bandit manchot résolu avec la méthode de l'UCB afin de maintenir un compromis entre l'exploration et l'exploitation des différentes zones de l'espace de recherche. Ceci permet à l'algorithme de sélectionner plus fréquemment des



zones, a priori intéressantes pour les positions des recherches locales. Expérimentalement, l'algorithme surpasse l'algorithme QRDS, surtout quand la fonction à optimiser est complexe (localement complexe, plateaux ...). L'inconvénient de cette approche est qu'elle est limitée à une utilisation pour des dimensions modérées, puisque le partitionnement régulier de l'espace de recherche des zones à considérer augmente de manière exponentielle avec la dimension. Un autre inconvénient est la nécessité de paramétrer les variables de l'algorithme URDS, à savoir le nombre de divisions par dimension et la constance d'exploration de la formule UCB.

4.2 Stratégie de restart assistée par recherche arborescente

Introduction Cette section présente un nouvel algorithme d'optimisation numérique multimodale dans un contexte boîte noire qui est une continuité des travaux de l'algorithme Ucb Random-restarts with Decreasing Step-size. Le problème du passage à l'échelle de ce dernier limite son utilisation à des problèmes d'optimisation de faible dimension. Ce nouvel algorithme, Upper Confidence Tree for Random restarts with Decreasing Step-size, propose l'utilisation de la méthode de renforcement Upper Confidence Tree réalisant un partitionnement hiérarchique de l'espace de recherche. L'objectif de cet algorithme est de maintenir un compromis entre la sélection des régions intéressantes et l'exploration de l'espace de recherche. Contrairement à la contribution précédente, il est utilisable pour des fonctions de plus grande dimension, car le nombre de régions à considérer n'augmente pas exponentiellement avec la dimension grâce à la recherche arborescente. Comme cet algorithme est une amélioration de la contribution précédente, la comparaison de performance de cet algorithme est effectuée avec le même algorithme à savoir Quasi Random with Decreasing Step size, sur des fonctions de tests issues de la littérature avec des dimensions supérieures aux travaux précédents. Cette étape est importante pour valider la méthodologie et l'algorithme.

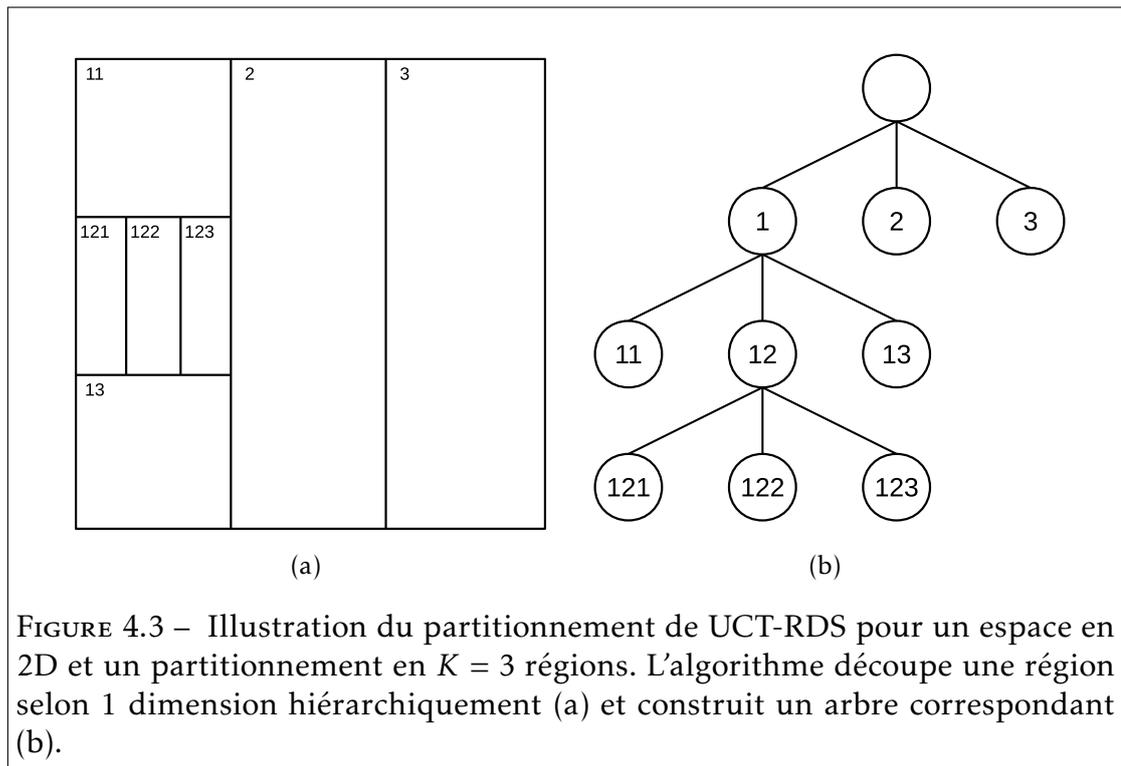
4.2.1 Motivations

L'algorithme présenté ici est une continuation des travaux de [36] (section 4.1). Dans [101], Munos propose un algorithme déterministe arborescent pour l'optimisation de problème mono modal, Search Optimistic Optimization (SOO). Cet algorithme recherche l'optimum global d'une fonction en construisant incrémentalement un partitionnement hiérarchique de l'espace de recherche. À chaque itération, il sélectionne la meilleure "feuille/nud" en fonction de la fitness associée au point central de la zone correspondante. Le meilleur nud (d'après la fitness) est sélectionné et le processus est répété jusqu'à ce qu'une feuille soit atteinte (politique arborescente). Cette dernière est ensuite étendue c'est-à-dire

qu'on effectue une partition de la région correspondante selon la dimension la plus grande. L'algorithme peut ainsi trouver l'optimum global sans connaissance de la rugosité de la fonction, mais ses performances sont médiocres sur des fonctions complexes ou lorsque la précision demandée est importante [109]. Dans [130] Valko *et al.* proposent une modification de cette méthode Stochastique Search Optimistic Optimization (StoSOO) où les nuds sont sélectionnés en utilisant une méthode stochastique construisant des bornes supérieures de confiance (upper confidence bounds) à travers les différentes partitions de l'espace de recherche. Cependant, l'algorithme n'implémente pas de recherche locale dans la région considérée et ne gère pas les problèmes multimodaux.

4.2.2 Upper Confidence Tree for Random restarts with Decreasing Step-size

L'algorithme d'optimisation multimodale proposé est Upper Confidence Tree for Random restarts with Decreasing Step-size (UCT-RDS). C'est une amélioration de l'algorithme RDS (section 1.3.1). Ce dernier est couplé avec un algorithme d'apprentissage par renforcement afin d'apprendre quelle(s) région(s) de l'espace de recherche il est intéressant de sélectionner pour la relance de la recherche locale. Ceci est réalisé par une méthode de renforcement UCT réalisant un partitionnement hiérarchique de l'espace de recherche. Tout comme les algorithmes RDS et QRDS, UCT-RDS effectue itérativement des recherches locales (Algorithme 5) jusqu'à ce que tous les optima soient découverts ou que le budget d'évaluations soit écoulé. Cependant, le point de relance des recherches locales de UCT-RDS ne se fait pas sur l'intégralité de l'espace de recherche, mais dans une de ses partitions. L'objectif principal de cet algorithme est de maintenir un compromis entre la sélection des régions intéressantes et l'exploration de l'espace de recherche. De plus, il est utilisable pour des fonctions de plus grandes dimensions, c'est-à-dire que le nombre de régions à considérer n'augmente pas exponentiellement avec la dimension. À cette fin, l'algorithme utilise un partitionnement similaire à Munos [101] : itérativement, l'algorithme sélectionne et partitionne en plusieurs K parties selon une dimension (figure 4.3a). Cette succession de partitionnements de l'espace de recherche peut être repré-



sentée sous forme arborescente (figure 4.3b). Ainsi, si l'on parcourt l'arbre jusqu'à atteindre une feuille, on obtient une région que l'on peut échantillonner pour la relance de la recherche locale. L'algorithme UCT (section 3.2.2) permet de sélectionner une région intéressante, c'est-à-dire ayant le meilleur compromis, à un instant t , entre une espérance de gain élevée et une exploration de l'espace de recherche maintenue.

L'algorithme UCT-RDS est détaillé dans l'algorithme 10. Il possède deux paramètres. Le paramètre K correspond au nombre de divisions effectuées à chaque partitionnement de l'espace de recherche et le paramètre correspondant à k_{UCT} une variable réglant le compromis exploration/exploitation. Il comprend la version initiale de l'algorithme UCT dans laquelle les états correspondent aux régions de l'espace de recherche et la politique par défaut est la recherche locale. Initialement, le nud racine (root) correspond à tout l'espace de recherche ensuite l'arbre est construit itérativement en répétant les 4 étapes suivantes. À l'étape de sélection, l'arbre est parcouru au travers de ses branches jusqu'à atteindre une feuille en appliquant la politique de recherche arbores-

cente. Vient ensuite l'étape d'expansion où un nouveau nud est créé et la région considérée est partitionnée selon un découpage orthogonal. Ce partitionnement s'effectue de la même manière qu'avec les algorithmes SOO et StoSOO : l'espace de recherche considéré est divisé selon une dimension créant K sous-ensembles. L'étape de la simulation réalise une recherche locale. Enfin la dernière étape est celle de la rétro propagation (backpropagation). Tous les nuds qui ont été parcourus pour atteindre la feuille vont voir leurs variables internes mises à jour en fonction du résultat de l'étape de simulation. Si l'on prend comme exemple la figure 4.3, l'initialisation de l'algorithme commence par un partitionnement horizontal générant les nuds 1,2 et 3. La première itération de la boucle parcourt l'arbre, sélectionne le nud 1 et découpe son espace de recherche verticalement donnant 11, 12 et 13. Durant cette phase une recherche locale est réalisée quand un nouveau nud est créé.

4.2.3 Analyse expérimentale

4.2.4 Fonctions de test

Afin d'évaluer les performances de l'amélioration proposée, l'algorithme a été testé sur trois fonctions multimodales et la comparaison a eu lieu avec l'algorithme QRDS. À l'origine, plusieurs fonctions synthétiques simples telles que *ci-gar* et *sinus* ont été considérées. Sont présentés ici les résultats sur des fonctions plus réalistes (f_{Hump} , f_{HumpSin} et f_{Icop}) puisqu'elles représentent plus fidèlement les problèmes du monde réel, où les optima ne sont pas distribués régulièrement dans l'espace de recherche. De plus, ces fonctions sont souvent utilisées dans la littérature [123, 119, 36, 76], ce qui permet de réaliser un benchmark identique avec URDS pour les fonctions de faible dimension, mais aussi d'être en mesure de comparer l'algorithme proposé avec d'autres algorithmes de la littérature.

La première fonction utilisée dans ces expériences est la f_{Hump} (voir 4.1.4). En suivant [123, 119, 36], les paramètres de la fonction pour toutes les expériences sont : $h = 1$, $r = 1.45$ et $\alpha = 1$. L'avantage principal de cette fonction est la non-possibilité de surapprentissage du fait de la randomisation des pics. La

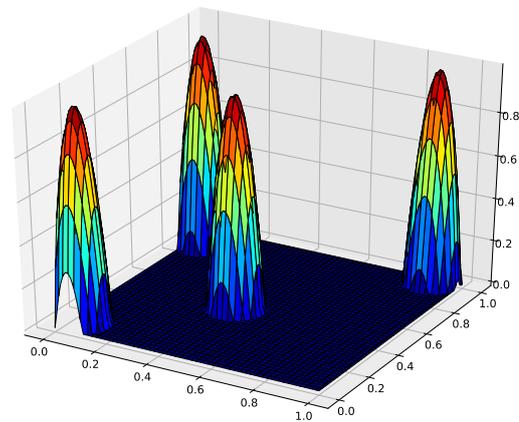
Input : $f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \epsilon_x$
Output : $\hat{\mathbf{X}}$

```

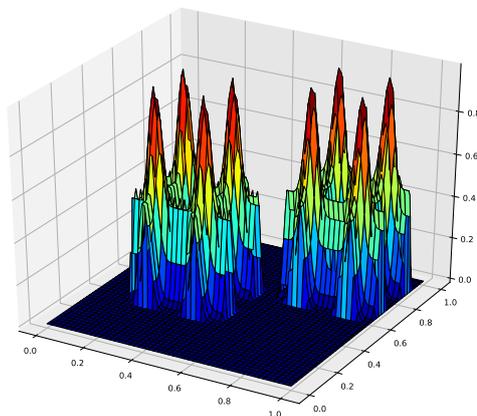
begin
   $\hat{\mathbf{X}} \leftarrow \emptyset$ 
  { initialization }
   $s_0 \leftarrow$  create root node (corresponding to the whole search-space)
   $\bar{\mathcal{A}}(s_0) \leftarrow$  split the region of  $s_0$  according to one dimension
  { UCT episodes }
  repeat
    { selection }
     $s_t \leftarrow s_0$ 
    while  $\bar{\mathcal{A}}(s_t) = \emptyset$  do
       $s_t \leftarrow \arg \max_a Q_{UCT}(s_t, a) + k_{UCT} \sqrt{\frac{\log N(s_t)}{N(s_t, a)}}$ 
       $a_t \leftarrow$  the selected  $a$ 
    end
    { expansion }
     $s_T \leftarrow$  create a child node of  $s_t$  from any  $a_t \in \bar{\mathcal{A}}(s_t)$ 
     $\bar{\mathcal{A}}(s_T) \leftarrow$  split the region of  $s_T$  according to one dimension
    { simulation }
     $\mathbf{x} \leftarrow$  sample the region of  $s_T$ 
     $\hat{\mathbf{X}} \leftarrow \text{SearchDS}(f, \sigma_0, \epsilon_\sigma, y^*, \epsilon_y, \mathbf{x}, \epsilon_x, \hat{\mathbf{X}})$ 
    { backpropagation }
     $r_T \leftarrow$  1 if a new optima found, 0 otherwise
    repeat
       $N_{s_t, a_t} \leftarrow N_{s_t, a_t} + 1$ 
       $Q_{UCT}(s_t, a_t) \leftarrow Q_{UCT}(s_t, a_t) + \frac{R_t - Q_{UCT}(s_t, a_t)}{N(s_t, a_t)}$ 
       $s_t \leftarrow$  parent node of  $s_t$ 
    until  $s_t = s_0$ 
  until All optima has been found OR number of evaluation has been reach
end

```

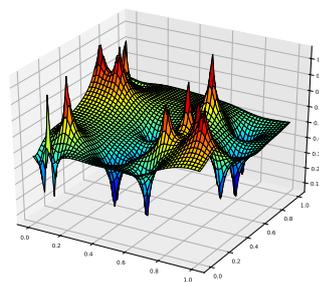
Algorithm 10 : Upper Confidence Tree for Random restarts with Decreasing Step-size



(a) f_{Hump} avec $\alpha = 1.0$, $r = 0.1$, $Q = 4$ (voir Eq. 4.3).



(b) f_{HumpSin} avec $s = 4$, $p = 2$, $z = 2$ and $r = 0.2$ (voir Eq. 4.4).



(c) f_{icop} avec $u_i = 1.0$, $\Omega = 20$, $p = 1.0$ (see Eq. 4.5).

FIGURE 4.4 – Fonctions multimodales utilisées dans les expériences (représentation en 2D).

deuxième fonction est la f_{HumpSin} (voir 4.1.4), proposée dans [36]. C'est une variation de la f_{Hump} où les pics ont des formes plus compliquées. L'avantage de la randomisation de f_{Hump} est conservé, mais au lieu d'avoir un seul optimum par pic f_{HumpSin} possède des pics sinusoïdaux à D -dimensions (voir équation. 4.4, équation. 4.1 et figure. 4.1d). Cette fonction bien plus difficile à optimiser que f_{Hump} est un réel challenge pour les algorithmes d'optimisation multimodale. Dans les expériences réalisées $p = 2$. La dernière fonction est f_{Icop} , proposée par [76]. Elle définit une classe de problèmes d'optimisation avec un paramétrage des caractéristiques du paysage (équation 4.5 et figure 4.4c).

$$f_{\text{Icop}}(\mathbf{x}) = \begin{cases} u_i & \text{if } \exists i, \|\mathbf{x} - \mathbf{x}_i\| \approx 0 \\ \frac{\sum_{i=1}^{\Omega} \frac{u_i}{\|\mathbf{x} - \mathbf{x}_i\|^p}}{\sum_{i=1}^{\Omega} \frac{1}{\|\mathbf{x} - \mathbf{x}_i\|^p}} & \text{sinon} \end{cases} \quad (4.5)$$

Pour créer une fonction f_{Icop} , Ω optima globaux \mathbf{x}_i sont générés avec leur fitness respective u_i . Si une fonction multimodale est souhaitée, la moitié des points de \mathbf{x}_i peuvent être définis comme optima globaux (c'est-à-dire $u_i = 1$) et l'autre moitié comme optima locaux (c'est-à-dire $0 \leq u_i \leq u_l < 1$). La finalité de la fonction Icop est de générer des classes de problèmes d'optimisation paramétrables :

- Ω : nombre d'optima (locaux ou globaux),
- u_i : fitness des i -ième optima globaux,
- u_l : fitness maximale autorisée pour les optima locaux,
- p : influence des solutions sur l'interpolation.

Dans les expériences ont été utilisées $u_i = 1$, $p = 1$ et différentes valeurs de Ω et u_l ont été testées.

4.2.5 Résultats

La comparaison des algorithmes UCT-RDS avec QRDS [119] a été réalisée en utilisant les fonctions définies précédemment. Chaque expérience a été réalisée 100 fois afin de rendre la comparaison statistiquement significative. Le budget d'évaluations alloué pour chaque fonction est de 10^6 évaluations. Pour les recherches locales, le seuil d'acceptation de la fitness vaut $\epsilon_y = 0.00001$ et le seuil d'acceptation de la position dans l'espace de recherche est $\epsilon_x = 0.001$. L'ajuste-

ment des paramètres des algorithmes proposés a été effectué suite à une étude empirique, car ces paramètres influencent les performances des algorithmes et ont aussi été spécifiés dans d'autres études [38, 127]. Les résultats rapportés ci-dessous correspondent à la moyenne et à l'intervalle de confiance ϵ , définie par : $\epsilon = \sigma/\sqrt{n}$, où σ est la l'écart type et n le nombre d'expériences (ici, $n = 100$).

La première expérience consiste à trouver le nombre maximal d'optima avec un budget d'évaluations (Table 4.3). Tout d'abord sur la fonction f_{Icop} les résultats de l'algorithme UCT-RDS sont au moins aussi bons qu'avec l'algorithme QRDS. Il est bon de rappeler que f_{Icop} est un test de robustesse pour UCT-RDS. Par définition, il n'y a pas de zone(s) prédéfinie(s) qui puisse(nt) être plus importante(s) qu'une(que plusieurs) autre(s), puisque les optima sont distribués aléatoirement. UCT-RDS est assez robuste puisque possédant des performances similaires à QRDS. UCT-RDS est même meilleur sur certains paramétrages de la fonction $Icop$. Cela peut s'expliquer dans le cas où un groupe d'optima est formé dans une même zone impliquant qu'UCT peut idéalement apprendre l'importance de cette zone. Des résultats similaires peuvent être observés sur la fonction f_{Hump} . Des paramètres élevés ont été utilisés (dimension 35 et 50 optima) et les résultats de UCT-RDS indiquent que ce dernier découvre 7 fois plus d'optima que QRDS en moyenne. Sur la fonction $f_{HumpSin}$, les performances de UCT-RDS sont aussi meilleures que QRDS avec deux fois plus d'optima trouvés. Pour rappel, cette fonction est très difficile à optimiser et les précédents algorithmes ont eu des performances très limitées [36]. Avec UCT-RDS il est possible de trouver 314 des 2048 optima en 5 dimensions alors que QRDS est capable d'en trouver que 12 en moyenne. En dimension 8, avec un paramétrage plus facile, UCT-RDS est capable de trouver 2 fois plus d'optima que QRDS (67.5 contre 32.9 en moyenne). Enfin, sur la figure 4.5 est étudiée l'influence des paramètres de UCT-RDS sur la fonction $f_{HumpSin}$ en dimension 8. Pour rappel cet algorithme a 2 paramètres : le nombre de partitions K à définir sur une région et le paramètre k_{UCT} réglant le compromis exploration/exploitation. En premier lieu, on peut observer que le nombre de partitions est important. Dans les expériences, il est reconnu que K doit légèrement augmenter avec la dimension. Deuxièmement, le paramètre k_{UCT} est très important. Une grande valeur tend à favoriser l'exploration tandis qu'une petite valeur tend à favoriser l'ex-

TABLEAU 4.3 – Nombre moyen d'optima découverts par chaque algorithme après 10^6 évaluations sur 100 exécutions (un chiffre élevé indique une meilleure performance). Différents paramètres ont été testés, en particulier pour f_{Icop} , afin d'évaluer la robustesse de UCT-RDS.

Function	D	QRDS	UCT-RDS	#Optima
f_{HumpSin} [$z = 4, r = 0.1$]	5	12.31 ± 0.25	314.64 ± 6.2 ($K = 3, k_{UCT} = 0.1$)	2048
f_{Icop} [$\Omega = 50, u_l = 0.9$]	5	11.64 ± 0.3	12.6 ± 0.25 ($K = 2, k_{UCT} = 6.4$)	25
f_{Icop} [$\Omega = 80, u_l = 0.9$]	5	12.7 ± 0.1	13.6 ± 0.1 ($K = 2, k_{UCT} = 1.0$)	40
f_{HumpSin} [$z = 2, r = 0.22$]	8	32.9 ± 0.3	67.5 ± 0.8 ($K = 2, k_{UCT} = 0.8$)	512
f_{Icop} [$\Omega = 100, u_l = 0.25$]	10	34.6 ± 3.4	39.2 ± 3.9 ($K = 7, k_{UCT} = 1.4$)	50
f_{Icop} [$\Omega = 96, u_l = 0.25$]	10	34.6 ± 0.4	35.9 ± 0.8 ($K = 2, k_{UCT} = 6.4$)	48
f_{Icop} [$\Omega = 128, u_l = 0.25$]	10	40.44 ± 0.8	40.4 ± 0.8 ($K = 3, k_{UCT} = 1.4$)	64
f_{Icop} [$\Omega = 144, u_l = 0.25$]	10	40.48 ± 0.9	40.45 ± 0.84 ($K = 2, k_{UCT} = 1.4$)	72
f_{Icop} [$\Omega = 24, u_l = 0.25$]	20	11.09 ± 0.1	11.24 ± 0.1 ($K = 2, k_{UCT} = 12.4$)	12
f_{Icop} [$\Omega = 60, u_l = 0.25$]	20	13.87 ± 0.4	13.94 ± 0.41 ($K = 9, k_{UCT} = 0.4$)	30
f_{Hump}	35	23.21 ± 0.3	30.21 ± 0.33 ($K = 13, k_{UCT} = 0.1$)	50

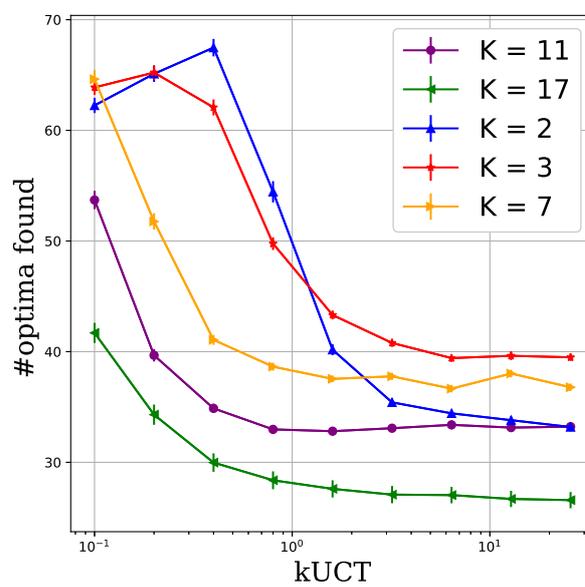


FIGURE 4.5 – Nombre d'optima découverts en fonction de la valeur de k_{UCT} (échelle en log), lorsque l'on partitionne une région en K sous-régions, sur f_{HumpSin} et en dimension 8.

ploitation. Il est reconnu que la stratégie optimale est quelque part entre une exploration extrême et une exploitation extrême, ce qui a été vérifié dans ces expériences.

4.2.6 Synthèse

Un nouvel algorithme d'optimisation multimodale sans utilisation de la dérivée de la fonction objective a été proposé. Ce dernier utilise un *Upper Confidence Tree* qui partitionne hiérarchiquement l'espace de recherche. Les sélections successives des partitions ont été modélisées comme un problème de bandit manchot. Ceci permet d'utiliser l'algorithme afin d'apprendre quelle(s) zone(s) de l'espace de recherche est (sont) intéressante(s) à considérer, le but étant de dépenser plus d'évaluations sur ces zones. L'algorithme a été testé sur 3 fonctions différentes fortement multimodales. Il en résulte qu'il est robuste, quelles que soient les dimensions étudiées ou la forme du paysage. Comparé aux autres approches, des résultats significativement meilleurs ont été observés sur toutes les fonctions de tests (en petite ou grande dimension).

Bibliographie du présent chapitre

- [3] Ali AHRARI, Kalyanmoy DEB et Mike PREUSS. « Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations ». In : *Evolutionary computation* 25.3 (2017), p. 439-471.
- [11] Anne AUGER et Nikolaus HANSEN. « A restart CMA evolution strategy with increasing population size ». In : *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. T. 2. 2005, p. 1769-1776.
- [14] Hans-Georg BEYER et Hans-Paul SCHWEFEL. « Evolution strategies—A comprehensive introduction ». In : *Natural computing* 1.1 (2002), p. 3-52.
- [31] Kenneth Alan DE JONG. « An Analysis of the Behavior of a Class of Genetic Adaptive Systems. » Thèse de doct. University of Michigan, 1975.
- [36] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Improving Multi-Modal Optimization Restart Strategy Through Multi-Armed Bandit ». In : *IEEE ICMLA 2018 : 17th IEEE International Conference On Machine Learning And Applications*. 2018.
- [37] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Upper Confidence Tree for planning restart strategies in Multi-Modal Optimization ». In : *Soft Computing* (juill. 2020). DOI : 10.1007/s00500-020-05196-w.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.
- [53] David E GOLDBERG, Jon RICHARDSON et al. « Genetic algorithms with sharing for multimodal function optimization ». In : *Genetic algorithms and their applications : Proceedings of the Second International Conference on Genetic Algorithms*. 1987, p. 41-49.
- [60] Nikolaus HANSEN et Stefan KERN. « Evaluating the CMA evolution strategy on multimodal test functions ». In : *International Conference on Parallel Problem Solving from Nature*. Springer. 2004, p. 282-291.
- [70] Serdar KADIOGLU, Meinolf SELLMANN et Markus WAGNER. « Learning a Reactive Restart Strategy to Improve Stochastic Search ». In : *International Conference on Learning and Intelligent Optimization*. Springer. 2017, p. 109-123.
- [76] Benjamin LACROIX, Lee A CHRISTIE et John A.W. MCALL. « Interpolated Continuous Optimisation Problems with Tunable Landscape Features ». In : *Proc. of GECCO 17 Companion*. 2017.

- [84] Florian LEPRÊTRE, Fabien TEYTAUD et Julien DEHOS. « Multi-armed bandit for stratified sampling : Application to numerical integration ». In : *TAAI 2017*. Taipei, Taiwan, 2017.
- [92] Samir W MAHFOUD. « Niching methods for genetic algorithms ». In : *Urbana* 51.95001 (1995), p. 62-94.
- [96] Ole J MENGSHOEL et David E GOLDBERG. « Probabilistic crowding : Deterministic crowding with probabilistic replacement ». In : *Proc. of GECCO conference*. 1999, p. 409-416.
- [101] Remi MUNOS. « Optimistic Optimization of Deterministic Functions without the Knowledge of its Smoothness ». In : *Advances in Neural Information Processing Systems* (2011), p. 783-791.
- [107] Alain PÉTROWSKI. « A clearing procedure as a niching method for genetic algorithms ». In : *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. 1996, p. 798-803.
- [109] Philippe PREUX, Remi MUNOS et Michal VALKO. « Bandits attack function optimization ». In : *IEEE Congress on Evolutionary Computation*. 2014.
- [113] Ingo RECHENBERG. « Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution ». In : (1973).
- [119] Marc SCHOENAUER, Fabien TEYTAUD et Olivier TEYTAUD. « A Rigorous Runtime Analysis for Quasi-Random Restarts and Decreasing Stepsize ». In : *Artificial Evolution*. Angers, France, 2011.
- [123] Gulshan SINGH et Kalyanmoy DEB DR. « Comparison of multi-modal optimization algorithms based on evolutionary algorithms ». In : *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 2006, p. 1305-1312.
- [127] Fabien TEYTAUD et Olivier TEYTAUD. « QR Mutations Improve Many Evolution Strategies : A Lot On Highly Multimodal Problems ». In : *Proc. of the 2016 GECCO conference*. ACM. 2016, p. 35-36.
- [130] Michal VALKO, Alexandra CARPENTIER et Remi MUNOS. « Stochastic Simultaneous Optimistic Optimization ». In : *Proc. of the 30th International Conference on Machine Learning*. T. 28. 2013.
- [139] Xiaodong YIN et Noel GERMA. « A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization ». In : *Artificial neural nets and genetic algorithms*. 1993, p. 450-457.

Troisième partie

Apprentissage supervisé de phénomènes biologiques et application à l'agriculture

Apprentissage automatique

Sommaire du présent chapitre

5.1 Principes	110
5.2 Algorithmes d'apprentissage supervisé	111
5.2.1 Les réseaux de neurones	113
5.2.2 Les forêts d'arbres décisionnels	116
5.2.3 Les Séparateurs à Vaste Marge (SVM)	118
5.3 Sélection de variables	119

Introduction Ce chapitre introduit l'apprentissage automatique en se focalisant sur l'apprentissage supervisé dans un contexte de problème de régression. Ce chapitre se divise en trois parties. La première explique le principe de l'apprentissage et définit les différentes étapes du processus d'apprentissage. La deuxième présente un éventail d'algorithmes d'apprentissage supervisé de la littérature avec des méthodes d'apprentissage différentes telles que les réseaux de neurones, les forêts d'arbres décisionnels et les séparateurs à vastes marges. Ces méthodes ont été choisies, car elles ont été appliquées pour des problématiques d'apprentissage de phénomène biologique similaires à la problématique de la prédiction de la teneur hebdomadaire en eau. La dernière partie introduit le principe de sélection de variables. Après une présentation des avantages et

limitations de cette technique, deux algorithmes de sélection de variables sont présentés.

5.1 Principes

L'apprentissage automatique ou Machine Learning (ML) en anglais est un domaine de *l'intelligence artificielle*. Il consiste à donner la capacité à des machines d'apprendre à partir de données/observations. Le concept d'apprentissage repose sur des concepts en mathématiques, en statistiques, en biologie, ... En général, on souhaite faire apprendre un phénomène (reconnaissance de la voix, prédiction des valeurs boursières, ...) en fonction de variables explicatives issues d'observations qui semblent pertinentes à la compréhension et à la prédiction du phénomène. Les variables explicatives peuvent prendre, en théorie, des natures très diverses (valeurs numériques, dates, perceptions, couleurs, adresses, positions géographiques, ...) sans pour autant être du même ordre de grandeur lorsque les variables sont numériques, ou bien a priori, sans relations ordinales, par exemple deux dates peuvent être différentes sans avoir de relation de supériorité l'une par rapport à l'autre. Ainsi il apparaît un problème de relation entre les variables qu'il convient de souligner et pour lequel une réponse possible consiste à exprimer toutes les informations numériquement (encodage) et à normaliser toutes les valeurs dans le but de rendre toutes les informations au même niveau d'importance. Si les données possèdent, pour chaque observation, un résultat attendu (appelé étiquette) alors on parle d'apprentissage *supervisé*. Dans le cas contraire, l'apprentissage est dit *non supervisé*. À noter qu'il existe d'autres types d'apprentissages : semi-supervisés, par transfert, ... Les algorithmes utilisés ne sont pas les mêmes en fonction du type d'apprentissage souhaité. Lorsque les étiquettes (c'est-à-dire les valeurs que l'on souhaite apprendre) correspondent à des valeurs continues, la tâche d'apprentissage effectuée est appelée *régression*. Lorsque les valeurs possibles appartiennent à des ensembles finis de catégories, on parle de *classification*. La figure 5.1 schématise les différences entre un problème de régression et de classification. Concernant l'apprentissage supervisé, le processus comporte deux phases :

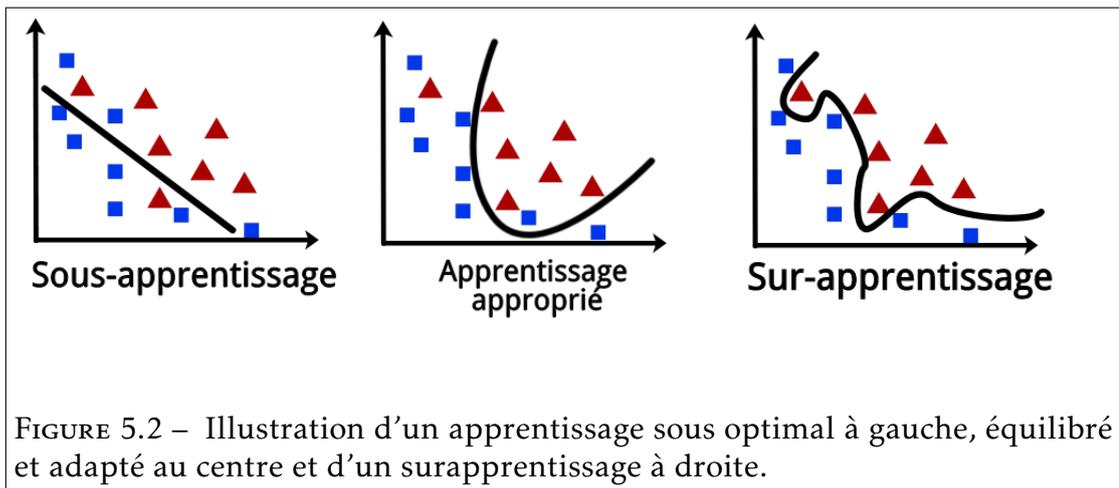
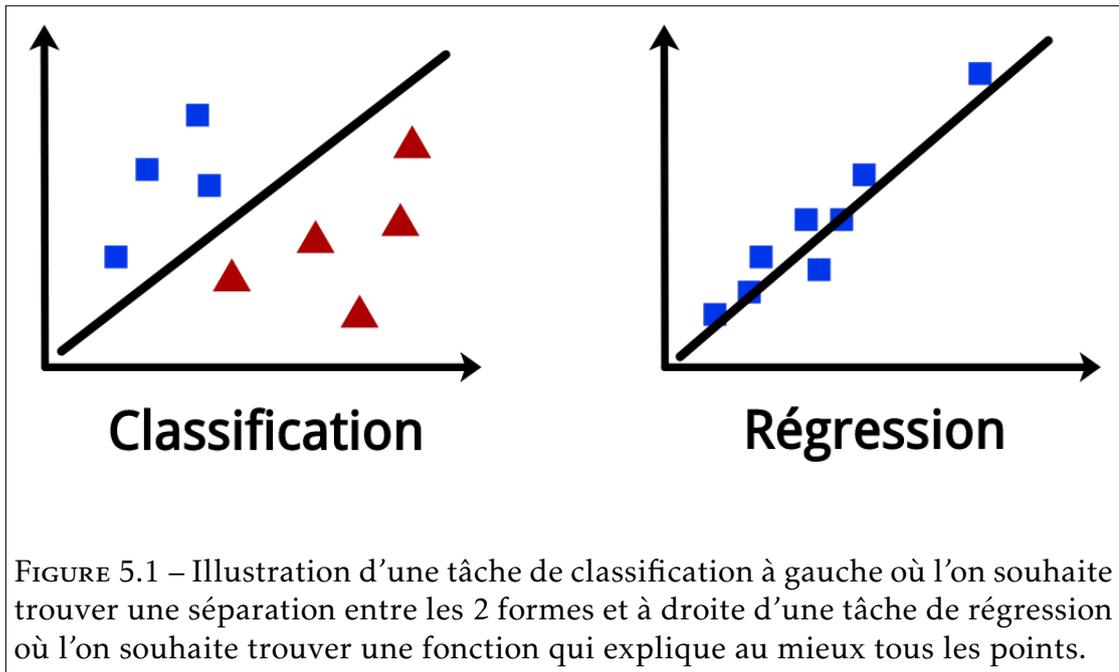
- L'entraînement (training). Dans cette phase, la tâche consiste, à partir

d'un nombre fini de données (base d'apprentissage), à générer un modèle capable d'estimer les prédictions du phénomène étudié par rapport aux observations faites. Pour ce faire, une mesure de l'erreur d'apprentissage (ou métrique) est établie. Elle peut être, par exemple, la Mean Squared Error, la Mean Absolute Error, le coefficient de corrélation, ... Ces métriques permettent à l'algorithme de quantifier l'erreur réalisée en pénalisant plus sévèrement les erreurs importantes. Ainsi, itérativement, l'algorithme peut se corriger et devenir plus efficace. Un modèle parfaitement appris possède une erreur d'apprentissage quasi nulle : $|y - \hat{y}| \equiv 0 \forall y \in Y$, avec y une observation parmi toutes les données d'apprentissage Y et \hat{y} la prédiction du modèle généré par un algorithme d'apprentissage supervisé. Durant cette phase, il est important de prendre en compte le phénomène de surapprentissage. En effet, les observations disponibles ne couvrent pas forcément toutes les réactions du phénomène et donc l'algorithme risque d'apprendre à parfaitement prédire les données d'apprentissage au risque de ne plus correctement prédire les cas non observés du phénomène (figure :5.2).

- La validation. Une fois que le modèle prédictif est appris, sa généralisation c'est-à-dire sa capacité à estimer le plus correctement possible les prédictions du phénomène sur des observations non disponibles en apprentissage (base de validation) est établie. On mesure ainsi la performance et la robustesse d'un modèle sur cette base. La métrique de l'erreur est la même que pour l'entraînement.

5.2 Algorithmes d'apprentissage supervisé

Les travaux de cette thèse portent sur l'utilisation d'algorithmes d'apprentissage supervisé appliqués aux problèmes de régression. Sont présentés ici les algorithmes ayant été utilisés au cours de la thèse.

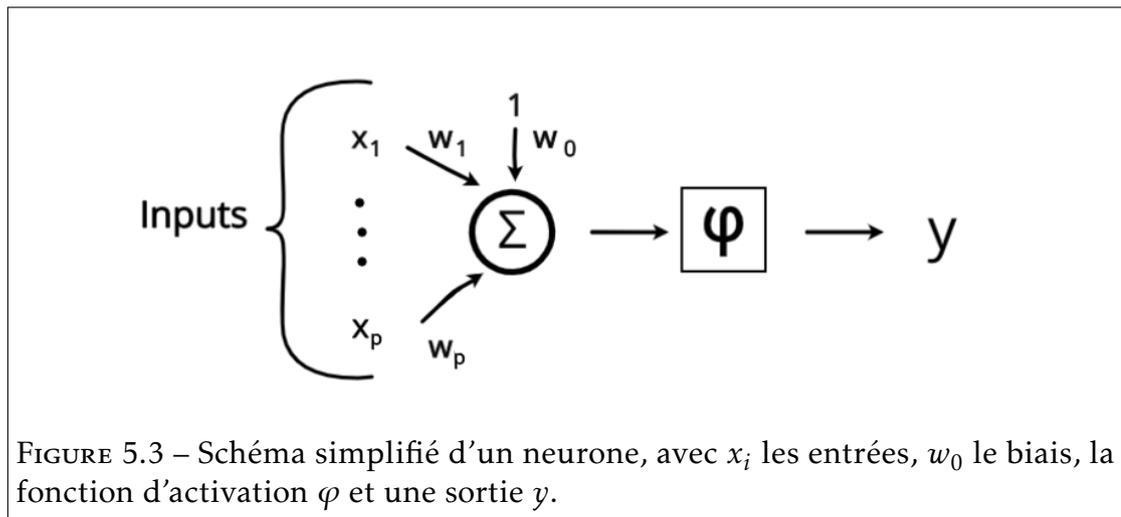


5.2.1 Les réseaux de neurones

Un réseau de neurones ou *Neural Network* en anglais est un système bio-inspiré par les réseaux de neurones biologiques[136]. L'intérêt principal de cette approche est la possibilité de synthétiser les informations fournies et de les transformer, souvent de manière non linéaire, afin de générer un modèle prédictif. Ce dernier est donc capable d'apprendre, sans connaissance a priori, des tâches spécifiques. Un réseau de neurones est donc constitué de neurones. La représentation schématique d'un neurone artificiel peut être vue sur la figure 5.3. Un neurone est constitué de :

- Une ou plusieurs entrée(s) correspondant aux signaux que reçoit le neurone. Sur la première couche, ce sont les valeurs des variables explicatives des observations qui sont transmises. À cette fin, ces données sont généralement normalisées entre elles afin de limiter les effets d'échelle résultant de l'intensité des signaux d'entrée.
- Des poids synaptiques associés à chaque entrée. Ils ont pour rôle d'accentuer ou au contraire d'atténuer l'intensité des signaux d'entrée. Ces valeurs sont modifiées lors de la phase d'apprentissage.
- Un biais. C'est une entrée constante dont l'intérêt est de reproduire un comportement de base.
- Une fonction d'activation. Elle est définie initialement et résulte d'un choix de conception du réseau de neurones. En fonction du type d'apprentissage et/ou des données, la fonction d'activation peut avoir un impact sur les performances d'apprentissage du modèle. L'objectif de cette fonction est de transformer la somme des signaux d'entrée combinés à leurs poids synaptiques en un signal de sortie.

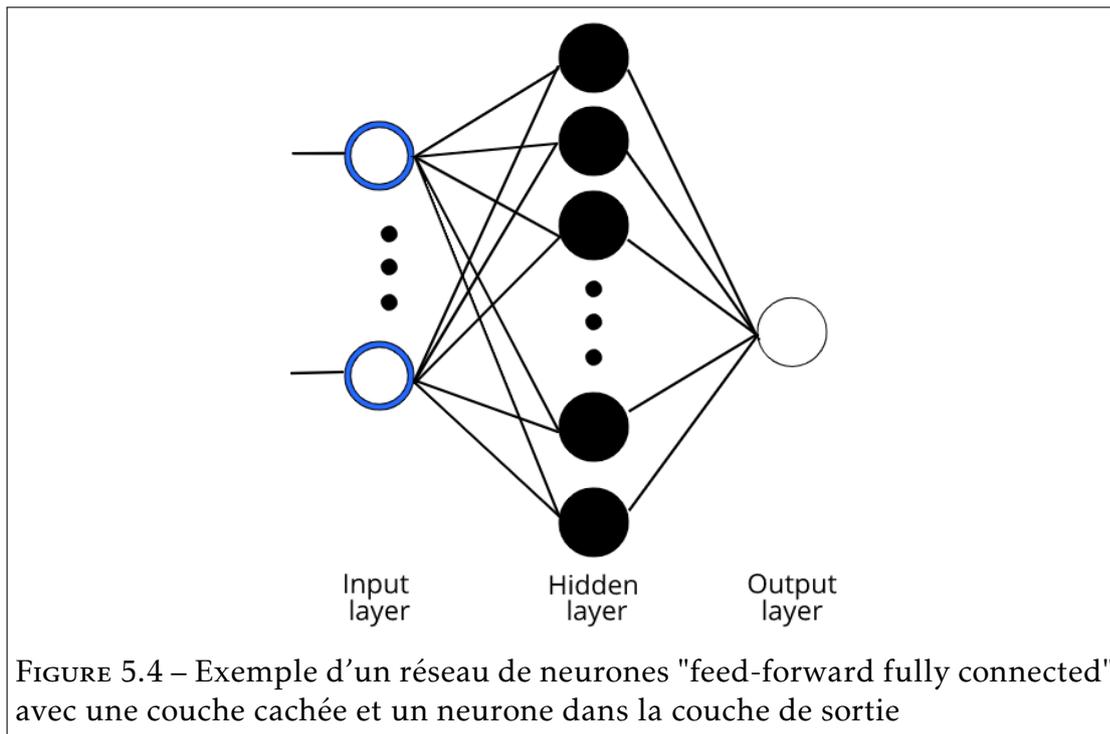
L'équation 5.1 explique comment un neurone effectue une opération de calcul interne comme étant la somme des entrées x_i multipliée par leurs poids correspondants w_i , et de l'ajout du biais w_0 . Le résultat est ensuite passé comme paramètre à la fonction d'activation φ qui peut être, par exemple une fonction sigmoïde, une fonction relu, une fonction tangente hyperbolique, ... De manière formelle, la valeur de sortie finale y du neurone avec p entrées possédant une



fonction d'activation φ est calculée ainsi :

$$y = \varphi \left(\sum_{i=0}^p x_i w_i \right) \quad (5.1)$$

Les neurones sont souvent regroupés en couches : *une couche d'entrée, une ou plusieurs couches cachées et une de sortie*. Le choix de la topologie d'un réseau de neurones est un domaine très étudié [64]. Il repose, le plus souvent, sur une étude empirique ainsi que sur l'expérience de son concepteur. Cependant, une considération générale est qu'un nombre important de neurones permet, en théorie, des performances meilleures tandis qu'un nombre plus restreint permet un modèle avec une meilleure capacité de généralisation. Dans le cas plus courant, l'architecture est *feed forward*, c'est-à-dire que les neurones de la même couche partagent les mêmes entrées, mais ne sont pas connectés entre eux. Les valeurs de sortie des neurones de cette couche servent alors comme valeurs d'entrées de la couche suivante. Dans ce manuscrit, le réseau de neurones utilisé est "*feed-forward fully connected*", ce qui signifie que chaque neurone appartenant à une couche est connecté à tous les neurones de la couche suivante sans qu'il n'y ait de boucle ou de cycle et donc de transmission d'informations d'une couche supérieure vers une couche inférieure. Ainsi, les informations sont transmises depuis la couche d'entrée vers la couche de sortie en passant par la/les couche(s)



cachée(s) (figure 5.4).

$$\varphi(s) = (1 - \exp^{-s}) / (1 + \exp^{-s}) \quad (5.2)$$

La couche de sortie est composée d'un ou plusieurs neurone(s) représentant la ou les différente(s) prédiction(s) souhaitée(s). Les réseaux de neurones sont des algorithmes d'apprentissage et nécessitent une phase d'apprentissage (section 5.1). Durant cette phase, le processus est le plus souvent supervisé. Ainsi, grâce à une base d'observations connues ayant les résultats attendus, le réseau de neurones peut être entraîné grâce à un processus itératif d'optimisation utilisant le gradient. Plus les exemples sont variés, meilleure est la capacité de généralisation et donc de prédiction du modèle. L'apprentissage utilise l'algorithme de la rétropropagation [136] (backpropagation) comme méthode d'ajustement des poids synaptiques associés à chaque neurone. L'erreur entre les valeurs prédites et les valeurs réelles est propagée depuis la couche de sortie jusqu'à la couche d'entrée en ajustant la valeur des poids synaptiques de chaque neurone. La généralisation du réseau est réalisée de la même manière sur une base de test,

c'est-à-dire sur des observations non disponibles dans la base d'apprentissage. Lorsque peu de données sont disponibles, il y a risque de surapprentissage, c'est-à-dire que le modèle peut apprendre les bruits statistiques et les observations et non le phénomène, ce qui peut conduire à des performances médiocres en généralisation. L'une des méthodes pour gérer ce problème consistant à combiner plusieurs réseaux avec des topologies différentes sur un même ensemble de données réduit le risque de surapprentissage. Malheureusement, en pratique, il est difficile à utiliser. Une méthode de régularisation pour éviter ce phénomène est le *dropout* [102]. L'idée sous-jacente est de simuler de nombreuses topologies à partir d'un seul réseau. Une autre façon d'éviter le surapprentissage consiste à régulariser les poids des neurones grâce aux normes L1 et L2 afin d'éviter qu'un ou plusieurs neurone(s) ne prenne(nt) trop de poids dans le réseau. Dans les travaux de cette thèse, l'implémentation du réseau de neurones a été réalisée grâce à la librairie Keras.¹

5.2.2 Les forêts d'arbres décisionnels

Un arbre de décision est un algorithme d'apprentissage supervisé invariant au passage à l'échelle, c'est-à-dire à l'augmentation du nombre de variables et de multiples transformations de valeurs des variables. Il est robuste à l'inclusion de variables non pertinentes et produit des modèles inspectables, c'est-à-dire qu'il est possible d'expliquer son comportement. Cependant, les arbres de décision sont rarement précis. En outre, si l'arbre grossit trop, autrement dit si de nombreuses variables sont présentes, il y a risque de surapprentissage. La première description de "Random Forest" est une combinaison de l'idée de bagging de Breiman [17] et d'une sélection de variables introduite par Ho [62]. Une forêt d'arbres décisionnels ou Random Forest en anglais est donc une méthode d'apprentissage ensembliste combinant le concept de sous-ensemble aléatoire et de bagging. L'apprentissage est effectué par un algorithme générant un ensemble d'arbres de décision entraînés sur des sous-ensembles de données observables et de variables explicatives en parallèle issues de la base d'apprentissage. La construction des arbres de décision est faite en utilisant une méthode de boos-

1. <https://keras.io/>

trap afin de mélanger aléatoirement des variables et des observations. Dans le contexte de la régression, la prédiction d'observations est déterminée par la moyenne des prédictions des sorties des arbres de la forêt. Cette approche permet d'améliorer les performances des arbres de décision et de réduire la variance sans augmenter le biais. Comme pour les réseaux de neurones, la généralisation est réalisée sur la base de tests en utilisant la même métrique que celle d'apprentissage.

Input : n : number of sample, S : a training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$ F : features, B : number of trees in forest.
Output : A learned random Forest regressor
begin
 $H \leftarrow \emptyset$
 for $i \in 1 \dots B$ **do**
 $S(i) \leftarrow$ A bootstrap sample from S
 $h_i \leftarrow$ RandomizedTreeLearn($S(i), F$)
 $H \leftarrow H \cup h_i$
 end
 return H
end

Algorithme 11 : Forêt d'arbres décisionnels

Input : S : a training set $(x_1, y_1), \dots, (x_n, y_n)$, F : features.
Output : A learned decision tree
begin
 $f \leftarrow$ subset of F
 foreach *node* **do** Split on best feature in f
 return The learned tree
end

Algorithme 12 : RandomizedTreeLearn

L'algorithme Random Forest est décrit par les algorithmes 11 et 12. Dans les travaux de la thèse, l'implémentation de l'algorithme Random Forest a été réalisée avec la librairie scikit-learn [80]. Ce dernier possède 3 paramètres principaux qu'il convient de régler :

- Le nombre minimum de feuilles par arbre, c'est-à-dire le nombre minimum de variables présentes dans l'arbre de décision.

- Le nombre d'arbres de décision présents dans la forêt. Plus la forêt est grande, meilleure est la prédiction, sous réserve qu'au moins la moitié des arbres soient capables de prédire correctement.
- Le nombre maximum de variables attribuées pour un arbre.

Le choix des paramètres peut être effectué par une technique de *grid search* mais repose sur l'expérience du concepteur.

5.2.3 Les Séparateurs à Vaste Marge (SVM)

Reposant sur des considérations théoriques de Vladimir Vapnik et sur le développement d'une théorie statistique de l'apprentissage de Vapnik-Chervonenkis, la méthode *Support Vector Machine* ou SVM a rapidement été adoptée pour sa capacité à travailler avec des données de grandes dimensions tout en ayant un nombre faible d'hyperparamètres à régler. De plus, SVM apporte une garantie théorique et de bons résultats en pratique. À l'origine développé, pour les tâches de classification [133], le SVM peut être aussi utilisé pour les tâches de régression [35]. Dans ce cas, la méthode porte le nom de *Support Vector Regression* ou SVR. Le principe de fonctionnement des SVM repose sur la discrimination des données suivant une frontière la plus simple possible appelée fonction d'approximation. La distance ou marge entre les différents groupes d'observation de part et d'autre de la séparatrice doit être maximale afin de garantir une meilleure robustesse face aux bruits, et donc de proposer un modèle avec une grande capacité de généralisation. Les données les plus proches de la frontière sont appelées vecteurs de support. L'hypothèse de la présence d'une frontière suppose que les données soient linéairement séparables, ce qui est rarement le cas en pratique. Ceci fait intervenir la seconde notion mise en place dans les SVM, à savoir le *kernel trick*. Ce sont des fonctions noyau qui doivent respecter les conditions du théorème de Mercer et qui ont l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace. Ces fonctions à noyau permettent de séparer les données en les projetant dans un espace vectoriel de plus grande dimension ou *feature space*.

La méthode de SVM peut être appliquée à de nombreux problèmes d'apprentissage et, en fonction des data, peut avoir des performances équivalentes

ou supérieures aux réseaux de neurones. Plus formellement, dans le cas de la régression, soit un ensemble d'apprentissages $S = \{(x_{s_1}, y_1), \dots, (x_n, y_n)\}$, l'objectif est de trouver une fonction d'approximation f aussi proche que possible de y . Si l'erreur du modèle est nulle cela implique que $f(x)$ explique parfaitement y . La méthode SVR suppose qu'il est possible d'accepter une marge d'erreur maximale ϵ entre $f(x)$ et y , dans le but d'éviter le surapprentissage. Ainsi, SVR calcule l'erreur lorsque la valeur absolue de la différence entre $f(x)$ et y est plus grande que ϵ . Durant l'apprentissage, si un point est prédit dans cette zone centrée sur y et de largeur ϵ alors le point est considéré comme correct. L'impact de la marge d'erreur est donc un paramètre crucial à régler pour que le modèle puisse avoir de bonnes performances en généralisation : une marge trop petite permet un meilleur apprentissage au risque de sur apprendre tandis qu'une marge trop grande peut avoir des performances médiocres en généralisation. L'implémentation de SVR a été réalisée avec l'implémentation de la librairie `scikit-learn`². Cet algorithme possède deux paramètres :

- ϵ , représentant la taille de la marge d'erreur.
- C , un paramètre de régularisation. La force de régularisation est inversement proportionnelle à C . Il doit être strictement positif.

5.3 Sélection de variables

Le processus d'apprentissage n'a pas nécessairement besoin de toutes les variables pour trouver un modèle expliquant le mieux le phénomène [22]. La sélection de variables est une méthode de prétraitement ayant pour but de réduire la dimensionnalité d'un problème. Ainsi, pouvoir choisir judicieusement les variables les plus pertinentes permet de créer des modèles pouvant être plus faciles à appréhender, plus rapides à apprendre, plus robustes et/ou plus explicables. En outre, si d'autres variables sont insérées, plus tard, pour améliorer le modèle existant, la méthode de sélection de variables pourra les conserver ou les rejeter automatiquement si elles ne contribuent pas à mieux expliquer l'observation se retrouvant non pertinente.

2. <https://scikit-learn.org/stable/>

L'objectif est de générer un modèle de plus faible complexité en excluant certaines des variables contribuant le moins à expliquer un phénomène tout en limitant la dégradation des performances du modèle. Dans le cas où plusieurs sorties sont à prédire, chaque sortie doit être considérée indépendamment, car il est possible que ce ne soit pas les mêmes variables qui contribuent le mieux à toutes les sorties. Dans ce cas, le modèle résultant peut être considéré comme une agrégation de plusieurs modèles allégés avec le nombre de variables préservées comme le nombre de variables uniques nécessaires à tous les modèles, c'est-à-dire qu'il y a une possibilité de redondance des variables. À partir d'un ensemble d'exemples contenant des entrées et des sorties, une méthode de sélection de variables utilise une métrique se basant sur les corrélations/reasons entre les variables pour établir un ordre de classement en fonction de leurs contributions pour expliquer les observations. Plusieurs méthodes de sélection de variables existent chacune ayant différentes stratégies telles que l'élimination récursive de variables (RFE) ou la méthode de filtrage (FM), la sélection itérative, ... Chaque méthode nécessite une métrique pour classer les variables par ordre d'importance. Ainsi, un estimateur est utilisé à cette fin. Tous les algorithmes pouvant générer cette métrique peuvent ainsi être utilisés, par exemple : l'algorithme de random forest (RF), la méthode des Support Vector Machine (SVM) avec un noyau linéaire, la méthode Least Absolute Shrinkage and Selection Operator (LASSO), ... L'un des avantages de ces méthodes est qu'elles ne nécessitent pas de connaissances a priori pour sélectionner les variables qui expliquent le mieux le problème. Toutefois, un inconvénient majeur de cette approche est la question de la pondération. En effet, le classement des variables est établi avant l'apprentissage en fonction des données disponibles (base d'apprentissage). Si une observation inconnue est en désaccord avec la base d'apprentissage, alors le classement peut être remis en question et cela pouvant induire le modèle à exclure une ou plusieurs variables pertinentes et de ce fait, pénaliser la généralisation du modèle résultant.

Dans cette thèse, 2 approches de sélection de variables ont été étudiées :

- La méthode de filtrage [79, 57] : Un estimateur est utilisé pour attribuer un score à chaque caractéristique et les classer par ordre d'importance. La sélection des meilleurs paramètres a priori se fait en fonction d'un

seuil. Toutes les variables avec un score supérieur ou égal au seuil sont préservées. La valeur du seuil est un hyper paramètre qui est fourni par l'utilisateur. Plus ce seuil est élevé, moins les caractéristiques ayant peu de contributions sont conservées.

- La méthode d'élimination récursive de variables [81] : À partir d'un ensemble de variables disponibles et d'un estimateur attribuant un score de contribution aux variables (par ex, les coefficients d'un modèle linéaire), l'objectif de la RFE est de récursivement éliminer un élément contribuant le moins à l'explication d'une sortie. Tout d'abord, l'estimateur est formé sur la base d'apprentissage et l'importance de chaque variable est ensuite établie. La variable ayant le score le plus faible est élaguée. Cette procédure est répétée sur le jeu de caractéristiques élagué jusqu'à ce que les K meilleures variables soient finalement atteintes. La valeur de l'hyper paramètre K est fournie par l'utilisateur.

On remarque que les stratégies de sélection de variables peuvent donc être très différentes. Par exemple, RFE utilise un processus itératif tandis que le processus de la méthode FM est réalisé en une seule itération. Lorsque le nombre de variables à estimer est important, la méthode RFE peut s'avérer être longue à calculer et donc prendre plus de temps que la méthode FM. Dans le cas où plusieurs sorties sont à prendre en compte, cette méthodologie ne peut pas être directement appliquée. En effet, les méthodes sont utilisées pour saisir les relations entre une ou plusieurs entrées et une sortie. Afin de remédier à ce problème, il est nécessaire de décomposer le problème avec k sorties en k problèmes avec une sortie. Chaque problème ayant les mêmes entrées mais une sortie différente. De cette façon, les méthodes de sélection de variables peuvent être appliquées pour chaque sortie. L'algorithme 13 et 14 décrivent les méthodes de sélection de variables utilisées dans cette thèse. La bibliothèque Scikit-learn a été utilisée pour implémenter ces deux approches utilisées dans la suite des travaux.

Synthèse Ce chapitre présente les problèmes d'apprentissage supervisé en se concentrant sur les problèmes de régression. Il se divise en trois parties. Dans un premier temps le principe de l'apprentissage est expliqué ainsi que les dif-

férentes étapes du processus d'apprentissage et de validation du modèle. La seconde partie présente différents algorithmes d'apprentissage supervisé à savoir les réseaux de neurones, les forêts d'arbres décisionnels et les Support Vector Machine. Le choix de ces derniers est motivé par la littérature récente, les utilisant pour apprendre des problèmes similaires à celui de la prédiction de la teneur hebdomadaire en eau. Ces algorithmes utilisent différentes stratégies afin d'apprendre à partir d'observations. La dernière partie introduit la sélection de variables avec ses avantages et limitations. Deux approches de sélection de variables sont ainsi présentées. Ces dernières permettent de sélectionner automatiquement les variables les plus pertinentes afin de réduire la dimensionnalité d'un problème d'apprentissage tout en limitant la dégradation des performances du modèle généré. Ces éléments posent la base des travaux développés dans le chapitre suivant.

Input : α : acceptance limit; X : training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$; F : set of features $\{f_1, \dots, f_n\}$; R : regression method.

Output : I : a set of most important features

```

begin
   $M \leftarrow \text{fit}(R, X)$ 
  foreach  $f \in F$  do
    if  $\text{ScoreImportance}_M(f) > \alpha$  then
       $I \leftarrow I \cup \{f\}$ 
    end
  end
  return  $I$ 
end

```

Algorithme 13 : méthode de filtrage (FM)

Bibliographie du présent chapitre

- [17] L BREIMAN et al. « Classification and Regression Trees ». In : (1984).
- [22] Girish CHANDRASHEKAR et Ferat SAHIN. « A survey on feature selection methods ». In : *Computers and Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, p. 16-28. ISSN : 0045-7906. DOI : <https://doi.org/10.1016/j.compeleceng.2013.11.024>. URL : <http://www.sciencedirect.com/science/article/pii/S0045790613003066>.

Input : K : number of features to select $\neq 0$; X : training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, R : regression method; F : a set of all features

Output : F : a set of most important features

Function $RFE(K, F, X)$

```

if  $K \neq 0$  then
  |  $M \leftarrow \text{fit}(R, X)$ 
  |  $f^* \leftarrow \underset{F}{\text{argmin}} \text{ScoreImportance}_M$ 
  |  $F \leftarrow F - \{f^*\}$ 
  | return  $RFE(K - 1, F, X)$ 
else
  | return  $F$ 
end

```

Algorithme 14 : Élimination récursive de variable (RFE).

- [35] Harris DRUCKER et al. « Support vector regression machines ». In : *Advances in neural information processing systems*. 1997, p. 155-161.
- [57] Isabelle GUYON et André ELISSEEFF. « An introduction to variable and feature selection ». In : *Journal of machine learning research* 3.Mar (2003), p. 1157-1182.
- [62] Tin Kam Ho. « Random decision forests ». In : *Proceedings of 3rd international conference on document analysis and recognition*. T. 1. IEEE. 1995, p. 278-282.
- [64] David HUNTER et al. « Selection of proper neural network sizes and architecturesA comparative study ». In : *IEEE Transactions on Industrial Informatics* 8.2 (2012), p. 228-240.
- [79] scikit LEARN. *Feature Selection*. 2020. URL : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html.
- [80] scikit LEARN. *random forest regressor*. 2020. URL : <https://scikit-learn.org/stable/modules/ensemble.html>.
- [81] scikit LEARN. *Recursive Feature Elimination*. 2020. URL : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.

-
- [102] NITISH SRIVASTAVA AND GEOFFREY HINTON AND ALEX KRIZHEVSKY AND ILYA SUTSKEVER AND RUSLAN SALAKHUTDINOV. « Dropout : A Simple Way to Prevent Neural Networks from Overfitting ». In : *Journal of Machine Learning Research* 15 (2014), p. 1929-1958. URL : <http://jmlr.org/papers/v15/srivastava14a.html>.
- [133] Vladimir N VAPNIK. « The nature of statistical learning theory ». In : (1995).
- [136] Paul WERBOS. « Beyond Regression : " New Tools for Prediction and Analysis in the Behavioral Sciences ». In : *Ph. D. dissertation, Harvard University* (1974).

Prédiction de phénomènes biologiques par apprentissage supervisé

Sommaire du présent chapitre

6.1 Apprentissage supervisé de la teneur en eau	126
6.1.1 Motivations	126
6.1.2 Travaux similaires	128
6.1.3 Méthodologie	128
6.1.4 Préparation des données	131
6.1.5 Analyse expérimentale	133
6.1.6 Synthèse	151

Introduction La problématique de la prédiction à la semaine de la teneur en eau dans le sol est reformulée comme un problème d'apprentissage. Ce chapitre propose une nouvelle méthodologie utilisant un algorithme d'apprentissage supervisé couplé avec un algorithme de sélection de variables pour la génération de modèles prédictifs. La constitution de la base d'observations est réalisée à partir de l'agrégation de données de capteurs issues de campagnes historiques

de culture sur plusieurs années et diverses conditions météorologiques. La validation de la méthode est effectuée sur 10 campagnes de culture avec l'utilisation d'une *leave-two-out cross-validation* pour obtenir des performances statistiquement significatives. La pertinence ainsi que les performances des algorithmes de sélection de variables sont aussi étudiées. Cette méthodologie permet d'apporter une réponse à une problématique actuelle [40] avec la mise au point d'un nouvel outil d'aide à la décision à destination des exploitants agricoles.

6.1 Apprentissage supervisé de la teneur en eau

6.1.1 Motivations

De nos jours, les exploitants agricoles ont besoin de capitaliser le plus d'informations possible afin d'avoir une meilleure compréhension de leur environnement et la possibilité d'être conseillés sur la conduite à tenir en matière d'irrigation sur leurs parcelles. Deux aspects essentiels sont à prendre en compte. Premièrement, la connaissance de la teneur actuelle en eau du sol est l'élément le plus critique dans ce processus [40]. Cette valeur est dynamique, ce qui signifie qu'un suivi quotidien est nécessaire pour analyser les tendances globales. L'un des indicateurs utilisés pour mener une campagne de culture est le potentiel hydrique ou *Soil Water Potential* (SWP). Cette métrique correspond à l'effort que la plante doit développer afin d'extraire l'eau du sol. Plus cette valeur est élevée, plus il est difficile pour la plante d'extraire l'eau et donc c'est à ce moment que le risque de stress hydrique apparaît. L'agriculteur peut utiliser directement cette métrique pour gérer ses cultures [72] ou bien la transformer en *Soil Water Content* (SWC) par l'intermédiaire de l'équation de Van Genuchten [132], sous réserve que la caractérisation du sol soit bien connue. L'acquisition du potentiel hydrique se fait par un tensiomètre (voir chapitre 1) implanté à la profondeur que l'on souhaite observer. Généralement, cette profondeur se situe au niveau des racines. À partir de cette valeur, des seuils peuvent être établis et caractérisent l'état hydrique du sol. À savoir : saturation, confort, vigilance et stress. Ces stades sont dépendants du type de sol. À partir de cette information, l'exploitant agricole a une connaissance des réserves actuelles d'eau présentes

dans ses champs. Deuxièmement, être capable de prédire les évolutions à court terme de cette métrique avec une vision hebdomadaire est un avantage majeur pour les exploitants agricoles, puisqu'ils peuvent ainsi planifier plus efficacement leurs irrigations en termes de quantité d'eau à apporter ou bien en nombre de tours d'eau à planifier (nombre d'irrigations à effectuer sur différentes parcelles). Les méthodologies proposées afin de faire des prévisions se basent sur les data précédemment collectées. Ces dernières sont utilisées comme entrées dans des modèles mathématiques. On peut citer par exemple : Johnson [69], STIC [19], AQUACROP [110], WEEDRIQ [111], water balance [6], Penman [63], ... Cette approche a prouvé sa pertinence et sa robustesse. Malheureusement, les modèles mathématiques sont source de simplification et d'introduction de biais. En outre, ils sont paramétriques, ce qui signifie qu'ils doivent être étalonnés pour pouvoir être utilisés de manière optimale [23, 124, 38]. Les phénomènes naturels ou biologiques étudiés comportent souvent des variables ayant des relations entre elles qui peuvent être non linéaires. Fort heureusement, les données collectées peuvent inclure des similitudes, des symétries naturelles, ... Ainsi, la conception d'un modèle apprenant directement à partir des observations et capable de trouver des relations entre les différentes variables et les valeurs à prédire devient dès lors pertinente. Une approche pouvant utiliser ces informations est l'apprentissage automatique (voir chapitre 5.1). La portée de cette contribution se limite à la culture de la pomme de terre pour laquelle l'irrigation joue un rôle crucial afin d'éviter le stress hydrique, c'est-à-dire un bilan négatif de l'approvisionnement en eau par rapport aux besoins de la plante. Ce phénomène peut mener à un effondrement des rendements [89]. L'utilisation de capteurs de qualité pouvant être un facteur limitant pour une grande majorité des agriculteurs en raison de leurs prix élevés et afin de démocratiser cette méthodologie, l'utilisation de jeux de capteurs à bas coût est proposée dans cette étude. L'utilisation de tensiomètres abordables est un bon compromis entre la précision des mesures et la possibilité d'être utilisés par un grand nombre d'agriculteurs. En outre, il devient possible à long terme de générer une base de données qui sera, bien qu'inexacte, mais pouvant couvrir différents types de sol, de cultures, de conditions climatiques, etc., ce qui est crucial pour l'apprentissage d'un modèle ayant des performances correctes.

6.1.2 Travaux similaires

Ces dernières années, les méthodes d'apprentissage automatique ont été appliquées avec succès sur de nombreuses applications du monde réel, et en particulier [24, 46, 75] ont démontré la pertinence de la modélisation de phénomènes biologiques complexes tels que l'évapotranspiration par transformation en un problème d'apprentissage résolu par des algorithmes d'apprentissage supervisé. [5] a mis en évidence que l'apprentissage automatique, dans la plupart des cas, surpasse les approches paramétriques pour la prévision de la biomasse et de l'humidité des sols à partir de données satellitaires. Dans le contexte de l'agriculture, [129] a appliqué avec succès l'apprentissage automatique pour prédire l'évapotranspiration hebdomadaire des vergers, et de ce fait les besoins en eau. Concernant la culture de la pomme de terre, divers sujets sont à ce jour étudiés et la littérature est bien fournie. Les méthodes utilisant les algorithmes d'apprentissage automatique ont été appliquées avec succès pour apporter des solutions sur des sujets tels que la prédiction du potentiel hydrique des feuilles [140], la modélisation du développement des racines [33], la croissance des tubercules [49], la prévision de l'évapotranspiration quotidienne [116, 125, 138], ... Dans la plupart des cas, les méthodes de résolution proposées sont élaborées à partir de parcelles ayant une taille modeste, et de ce fait, les réponses apportées sont données pour un climat, un contexte géographique et un type de sol particulier. Elles ne peuvent donc pas être appliquées directement à des fins de mise en production [47]. Ainsi, il est encore pertinent de développer des modèles simples d'utilisation pour la pratique de l'agriculture irriguée, mais aussi développés pour être utilisés sur de grandes surfaces agricoles [47].

6.1.3 Méthodologie

La question de la prévision de l'évapotranspiration est bien documentée. Cependant, le constat est différent pour la SWP. Cette métrique est assez peu étudiée et une proposition de méthodologie pour la prédiction hebdomadaire de cet indicateur, en utilisant une approche par apprentissage automatique alimenté par des capteurs, n'a pas été proposée. Pour la plupart des études citées précédemment impliquant des capteurs implantés dans le sol, une seule

profondeur est prise en compte. Cependant, une observation à plusieurs profondeurs permettrait d'observer les informations relatives aux échanges d'eau entre les différentes couches, par exemple la percolation, les remontées capillaires, ... Cette contribution utilise donc les mêmes algorithmes, à savoir les réseaux de neurones (NN), les forêts d'arbres décisionnels (RF) et les séparateurs à vastes marges (SVM) (voir section 5.1). Toutefois, les travaux portent sur la prédiction du SWP et non sur l'évapotranspiration et ceci sur une plus grande fenêtre temporelle, à savoir 7 jours par rapport à 1 jour dans la plupart des études mentionnées. De plus, la prédiction du SWP se fera sur trois profondeurs. Par rapport aux autres études, les capteurs utilisés ici ne sont pas supposés être nécessairement précis, l'objectif étant de pouvoir tester la pertinence de leur utilisation pour collecter des données et ainsi générer une base d'apprentissage pour ce phénomène dans des conditions de mise en production. En outre, le choix des variables est un point important à prendre en compte, lorsque l'on travaille sur des problèmes d'apprentissage [90, 25, 142]. Par conséquent, une étape de sélection automatique de variables est ajoutée à la méthodologie. En effet, il est possible que toutes les variables ne soient pas nécessairement utiles aux algorithmes pour expliquer les observations (section 5.3). Ainsi, les performances de deux approches de sélection de variables : la méthode d'élimination récursive de variables (RFE) [81, 54] et la méthode de filtrage (FM) sont étudiées. Dans cette étude, les prévisions météorologiques sont supposées correctes et la prédiction se limite à 7 jours en raison de la fiabilité des modèles météorologiques actuels. L'ensemble des données disponibles est ensuite divisé en deux parties :

- L'ensemble d'apprentissages constitué des données disponibles pour concevoir un modèle prédictif.
- L'ensemble de tests correspondant à une campagne de culture entière sur une culture qui n'est pas dans la base d'information. Cette méthode permet d'avoir une base de tests indépendante et sans introduction de biais. En effet, puisque les données proviennent de seulement quelques séries temporelles, utiliser un échantillonnage aléatoire peut conduire à tester les prévisions sur des points proches de ceux déjà appris.

L'étape de validation du modèle prédictif est effectuée par une méthode de

validation croisée : deux cultures sont retirées de l'ensemble des données disponibles pour servir d'ensemble de tests et les 8 autres cultures correspondent à l'ensemble d'apprentissages. L'opération est ainsi répétée pour chaque paire de champs possibles. Ce choix de conception est un compromis entre la faible quantité de données d'apprentissage actuellement disponibles et la précision statistique nécessaire pour pouvoir interpréter les résultats. Ainsi, le nombre de tests issus de la validation croisée est $N = 45$ où $N = (\text{nombreDeChamps} * (\text{NombreDeChamps} - 1))/2$ car le nombre de cultures disponibles est de 10. De cette manière, l'échantillon de test devient suffisamment important pour être statistiquement significatif. Ce problème n'est pas simple, car il y a 21 sorties à prédire à savoir 3 profondeurs à estimer sur 7 jours. Ainsi, deux séries d'expériences sont proposées :

- Pour la première expérience, l'impact de la méthode de sélection automatique de variables dans le processus de création de modèles ainsi que leurs performances sont étudiés. Puisque ces méthodes ont besoin d'un estimateur pour générer une métrique de contribution (section 5.3), l'impact de ce dernier est aussi observé. Enfin, l'importance de la valeur de l'hyper paramètre de la méthode est étudiée.
- Pour la deuxième expérience, une comparaison des meilleurs modèles issus de chaque technique (avec ou sans méthode de sélection de variables) pour chaque algorithme d'apprentissage est effectuée. La performance globale ainsi que l'évolution de l'erreur en fonction du i -ème jour à prédire en fonction de plusieurs métriques est analysée.

Les précipitations et les irrigations ont un fort impact sur la teneur en eau du sol entre les profondeurs de 0 et 20 cm, correspondant à la zone située au-dessus du premier capteur. Il est donc difficile de prendre en compte l'évolution de la teneur en eau dans cette zone. L'unité utilisée par les capteurs est le Kpa et sa plage de valeurs se situe entre 0 kpa (saturation en eau) et 200 kpa (sécheresse). À noter que les valeurs entre 0 et 20 kpa (sol saturé) et au-dessus de 150 kpa (sol très sec), ne peuvent pas être mesurées avec précision par les capteurs.

Dans ce travail, le réseau de neurones proposé est un réseau de type *feed forward* développé grâce à la bibliothèque Keras. Le réseau est composé d'une couche cachée de 600 neurones utilisant une fonction d'activation sigmoïde et

une couche de sortie de taille 21 (la raison de la taille de la couche de sortie est donnée dans la section 6.1.4) utilisant une fonction d'activation linéaire. En outre, un *dropout* à 0,2 est utilisé afin d'éviter le surapprentissage. La topologie ainsi que les paramètres proposés sont le résultat d'une étude empirique. Est présentée ici celle qui a obtenu les meilleurs résultats. Pour Random Forest, après une étude empirique, le nombre d'arbres décisionnels a été fixé à 500. Pour le SVR, une version avec un noyau "linéaire" est utilisée. D'autres noyaux ont été testés, mais ce dernier donne les meilleurs résultats. La valeur du paramètre C est de 5 et la valeur de ϵ est de 10^{-1} .

6.1.4 Préparation des données

Le modèle est construit en utilisant les data provenant de campagnes historiques de culture de pomme de terre. Les champs sont proches les uns des autres et sont localisés dans le nord de la France sur la façade maritime. Le climat y est océanique et caractérisé par une météo venteuse et changeante ainsi que des températures estivales modérées, un taux d'humidité important et des précipitations fréquentes, mais peu intenses. D'après les experts, les champs ont la même texture de sol (section 1 de la partie introduction). 3 profondeurs sont considérées : 20, 30 et 40 cm . Ces profondeurs correspondent à celles des racines de pomme de terre dans la région des Haut-de-France et donc où la prédiction de la teneur en eau est souhaitée. Effectivement, c'est à ces profondeurs que la plante peut puiser l'eau dont elle a besoin. Les capteurs étant abordables trois points de mesures par profondeur sont utilisés pour un total de 9 mesures. Puisque les capteurs ne sont pas précis, la médiane des 3 points de mesures est un compromis acceptable. Les données météo sont fournies par des stations météo situées à moins de 5 km des champs. Les campagnes de mesures couvrent une période de 3 ans (2016, 2018 et 2019) sur les mois d'avril à septembre. L'année 2016 est caractérisée par de fortes précipitations et 2019 par des déficits de précipitations. Sur ces 3 ans, 3 à 4 champs ont été suivis annuellement, ce qui amène la base d'observations à un total de 10 campagnes de culture. Chaque observation correspond à un jour d'une campagne et est considérée comme un point indépendant. La période de culture pour les pommes de

terre est très courte (environ 90 jours). À cela il faut enlever les jours où les données capteurs ne sont pas exploitables. Il y a différentes explications : la période d'étalonnage du capteur, des problèmes techniques, déplacement du capteur pour permettre le passage d'engins agricoles, ... Après ce travail de préparation de données, le nombre de points de mesures (observations) est d'environ 750. Les variables fournies par les agronomes sont présentées dans la table 6.1. Pour

TABLEAU 6.1 – Les 5 variables utilisées dans cette contribution.

Variable	Description
Water_Pressure	Valeur du potentiel hydrique appliqué au capteur en <i>Kpa</i> (3 profondeurs étudiées)
Mean_Temperature	Température moyenne en <i>Celcius</i> de la journée
Rainfall	Apport en eau (irrigation + pluie) en <i>mm</i> de la journée.
AGE	Age de la plante exprimé en <i>Celcius</i> par jour résultant de la somme des degrés au dessus d'un seuil (6 degré <i>Celcius</i> pour la pomme de terre)
Dry	Indique si le sol est sec, c'est-à-dire si le potentiel hydrique est supérieur à 150 <i>Kpa</i>

chaque exemple, la météo des 3 derniers jours et des prévisions pour les 7 prochains jours sont disponibles. Le nombre total de variables est ainsi de 36. La base de données inclut la valeur journalière moyenne des tensiomètres à 3 profondeurs dans différents champs, de la période de semis à la sénescence, ainsi que les apports en eau (irrigation + précipitations) et la température de l'air. Afin d'éviter les effets d'échelle, les données d'entrées sont normalisées. L'équation 6.1 permet de normaliser les données brutes.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (6.1)$$

La figure 6.1 présente la matrice de corrélation des différentes variables à disposition. Sont disponibles les 5 variables précédemment énoncées parmi la gamme de temps étudiée :

— Potentiel hydrique : P1, P2, P3 et P_day_* pour tous les jours précédents

et à venir.

- Mean temperature : T_mean_* pour tous les jours précédents et à venir.
- Précipitation : Rainfall_* pour tous les jours précédents et à venir.
- Age : Age_day_* pour tous les jours précédents et à venir
- Dry : Dry_P1, Dry_P3, Dry_P3 Valeur binaire indiquant si la valeur est au-delà du seuil d'imprécision.

Parmi ces variables, certaines corrélations semblent émerger : la pluie et les irrigations prévisionnelles (les apports) ont une corrélation négative avec le potentiel hydrique (P1, P2 et P3), ce qui est attendu puisque la pluie et les irrigations sont, dans cette étude, les seuls moyens de faire décroître le potentiel hydrique. Cependant, il est à noter que ces variables ont une plus forte influence sur la première couche (P1) que les autres (P2 et P3). L'âge et la température ont un impact modéré.

6.1.5 Analyse expérimentale

Comme détaillé dans la section 6.1.3, les expériences ont pour but de montrer la pertinence d'une part de l'utilisation d'algorithmes d'apprentissage dans la résolution de problème de prédiction de phénomène biologique et d'autre part de l'efficacité des méthodes de sélection de variables dans le processus de conception de modèle. Comme expliqué dans la section 6.1.3, les données sont divisées en 2 bases indépendantes. L'évaluation de la précision des modèles est effectuée avec plusieurs métriques, à savoir la *Mean-Absolute-Error* (MAE), la *Root Mean Squared Error* (RMSE) et le coefficient de corrélation (r^2) défini par les équations 6.2, 6.3, 6.4

$$\text{MAE}_{depth(j)} = \frac{1}{n} \sum_{i=1}^n |y_{i,j} - \hat{y}_{i,j}| \quad (6.2)$$

$$\text{RMSE}_{depth(j)} = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_{i,j} - \hat{y}_{i,j})^2} \quad (6.3)$$

$$r^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (6.4)$$

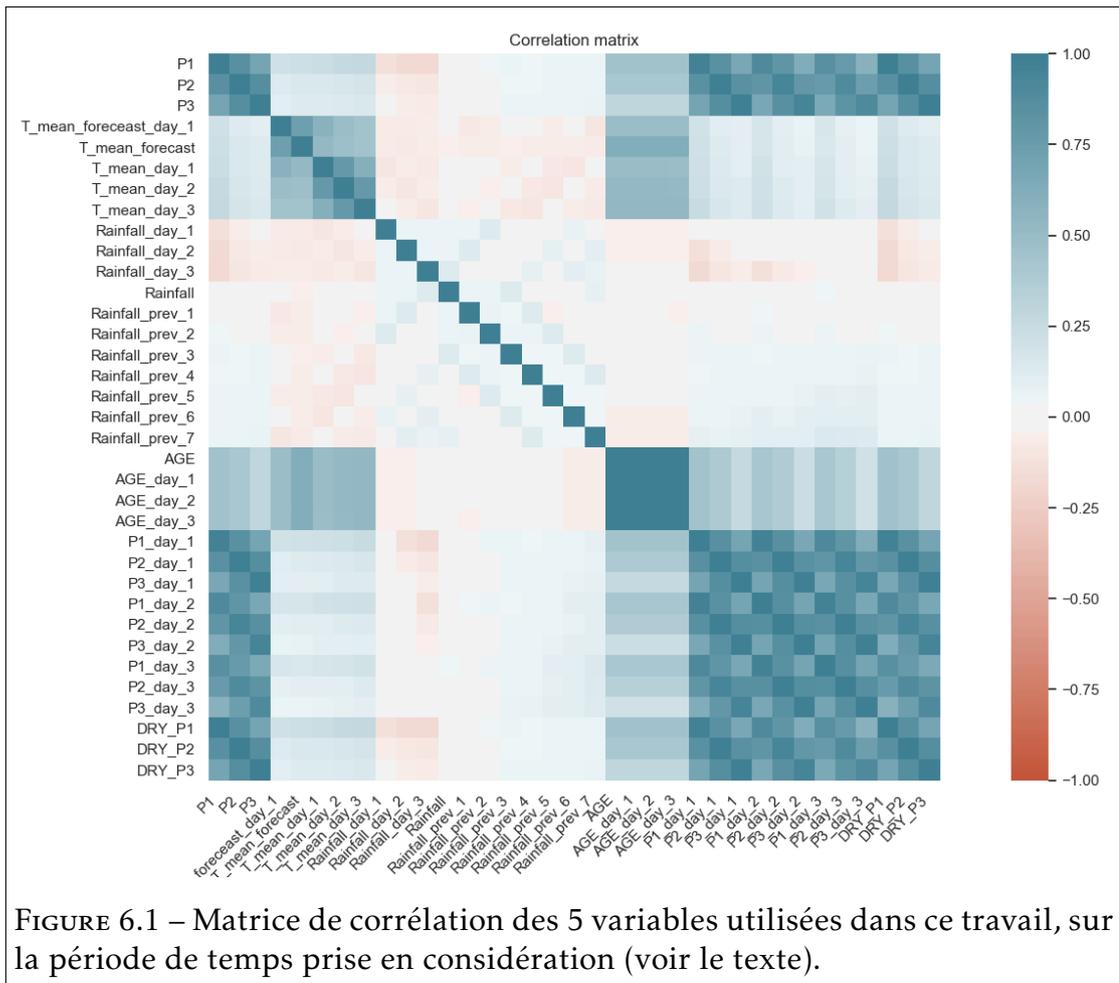


FIGURE 6.1 – Matrice de corrélation des 5 variables utilisées dans ce travail, sur la période de temps prise en considération (voir le texte).

où n est le nombre d'exemples, $y_{i,j}$ les valeurs réelles de tensiomètre pour le jour i à la profondeur j et $\hat{y}_{i,j}$ la prédiction faite par le modèle pour le jour i à la profondeur j . Les erreurs sont ensuite moyennées sur les 3 profondeurs :

$$\text{MAE} = \frac{1}{3} \sum_{j=1}^3 \text{MAE}_{depth(j)}$$

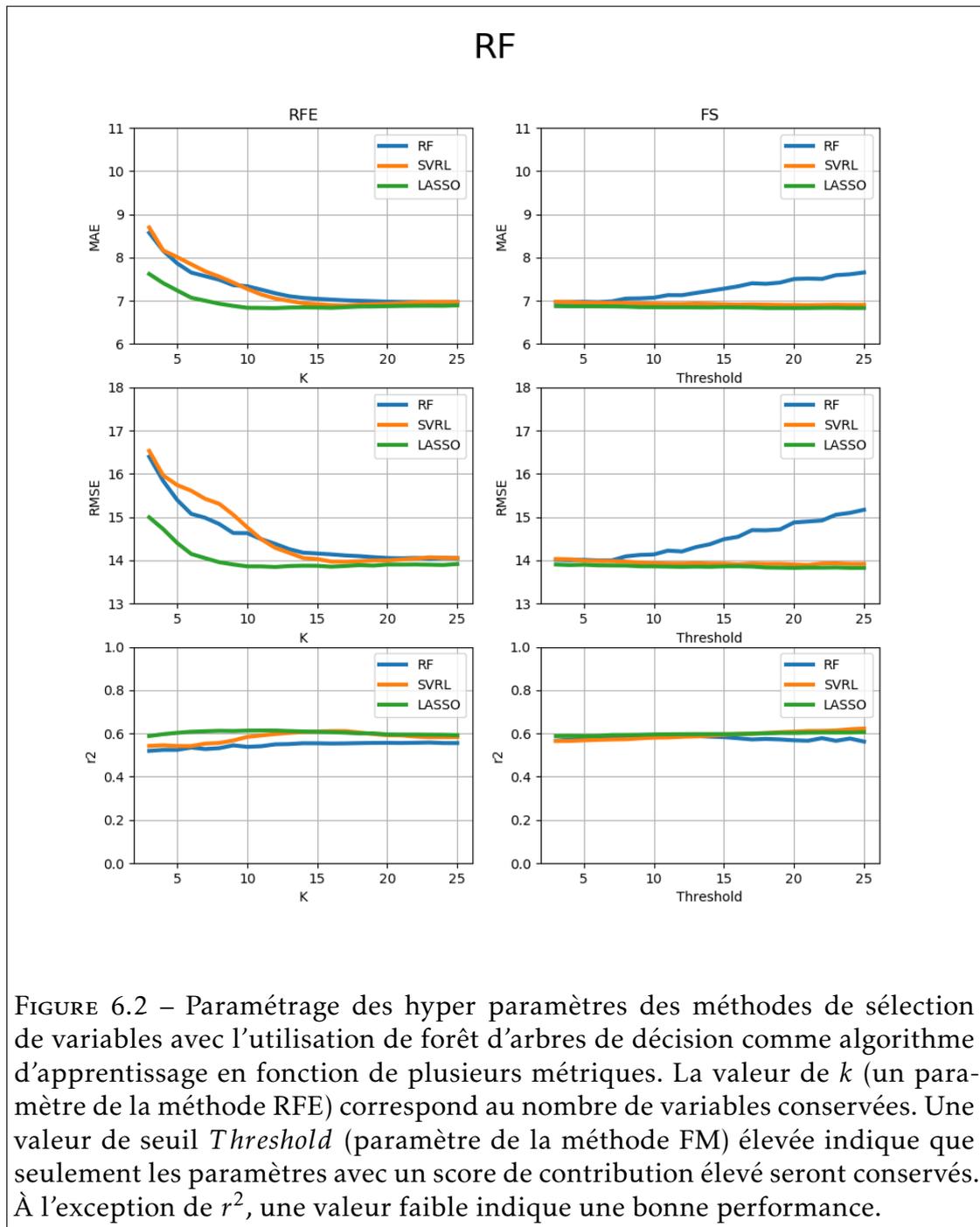
$$\text{RMSE} = \frac{1}{3} \sum_{j=1}^3 \text{RMSE}_{depth(j)}$$

$$r^2 = \frac{1}{3} \sum_{j=1}^3 r_{depth(j)}^2$$

Le but est, en effet, de minimiser une erreur globale reflétant la moyenne des erreurs à chaque profondeur.

Pour la première expérience, les figures 6.2, 6.3, et 6.4 présentent l'influence de l'hyper paramètre des méthodes de sélection de variables (*Threshold* pour la méthode de filtrage et k , le nombre de variables à conserver pour la méthode d'élimination récursive de variables) en fonction de plusieurs métriques définies par les équations 6.1.5 6.1.5 et 6.1.5 après qu'une *leave-two-out cross-validation* soit réalisée. Comme expliqué dans la section 6.1.3 utiliser une *leave-two-out cross validation* permet d'avoir des performances statistiquement significatives et ainsi de comparer les moyennes.

Concernant la méthode FM, en abscisses, une forte valeur de *Threshold* signifie qu'un petit nombre de variables est conservé. À l'opposé, pour la RFE, une haute valeur indique qu'un large nombre de variables est conservé. Sur ces graphiques sont représentés plusieurs estimateurs utilisés pour sélectionner les variables : l'algorithme SVR avec un noyau linéaire (SVRL), la méthode Least Absolute Shrinkage and Selection Operator (LASSO) avec une validation croisée et l'algorithme Random Forest Regressor (RF). En ce qui concerne les résultats, les performances des méthodes de sélection de variables sont assez similaires entre les algorithmes d'apprentissage. La diminution rapide de l'erreur (MAE et RMSE) lorsque l'hyper paramètre est faible c'est-à-dire jusqu'à



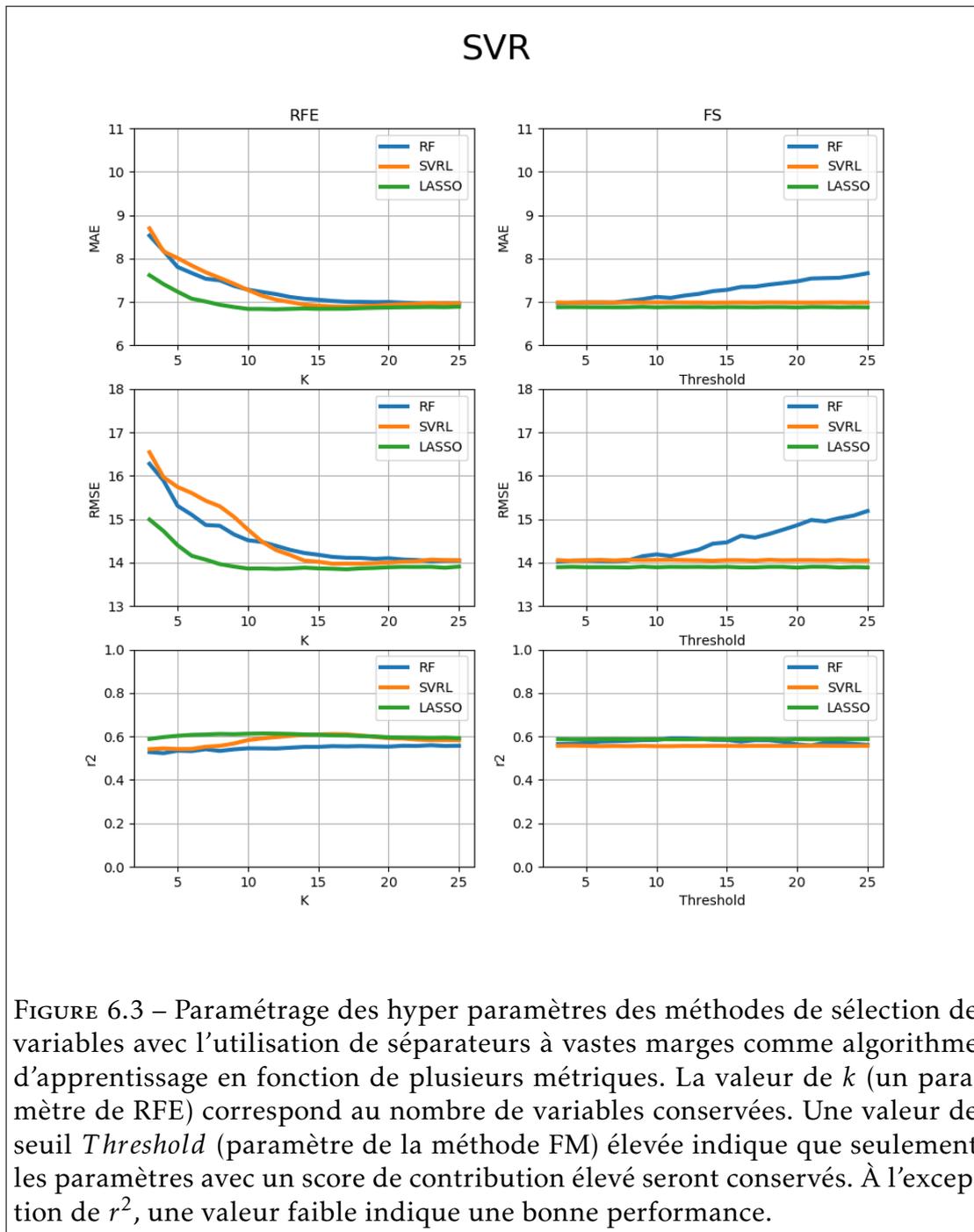
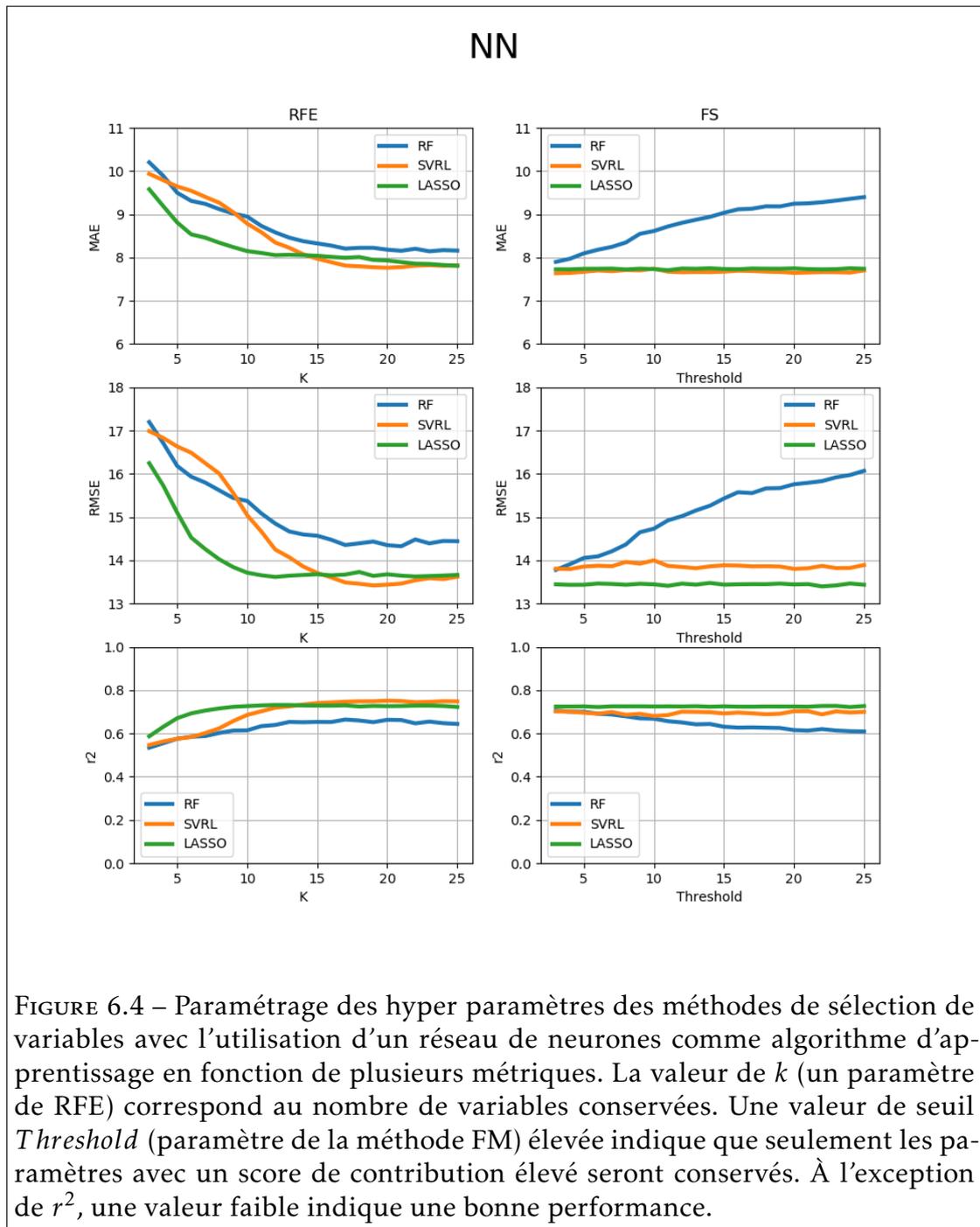


FIGURE 6.3 – Paramétrage des hyper paramètres des méthodes de sélection de variables avec l’utilisation de séparateurs à vastes marges comme algorithme d’apprentissage en fonction de plusieurs métriques. La valeur de k (un paramètre de RFE) correspond au nombre de variables conservées. Une valeur de seuil *Threshold* (paramètre de la méthode FM) élevée indique que seulement les paramètres avec un score de contribution élevé seront conservés. À l’exception de r^2 , une valeur faible indique une bonne performance.



10 pour la méthode LASSO et 15 pour l'algorithme RF et l'algorithme SVRL indique qu'à partir de cette valeur les modèles ont suffisamment de variables pour pouvoir apprendre correctement le phénomène. D'après les résultats, la méthode LASSO est capable de sélectionner les paramètres les plus pertinents lorsque l'hyper paramètre est faible, et ce quel que soit l'algorithme d'apprentissage. Concernant l'algorithme RF avec la sélection de variables, d'après les résultats en utilisant la méthode RFE, plus le paramètre k est élevé, meilleure est la prédiction. En effet, la MAE et la RMSE diminuent toutes deux alors que le coefficient de corrélation r^2 augmente légèrement, ce qui est attendu. En ce qui concerne l'estimateur, les performances des algorithmes RF et SVRL sont équivalentes et la méthode LASSO avec validation croisée a une meilleure dynamique. La méthode LASSO trouve un ensemble de paramètres pertinents avec $k = 10$ ce qui est bien mieux que les algorithmes RF et SVRL quelle que soit la métrique à cette valeur de k . Au-delà de cette valeur, les performances en MAE et en RMSE convergent vers celles des autres estimateurs, mais le score r^2 reste légèrement supérieur au SVRL et supérieur au RF. Cela suggère que la méthode LASSO a réussi à sélectionner des variables plus pertinentes. Le modèle qui en résulte présente des écarts extrêmes moindres (car la RMSE est plus faible) que les autres méthodes, ce qui devrait également expliquer le coefficient de corrélation plus élevé. Comme expliqué dans la section 5.3, étant donné que plusieurs sorties doivent être prises en compte, il y a k variables sélectionnées par sortie. Mais comme ces sorties ne sont pas forcément identiques, la combinaison de toutes ces sorties forme un ensemble plus vaste. Par exemple, pour l'algorithme RF utilisant une méthode RFE avec la méthode LASSO comme estimateur et avec l'hyper paramètre $k = 10$, les modèles résultants utilisent 10 variables par sortie, mais l'ensemble des uniques variables est de 30, soit environ 25% de moins que le modèle avec toutes les variables. De son côté, pour la méthode FM, la stratégie est différente. Plus le seuil est élevé, plus le nombre de variables conservé est faible. Ceux dont le score d'importance est supérieur ou égal au seuil sont retenus. Il n'y a donc aucune garantie quant à un nombre constant de variables par sortie. Ici, la méthode LASSO et l'algorithme SVRL ont des performances similaires tandis que l'algorithme RF montre une dégradation à partir de la valeur $Threshold = 7$. Cela suggère que l'algorithme RF note moins perti-

nemment les variables. Ici, le meilleur réglage de l'hyper paramètre est atteint pour la méthode LASSO et l'algorithme SVRL lorsque $Threshold = 16$ pour ces valeurs, le nombre total de variables uniques est alors de 43 (avec jusqu'à 33 variables par sortie). Les performances de l'algorithme SVR étant identiques à celui de l'algorithme RF, les conclusions sont donc les mêmes. Le nombre de variables par sortie selon la méthode de sélection et les estimateurs sont les mêmes que décrit ci-dessus. Enfin pour l'algorithme NN, en utilisant la méthode RFE, d'après les performances, l'erreur absolue (MAE) est plus élevée que pour les algorithmes RF ou SVR. Mais la RMSE et le r^2 sont meilleurs, avec des valeurs respectives de 13,7 et 0,78 obtenues avec l'utilisation de l'algorithme SVRL comme estimateur et une valeur de $K = 20$. Les modèles résultants utilisent 20 variables par sortie et l'ensemble de variables uniques est de 33. Ces résultats indiquent aussi qu'en moyenne, l'algorithme NN présente moins d'erreurs extrêmes. En ce qui concerne les estimateurs, l'algorithme RF a des performances moindres, quelle que soit la méthode de sélection de variables. Cela fait écho à ce qui a été dit précédemment. En ce qui concerne la méthode FM, à l'exception de l'algorithme RF, l'algorithme SVRL et la méthode LASSO ont également des performances similaires. L'impact du seuil et de l'estimateur étant faible, si l'on considère $threshold = 25$ (la valeur de paramétrage maximum considéré), alors le nombre de variables par sortie est au maximum de 29 pour un total de 35 variables uniques.

Discussions En ce qui concerne la méthode de sélection de variables, en premier lieu, reformuler un problème avec plusieurs sorties en plusieurs problèmes avec une sortie en utilisant une méthode de sélection de variables permet de concevoir automatiquement des modèles ayant les variables les plus pertinentes pour chaque sortie et donc plus spécialisées. L'estimateur utilisé pour attribuer un score d'importance pour les variables a un impact significatif sur le modèle résultant en termes de performances, tandis que la méthode de sélection de variables a un impact sur le nombre de variables retenues.

Pour les algorithmes RF et SVR, utiliser RFE comme méthode de sélection de variables combinée avec la méthode LASSO avec validation croisée comme estimateur offre les modèles les plus intéressants, a priori, puisqu'elle présente

une faible erreur tout en conservant un coefficient de corrélation plus élevé que les autres méthodes. Enfin, pour l'algorithme NN, utiliser la méthode RFE avec l'algorithme SVRL comme estimateur donne les meilleures performances (r^2 et RMSE).

Comparaison de performances La Fig. 5 montre les boxplots mis en couleurs selon l'algorithme d'apprentissage utilisé (RF, SVR et NN) et regroupés selon la méthode de sélection de variables. Chaque boxplot d'un groupe correspond respectivement à :

- Sans méthode de sélection, c'est-à-dire en utilisant toutes les variables disponibles (NONE).
- Méthode d'élimination récursive des variables (RFE).
- Méthode de filtrage (FM).

Le meilleur paramétrage de chaque méthode de sélection de variables a été utilisé. Pour les algorithmes RF et SVR en utilisant la méthode RFE, le paramètre k est de 10 et en utilisant la méthode FM, le paramètre *Threshold* est de 16. Ces deux méthodes utilisent la méthode LASSO comme estimateur. Pour l'algorithme NN en utilisant la méthode RFE, le paramètre k est de 20 et en utilisant la méthode FM, le paramètre *Threshold* est de 25. L'estimateur utilisé pour la méthode RFE et la méthode FM est l'algorithme SVRL. Premièrement, concernant les méthodes d'apprentissage (avec ou sans sélection de variables), à part pour SVR, les moyennes sont assez similaires, quelles que soient les métriques. Pour la MAE et la RMSE, les erreurs maximales sont plutôt faibles (ce qui est particulièrement le cas pour l'algorithme NN) lorsque la sélection de variables est utilisée. En revanche, les valeurs minimales de r^2 sont plus élevées. C'est un point intéressant, car il montre que les modèles de plus faible dimension ont des performances similaires au modèle avec toutes les variables (NONE). En outre, les performances du SVR sont nettement améliorées lorsque la sélection de variables est utilisée. Si l'on compare les algorithmes d'apprentissage entre eux, lorsqu'aucune sélection de variables n'est utilisée (NONE), RF a la meilleure performance moyenne. Cependant, l'algorithme NN a une valeur minimale de r^2 plus élevée. Lorsque la sélection de variables est utilisée, SVR a des performances similaires à celles de l'algorithme RF. L'algorithme NN de son côté a une

erreur moyenne certes plus élevée, mais des valeurs maximales plus faibles en MAE et RMSE. La dispersion est également plus faible avec moins de valeurs extrêmes hautes. Pour r^2 , le minimum est aussi plus élevé pour l'algorithme NN que pour l'algorithme RF. Les tableaux 6.3, 6.4 et 6.2 détaillent respectivement les performances des figures 6.2, 6.4 et 6.3. Les résultats confirment cette analyse.

Si les performances globales sont détaillées pour faire apparaître l'évolution de l'erreur en fonction du nombre de jours à prédire pour chaque profondeur (P1, P2 et P3), les figures 6.6, 6.7 et 6.8 montrent cette évolution en fonction des mêmes métriques (MAE, RMSE et r^2). Ici, les mêmes hyper paramètres sont conservés. Ainsi les graphiques montrent un autre point de vue des performances. Pour les algorithmes RF, SVR et NN, la prédiction à 1 jour est correcte au vu des bonnes performances obtenues avec un score r^2 supérieur à 0,92, quelle que soit la méthode de sélection utilisée. Cette précision est équivalente à la performance d'autres études de la littérature sur l'évapotranspiration. Lorsque le nombre de jours à prédire augmente, la performance reste correcte jusqu'à 4 jours puis se détériore ensuite au-delà. À 7 jours, la MAE et la RMSE augmentent fortement, mais restent correctes. Cependant, le score r^2 s'effondre, ce qui indique que la prévision semble plus incertaine. Si l'on se concentre sur les différentes profondeurs, on constate que l'évolution de l'erreur n'est pas la même selon la profondeur. À 20 cm, l'erreur augmente plus rapidement que pour 30 et 40 cm. Cela peut s'expliquer par une dynamique plus importante à 20 cm qu'à 30 et 40 cm. Une autre explication possible est que la base d'apprentissage ainsi que la base de tests ont peu d'observations avec des valeurs élevées de tensiomètre (près de 200 Kpa) à ces profondeurs.

Finalement, en ce qui concerne la méthode de sélection de variables, les différents algorithmes ont des performances similaires lorsque la sélection de variables est utilisée, mais différentes du modèle avec toutes les variables (NONE). Ce dernier apprend mieux la première profondeur (P1) que les deux autres (P2 et P3), quel que soit le nombre de jours à prédire, contrairement aux modèles avec sélection de variables apprenant les différentes profondeurs de manière plus égale. Les scores r^2 sont aussi nettement moins dispersés avec la méthode de sélection de variables (et meilleure pour la P2 et P3) qu'avec le modèle

NONE. Ce comportement est confirmé avec les autres métriques. Ainsi, si l'on prend en compte toutes les profondeurs, jusqu'à 5 jours, il est préférable d'utiliser le modèle avec une méthode de sélection de variables (RFE ou FM). Au-delà, si l'objectif est d'obtenir une bonne estimation de la première profondeur (P1), il est alors plus pertinent d'utiliser le modèle NONE. Ces résultats sont intéressants, car ils mettent en évidence le fait que le modèle global (NONE) peut apprendre plus efficacement une 1 profondeur (plutôt que sur les autres) tandis que la sélection de variables est capable de sélectionner efficacement les variables les plus pertinentes et avoir des modèles avec des performances à court terme meilleures pour chaque profondeur.

Discussions Les résultats sont encourageants, car ils montrent que les méthodes de sélection de variables sont pertinentes pour générer des modèles performants. De plus, le processus de conception automatique de modèles de plus faibles dimensions est plus pertinent au vu des résultats encourageants puisque les différences sont (au pire) identiques qu'avec l'inclusion de toutes les variables. De plus, la dispersion est aussi plus faible. Les modèles montrent aussi des différences de performances entre les profondeurs réduites (quelle que soit la métrique).

TABLEAU 6.2 – Détails des performances à l'issue d'une leave-two-out cross validation avec un réseau de neurones selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.

	NN	NN_FM	NN_RFE
Moy.	7.94	8.14	7.73
Min.	1.15	1.15	1.30
Max.	27.64	27.64	27.13
25 %	4.32	4.11	4.54
50 %	6.65	7.02	6.40
75 %	10.01	10.18	9.78

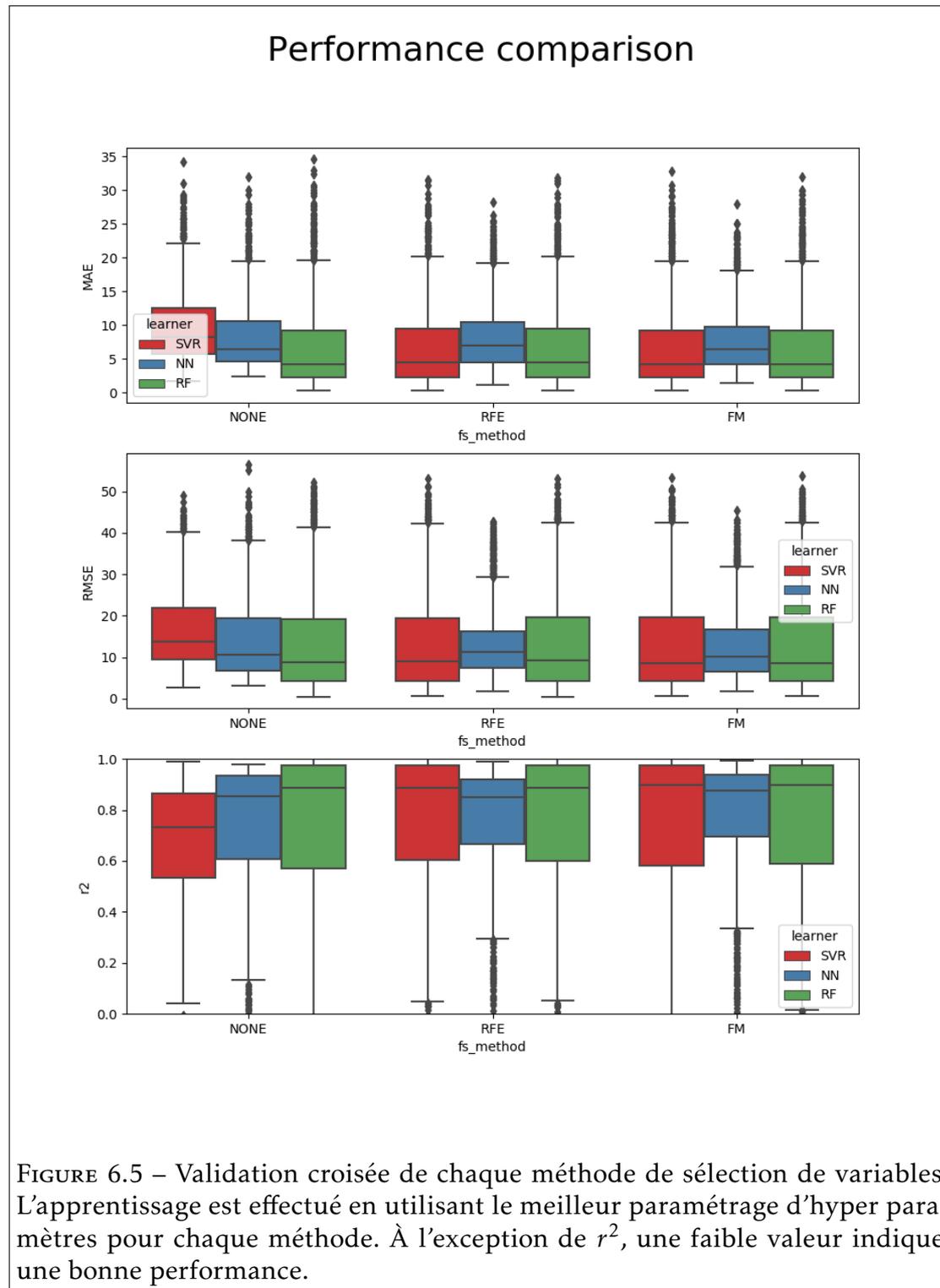
Outil d'Aide à la Décision Puisque le réseau de neurones est la méthode d'apprentissage ayant le moins d'erreurs extrêmes (borne haute), cette méthode est

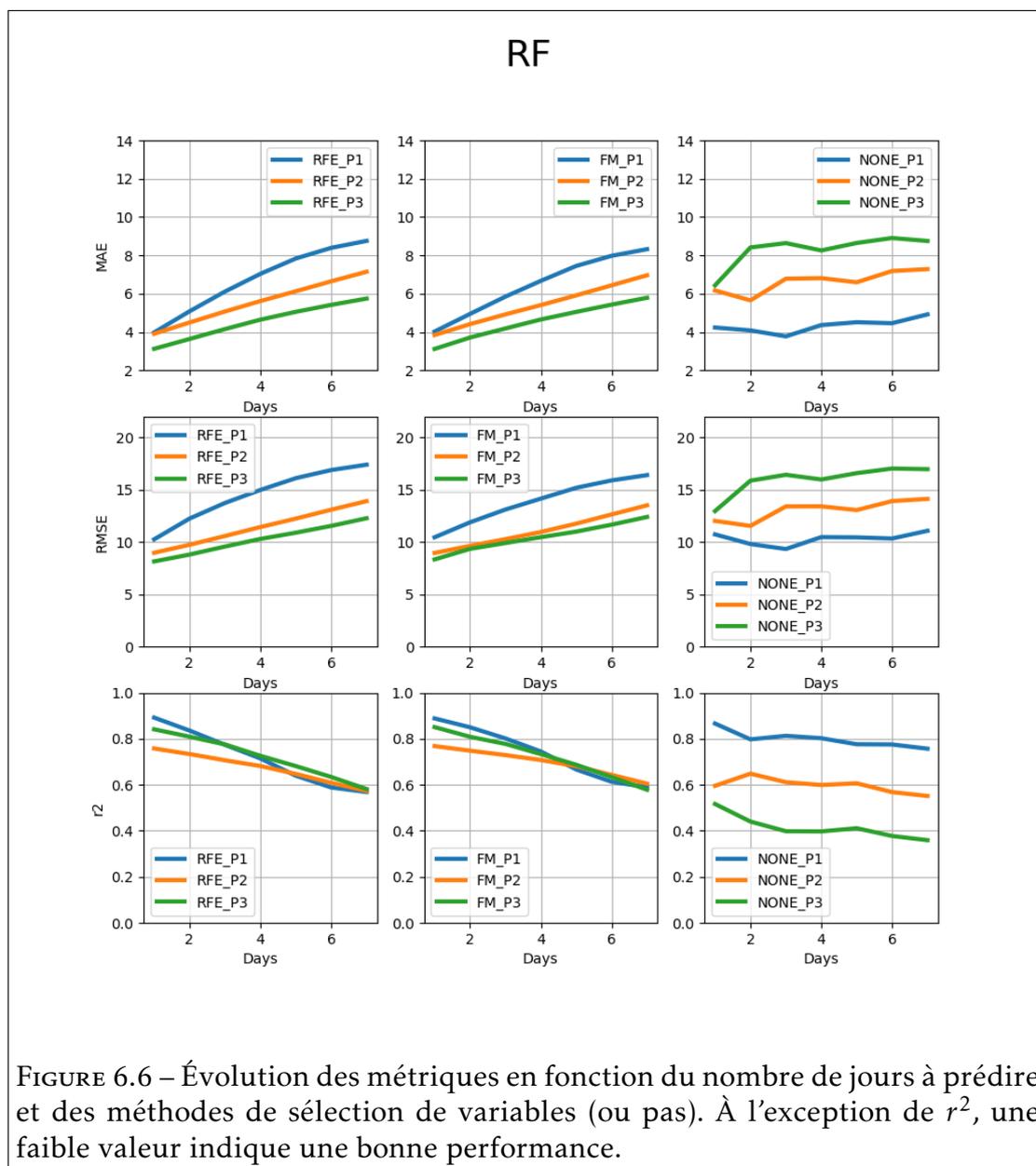
TABLEAU 6.3 – Détails des performances à l’issue d’une leave-two-out cross validation avec une forêt d’arbres décisionnels selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.

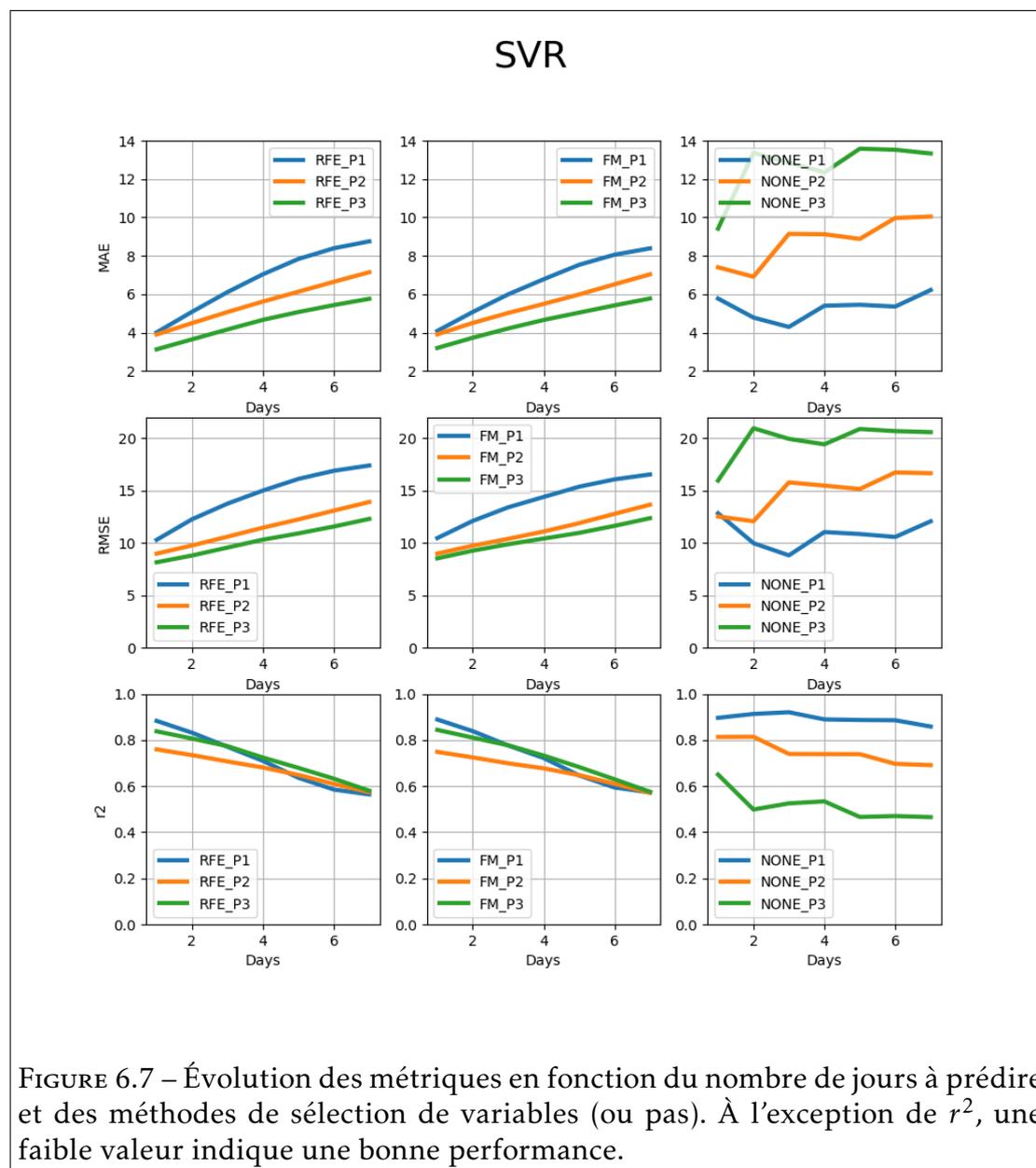
	RF	RF_FM	RF_RFE
Moy.	6.84	6.86	6.84
Min.	0.28	0.29	0.28
Max.	32.15	32.04	32.15
25%	2.25	2.28	2.24
50%	4.35	4.29	4.39
75%	9.18	9.08	9.36

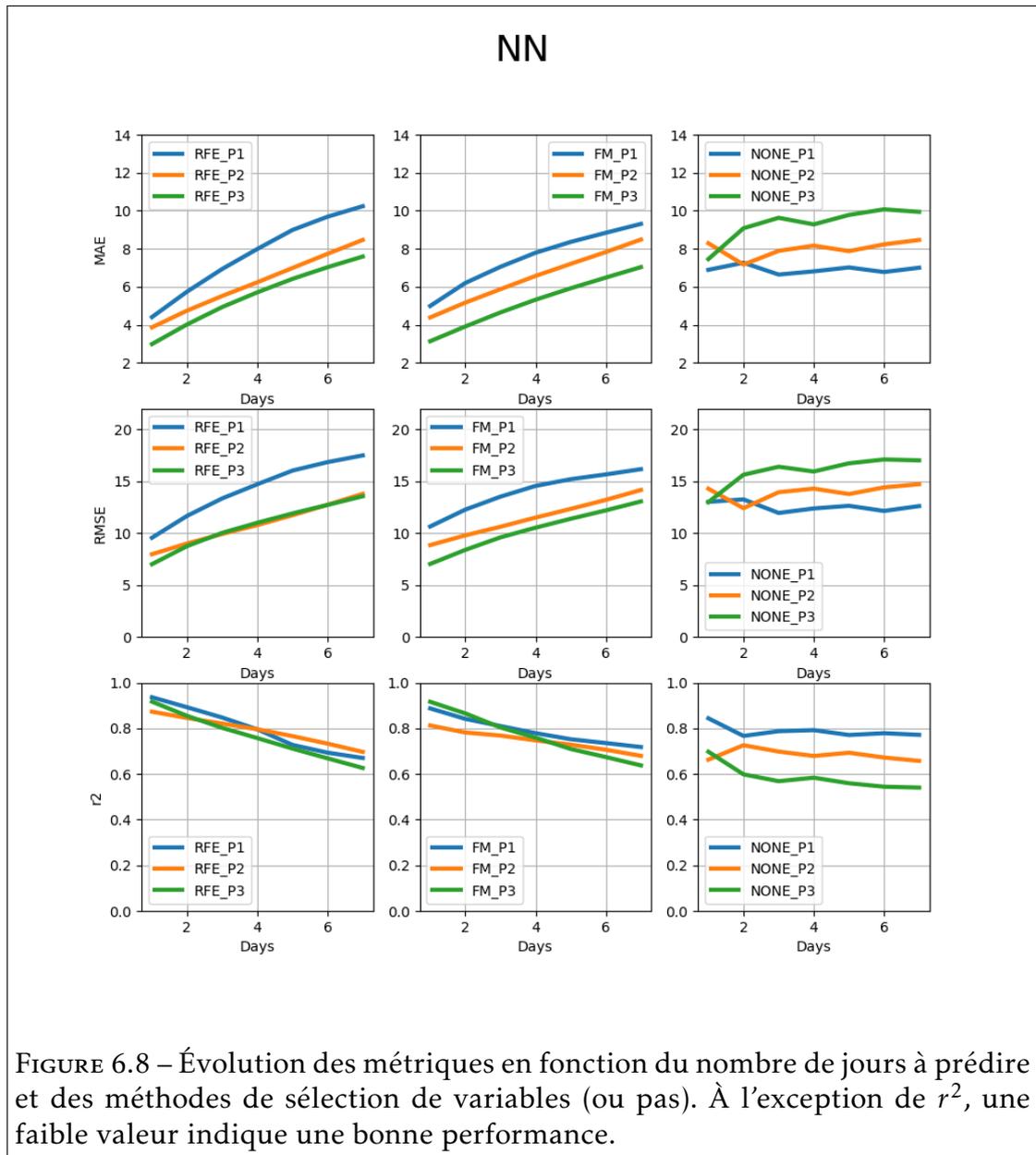
TABLEAU 6.4 – Détails des performances à l’issue d’une leave-two-out cross validation avec un séparateur à vastes marges, selon différentes méthodes de sélection de variables. Une valeur faible indique une meilleure performance.

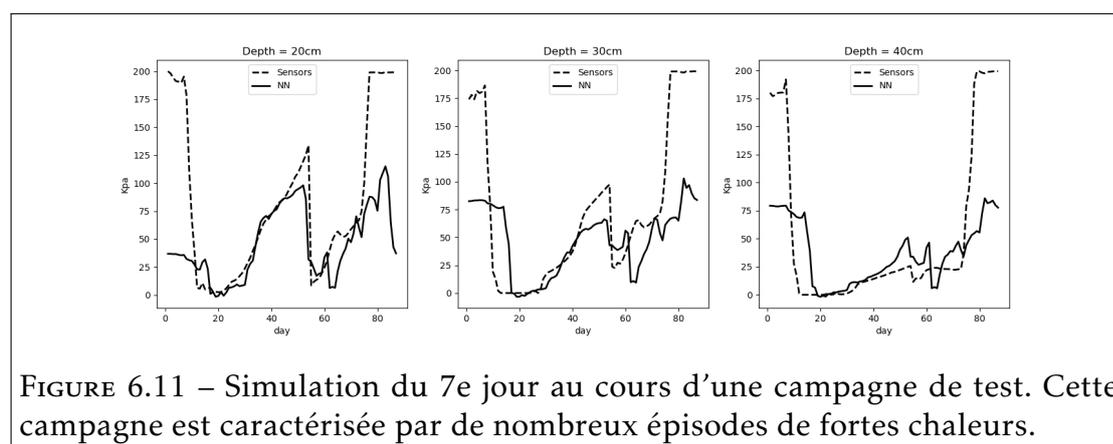
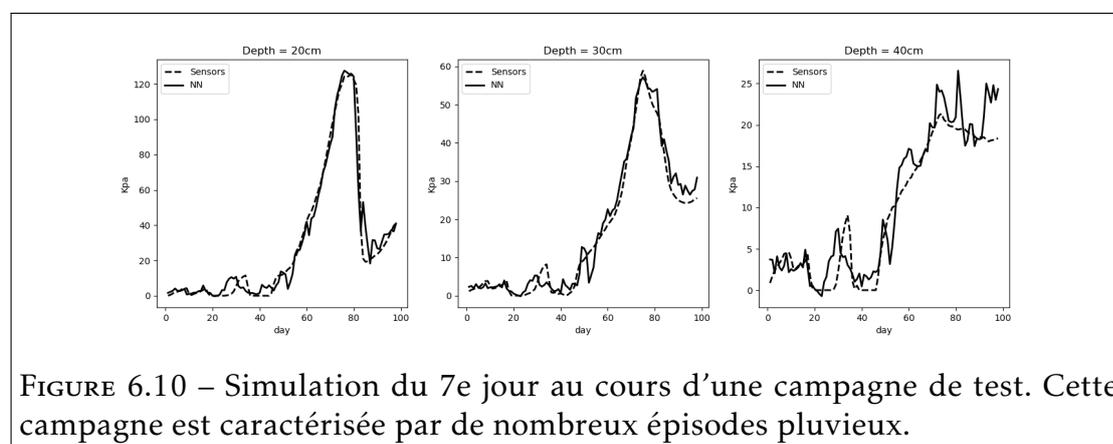
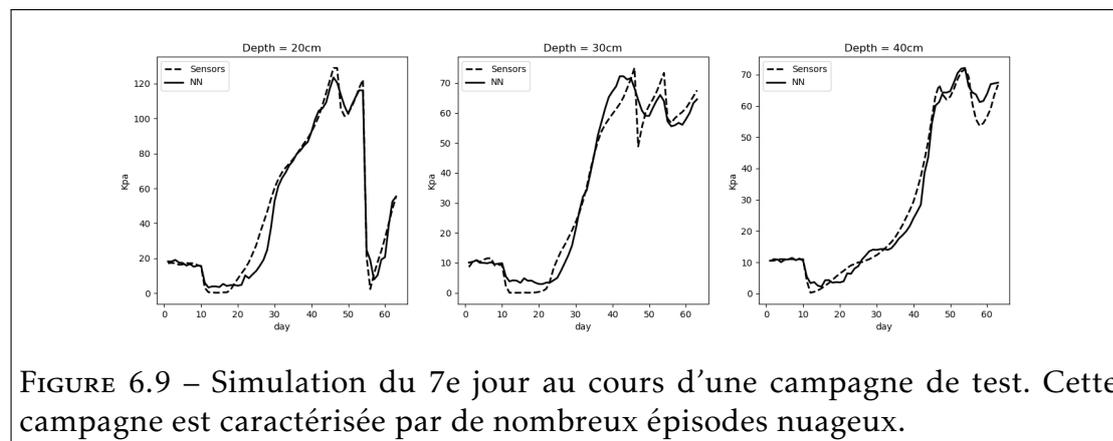
	SVM	SVM_FM	SVM_RFE
Moy.	6.86	6.88	6.85
Min.	0.27	0.28	0.28
Max.	32.76	32.10	32.76
25%	2.24	2.23	2.25
50%	4.33	4.40	4.36
75%	9.27	9.33	9.03











sélectionnée. En effet, sur un terrain de plusieurs hectares, la variation du SWP entre deux zones de la parcelle peut être facilement supérieure à 10 kpa, c'est-à-dire plus élevée que l'erreur commise par les algorithmes d'apprentissage. Il est donc préférable de choisir un algorithme avec une meilleure performance lorsque les conditions du phénomène sont défavorables, c'est-à-dire lorsque l'erreur de prédiction risque d'être élevée. Les figures 6.9, 6.10, 6.11 montrent les prévisions de l'outil d'aide à la décision (OAD) en utilisant le réseau de neurones comme algorithme d'apprentissage. Ces courbes représentent une solution proposée aux exploitants agricoles pour répondre à la problématique de la prédiction hebdomadaire de la SWP. La courbe noire illustre la valeur moyenne d'une journée mesurée réellement par un tensiomètre 7 jours plus tard alors que la courbe en pointillé représente la prédiction du modèle pour ce même jour. En fait, c'est la prédiction la plus difficile de notre expérience. La figure 6.9 représente une campagne pour laquelle notre modèle a eu les meilleures performances. La prévision du 7^e jour pour la campagne de test est très proche de la vérité. La figure 6.10 représente une campagne caractérisée par des pluies abondantes et régulières. Sur les 2 premières profondeurs (20 et 30 cm) les performances sont aussi plus que correctes. La prévision à la dernière profondeur semble plus chaotique. Cependant, il faut prendre en compte l'effet d'échelle du graphique. En effet, l'erreur est d'environ 5 Kpa, ce qui n'est pas très significatif pour des champs de plusieurs hectares. La figure 6.11 illustre la campagne avec les pires performances de notre modèle. Cette campagne a été affectée par une sécheresse et le sol a souvent été asséché. La base d'apprentissage n'inclut que très peu d'observations couvrant ces conditions. Ceci pouvant expliquer les prévisions médiocres, dans des conditions extrêmes, en particulier au début (avant le jour 20) et à la fin (après le jour 70) de la campagne. La chute du potentiel hydrique observable au jour 64 a pour origine une pluie que le pluviomètre avait indiquée alors que les tensiomètres ne l'avaient pas mesurée. Une explication possible est que la pluie s'est réellement produite, mais à une certaine distance de la culture (puisque les capteurs de pluie peuvent être situés à une distance maximale de 5 km).

Discussions Comme présenté sur les figures, le modèle (ici un réseau de neurones) est capable d'apprendre les tendances de chaque campagne, même si la qualité des prévisions peut être variable. Ces illustrations présentent plusieurs visuels des performances du modèle qui sera à terme proposé dans l'OAD développé afin de fournir ces prédictions aux exploitants agricoles dans le but de les informer sur les évolutions.

6.1.6 Synthèse

Dans cette contribution, le problème du potentiel hydrique dans le sol a été modélisé en un problème d'apprentissage supervisé. Le problème n'est pas trivial puisqu'il y a 21 sorties à prédire (3 profondeurs pour chaque jour et 7 jours à prédire). En outre, une méthodologie de sélection de variables afin de concevoir automatiquement des modèles en utilisant les variables les plus pertinentes est utilisée. L'objectif est d'aider les exploitants agricoles dans leur processus de décision d'un point de vue économique et écologique. Les informations fournies par des tensiomètres à bas coût ont été utilisées afin de rendre leur utilisation aussi démocratique que possible. Du fait du faible nombre de champs actuellement disponibles, une leave-two-out cross validation a permis de présenter des résultats statistiquement significatifs. La similitude des performances entre les méthodes de sélection de variables indique qu'un modèle issu de cette méthodologie peut avoir des performances similaires à un modèle ayant toutes les variables disponibles. En outre, si de nouvelles variables sont introduites, cette méthode rejettera automatiquement les variables les moins pertinentes. Toutefois, l'inconvénient majeur de ces méthodes implique le réglage des hyper paramètres. En ce qui concerne la méthode en elle-même, l'utilisation d'une approche par apprentissage automatique est pertinente puisque les résultats indiquent qu'il est possible d'obtenir une prédiction hebdomadaire correcte de l'évolution de la teneur en eau dans le sol. Cette contribution permet de valider la méthodologie et d'apporter une bonne alternative aux modèles paramétriques, ou à un échantillonnage destructif coûteux.

Bibliographie du présent chapitre

- [5] Iftikhar ALI et al. « Review of Machine Learning Approaches for Biomass and Soil Moisture Retrievals from Remote Sensing Data ». In : *Remote Sensing* 7.12 (2015), p. 16398-16421. ISSN : 2072-4292. DOI : 10.3390/rs71215841. URL : <https://www.mdpi.com/2072-4292/7/12/15841>.
- [6] Richard G ALLEN et al. « Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56 ». In : *Fao, Rome* 300.9 (1998), p. D05109.
- [19] N BRISSON et al. « An overview of the crop model stics ». In : *European Journal of Agronomy* 18.3 (2003). Modelling Cropping Systems : Science, Software and Applications, p. 309-332. ISSN : 1161-0301. DOI : [https://doi.org/10.1016/S1161-0301\(02\)00110-7](https://doi.org/10.1016/S1161-0301(02)00110-7). URL : <http://www.sciencedirect.com/science/article/pii/S1161030102001107>.
- [23] Dachun CHEN et Yingjie MA. « Optimized algorithm for estimating parameters by solving Van Genuchten equation based on stochastic particle swarm optimization ». In : *Nongye Gongcheng Xuebao(Transactions of the Chinese Society of Agricultural Engineering)* 22.12 (2006), p. 82-85.
- [24] Han CHEN, Jinhui Jeanne HUANG et Edward McBEAN. « Partitioning of daily evapotranspiration using a modified shuttleworth-wallace model, random Forest and support vector regression, for a cabbage farmland ». In : *Agricultural Water Management* 228 (2020), p. 105923. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2019.105923>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377419313034>.
- [25] Carlton CHU et al. « Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images ». In : *Neuroimage* 60.1 (2012), p. 59-70.
- [33] Dilini DELGODA et al. « Root zone soil moisture prediction models based on system identification : Formulation of the theory and validation using field and AQUACROP data ». In : *Agricultural Water Management* 163 (2016), p. 344-353. ISSN : 0378-3774.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.

- [40] EOS. *soil moisture control is an essential farming constituent*. 2019. URL : <https://eos.com/blog/soil-moisture-control-is-an-essential-farming-constituent/>.
- [46] Siyuan FENG et al. « 3D walking based on online optimization ». In : *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2013, p. 21-27.
- [47] Roberto FILGUEIRAS et al. « Soil water content and actual evapotranspiration predictions using regression algorithms and remote sensing data ». In : *Agricultural Water Management* 241 (2020), p. 106346. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2020.106346>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377420303097>.
- [49] Jérôme G FORTIN et al. « A neural network experiment on the site-specific simulation of potato tuber growth in Eastern Canada ». In : *Computers and Electronics in Agriculture* 73.2 (2010), p. 126-132.
- [54] Pablo M. GRANITTO et al. « Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products ». In : *Chemometrics and Intelligent Laboratory Systems* 83.2 (2006), p. 83-90. ISSN : 0169-7439. DOI : <https://doi.org/10.1016/j.chemolab.2006.01.007>. URL : <http://www.sciencedirect.com/science/article/pii/S0169743906000232>.
- [63] Terry HOWELL et Steve EVETT. « The Penman-Monteith Method ». In : (jan. 2004).
- [69] K.B. JOHNSON, S.B. JOHNSON et P.S. TENG. « Development of a simple potato growth model for use in crop-pest management ». In : *Agricultural Systems* 19.3 (1986), p. 189-209. ISSN : 0308-521X. DOI : [https://doi.org/10.1016/0308-521X\(86\)90052-1](https://doi.org/10.1016/0308-521X(86)90052-1). URL : <http://www.sciencedirect.com/science/article/pii/0308521X86900521>.
- [72] Kassu Tadesse KASSAYE et al. « Monitoring soil water content for decision supporting in agricultural water management based on critical threshold values adopted for Andosol in the temperate monsoon climate ». In : *Agricultural Water Management* 229 (2020), p. 105930. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2019.105930>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377419316312>.
- [75] M KUMAR et al. « Estimating evapotranspiration using artificial neural network ». In : *Journal of Irrigation and Drainage Engineering* 128.4 (2002), p. 224-233.

- [81] scikit LEARN. *Recursive Feature Elimination*. 2020. URL : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.
- [89] C. D. van LOON. « The effect of water stress on potato growth, development, and yield ». In : *American Potato Journal* 58 (1981). URL : <https://doi.org/10.1007/BF02855380>.
- [90] Fabian Löw et al. « Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using support vector machines ». In : *ISPRS journal of photogrammetry and remote sensing* 85 (2013), p. 102-119.
- [110] Dirk RAES et al. « AquaCrop—the FAO crop model to simulate yield response to water : II. Main algorithms and software description ». In : *Agronomy Journal* 101.3 (2009), p. 438-447.
- [111] Eric RAMAT et Bertrand VANDOORNE. *Plant growth model for decision making support*. Rapp. tech. Université du Littoral Côte d’Opale, et ISA Lille, 2002.
- [116] AA SABZIPARVAR et H TABARI. « Comparison of artificial neural network models and non-linear regression methods for estimation of potato crop evapotranspiration in a semi-arid region of Iran ». In : *The international conference on intelligent network and computing*, Nov. 2010, p. 26-28.
- [124] Soroosh SOROOSHIAN, Qingyun DUAN et Vijai Kumar GUPTA. « Calibration of rainfall-runoff models : Application of global optimization to the Sacramento Soil Moisture Accounting Model ». In : *Water Resources Research* 29.4 (1993), p. 1185-1194. DOI : 10.1029/92WR02617. eprint : <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/92WR02617>. URL : <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/92WR02617>.
- [125] Hossein TABARI et al. « Applicability of support vector machines and adaptive neurofuzzy inference system for modeling potato crop evapotranspiration ». In : *Irrigation science* 31.4 (2013), p. 575-588.
- [129] Roque TORRES-SANCHEZ et al. « A Decision Support System for Irrigation Management : Analysis and Implementation of Different Learning Techniques ». In : *Water* 12.2 (2020). ISSN : 2073-4441. DOI : 10.3390/w12020548. URL : <https://www.mdpi.com/2073-4441/12/2/548>.
- [132] M Th VAN GENUCHTEN. « A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1 ». In : *Soil science society of America journal* 44.5 (1980), p. 892-898.

-
- [138] Sevim Seda YAMAÇ et Mladen TODOROVIC. « Estimation of daily potato crop evapotranspiration using three different machine learning algorithms and four scenarios of available meteorological data ». In : *Agricultural Water Management* 228 (2020), p. 105875.
- [140] R ZAKALUK et R Sri RANJAN. « Artificial neural network modelling of leaf water potential for potatoes using RGB digital images : a greenhouse study ». In : *Potato Research* 49.4 (2006), p. 255-272.
- [142] Hai-xiang ZHAO et Frédéric MAGOULÈS. « Feature selection for support vector regression in the application of building energy prediction ». In : *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE. 2011, p. 219-223.

Conclusion

Ce manuscrit se place dans le contexte de la collaboration entre le LISIC et la société WEENAT. L'objectif de cette collaboration est la modélisation et la conception d'outils d'aide à la décision pour les exploitants agricoles en se focalisant sur la culture de la pomme de terre. Les chapitres 1, 3 et 5 présentent un panorama de la littérature relative aux problèmes d'optimisation continue multimodale, aux problèmes d'apprentissage par renforcement et supervisé. Plusieurs contributions majeures sont présentées dans ce manuscrit. Les contributions [36] et [38] ont été publiées dans des conférences internationales avec comité de relecture tandis que la contribution [37] a été publiée dans un journal. En premier lieu, le chapitre 2 détaille une nouvelle méthodologie de réglage automatique de paramètres de modèles agronomiques par transformation en un problème d'optimisation numérique multimodale boîte noire dans un contexte hors ligne. La fonction objective est donnée par analyse inverse et la résolution du problème est effectuée par un algorithme d'optimisation utilisant une stratégie d'évolution combinée avec une stratégie de restart. Une comparaison d'algorithmes a montré la pertinence de l'algorithme *QRDS* pour résoudre ce type de problème. Ces résultats permettent de valider la démarche et la méthodologie pour répondre à cette problématique. Le chapitre 4 propose d'améliorer la sélection des positions initiales des recherches locales de l'algorithme *RDS*. Deux nouvelles contributions sont détaillées. Ces deux nouveaux algorithmes d'optimisation numérique multimodale boîte noire avec une stratégie d'évolution utilisent une stratégie de restart assistée par un algorithme d'apprentissage par renforcement. Le problème du choix de la position initiale est reformulé comme un problème de bandit manchot et le compromis exploration/exploitation est géré par la méthode *Upper Confidence Bound*.

Le premier algorithme *UCB Random-restarts with Decreasing Step-size* (URDS) partitionne initialement l'espace de recherche et considère chaque sous-espace comme un bras du bandit. Le choix d'un sous-espace est motivé par le score UCB de ce dernier. Les résultats démontrent la pertinence de l'algorithme à apprendre dynamiquement dans quelle(s) zone(s) de l'espace de recherche l'algorithme doit concentrer son budget d'évaluations afin de maximiser son espérance pour trouver le maximum d'optima. Il montre, de plus, sa supériorité face aux algorithmes *RDS* et *QRDS* pour des problèmes de faible dimension. Le deuxième algorithme, *Upper Confidence Tree for Random restarts with Decreasing Step-size* (UCT-RDS), est une continuité des travaux de l'algorithme URDS pour une application sur des problèmes de plus grande dimension. Cet algorithme partitionne dynamiquement l'espace de recherche grâce à l'algorithme *Upper Confidence Tree*. Le compromis exploration/exploitation est maintenu durant la phase de sélection à l'aide de la méthode UCB1. La construction dynamique de l'arbre permet à l'algorithme de gérer des problèmes de plus grande dimension puisque le nombre de sous-espaces est indépendant de la dimension du problème. Les résultats démontrent la pertinence de ce nouvel algorithme pour sélectionner efficacement les zones les plus prometteuses. Les performances de *UCT-RDS* sont équivalentes à celles de l'algorithme *QRDS* et meilleures lorsque les problèmes sont difficiles, c'est-à-dire lorsque le nombre d'optima à découvrir est important ou bien si la fonction possède des plateaux. Le chapitre 6 propose une redéfinition d'un problème de prédiction de phénomènes biologiques en un problème d'apprentissage. Ce problème est résolu par un algorithme d'apprentissage supervisé. La base de données est développée à partir de l'agrégation de mesures historiques issues des mesures capteurs. En outre, le choix des variables est automatiquement réalisé par différents algorithmes de sélection de variables. Au vu du nombre restreint de parcelles disponibles, une *leave-two-out-cross-validation* permet d'obtenir un échantillon suffisamment important pour que les performances des modèles soient statistiquement significatives. Il en résulte qu'il est possible d'une part de prédire efficacement le phénomène sur une période hebdomadaire et d'autre part de générer des modèles de plus faible dimension ayant des performances similaires aux modèles de base, c'est-à-dire ayant toutes les variables disponibles en entrée.

Les différentes contributions détaillées dans ce manuscrit ont aussi pour vocation d'ouvrir des perspectives sur des travaux futurs. En effet, concernant l'optimisation des paramètres de modèle biologique, les jeux de paramètres optimisés pourraient servir de portefeuille de solutions et ainsi fournir des réponses satisfaisantes pour de nouvelles parcelles sans avoir recours à un nouveau processus d'optimisation. À propos des améliorations de l'algorithme RDS, l'algorithme *UCT-RDS* démontre un fort potentiel dans la résolution de problèmes multimodaux. Remplacer la recherche locale par un algorithme d'optimisation utilisant une stratégie d'évolution à population comme *EMNA* pourrait améliorer les performances actuelles. Le découpage orthogonal de l'algorithme est aussi une limitation. Un découpage indépendant de la dimension ou bien une méthode d'échantillonnage comme l'hypercube latin permettraient à *UCT-RDS* de gérer des problèmes de plus grande dimension. Enfin, l'approche par apprentissage supervisé pour résoudre le problème de l'évolution de la teneur en eau est une piste sérieuse à étudier. Les méthodes "few-shot learning" sont également une piste d'exploration intéressante. Elles offrent la capacité d'apprendre à partir d'un nombre (très) limité d'exemples, ce qui pourrait pallier au nombre d'observations actuellement limité. Avec davantage de données, d'autres types de sols ou de cultures ou de conditions climatiques pourraient être étudiés et ainsi améliorer la capacité de généralisation des modèles prédictifs actuels. Les méthodes de sélection de variables ont un impact important sur le processus de conception du modèle. Il est aussi pertinent d'explorer d'autres approches ou algorithmes de sélection de variables. Inclusion de nouvelles variables comme la temporalité est à étudier. Dans ce cas une piste possible est l'utilisation de réseaux récurrents types *RNN* ou *LSTM*. Leurs capacités à trouver des symétries naturelles, des schémas ou d'autres similarités liées aux phénomènes temporels devraient améliorer les performances actuelles.

En conclusion, ce manuscrit vise à faire prendre conscience des enjeux socio-économiques liés à l'utilisation des ressources en eau potable et des problèmes de gestion de l'irrigation dans l'agriculture. Il propose aussi des réponses aux problématiques grâce aux techniques modernes d'optimisation et d'apprentissage dans un contexte d'agriculture connectée plus écoresponsable.

Bibliographie du présent chapitre

- [36] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Improving Multi-Modal Optimization Restart Strategy Through Multi-Armed Bandit ». In : *IEEE ICMLA 2018 : 17th IEEE International Conference On Machine Learning And Applications*. 2018.
- [37] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Upper Confidence Tree for planning restart strategies in Multi-Modal Optimization ». In : *Soft Computing* (juill. 2020). DOI : 10.1007/s00500-020-05196-w.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.

Bibliographie

- [1] *Agriculture et Territoire* Chambre d'agriculture du Nord-Pas-de-Calais. <https://nord-pas-de-calais.chambre-agriculture.fr/pommes-de-terre/>. Accessed : 2020-05-6.
- [2] Sajjad AHMAD, Ajay KALRA et Haroon STEPHEN. « Estimating soil moisture using remote sensing data : A machine learning approach ». In : *Advances in Water Resources* 33.1 (2010), p. 69-80.
- [3] Ali AHRARI, Kalyanmoy DEB et Mike PREUSS. « Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations ». In : *Evolutionary computation* 25.3 (2017), p. 439-471.
- [4] Francisco Chicano et AL.. « Optimizing one million variable NK landscapes by hybridizing deterministic recombination and local search ». In : *GECCO* (2017), p. 753-760.
- [5] Iftikhar ALI et al. « Review of Machine Learning Approaches for Biomass and Soil Moisture Retrievals from Remote Sensing Data ». In : *Remote Sensing* 7.12 (2015), p. 16398-16421. ISSN : 2072-4292. DOI : 10.3390/rs71215841. URL : <https://www.mdpi.com/2072-4292/7/12/15841>.
- [6] Richard G ALLEN et al. « Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56 ». In : *Fao, Rome* 300.9 (1998), p. D05109.
- [7] Peter AUER, Nicolo CESA-BIANCHI et Paul FISCHER. « Finite-time analysis of the multiarmed bandit problem ». In : *Machine learning* 47.2-3 (2002), p. 235-256.
- [8] Peter AUER, Nicolò CESA-BIANCHI et Paul FISCHER. « Finite-time Analysis of the Multiarmed Bandit Problem ». In : *Mach. Learn.* 47.2-3 (2002), p. 235-256. ISSN : 0885-6125. DOI : 10.1023/A:1013689704352.

- [9] Anne AUGER. « Benchmarking the (1+ 1) evolution strategy with one-fifth success rule on the BBOB-2009 function testbed ». In : *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference : Late Breaking Papers*. 2009, p. 2447-2452.
- [10] Anne AUGER et Nikolaus HANSEN. « A restart CMA evolution strategy with increasing population size ». In : *2005 IEEE congress on evolutionary computation*. T. 2. IEEE. 2005, p. 1769-1776.
- [11] Anne AUGER et Nikolaus HANSEN. « A restart CMA evolution strategy with increasing population size ». In : *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. T. 2. 2005, p. 1769-1776.
- [12] Thomas BACK, Frank HOFFMEISTER et Hans-Paul SCHWEFEL. « A survey of evolution strategies ». In : *Proceedings of the fourth international conference on genetic algorithms*. T. 2. 9. Morgan Kaufmann Publishers San Mateo, CA. 1991.
- [13] Hans-Georg BEYER. *The Theory of Evolution Strategies*. Springer Science & Business Media, 2001.
- [14] Hans-Georg BEYER et Hans-Paul SCHWEFEL. « Evolution strategies—A comprehensive introduction ». In : *Natural computing* 1.1 (2002), p. 3-52.
- [15] *bnpe eau de France*. <https://bnpe.eaufrance.fr/acces-donnees/codeDepartement/59/annee/2017/usage/2>. updated : 2020-01-16.
- [16] Léon BOTTOU. « Stochastic gradient descent tricks ». In : *Neural networks : Tricks of the trade*. Springer, 2012, p. 421-436.
- [17] L BREIMAN et al. « Classification and Regression Trees ». In : (1984).
- [18] *brgm*. <https://www.brgm.fr/publication-presse/etat-nappes-eau-souterraine-1er-mai-2019>. published : 2019-05-01.
- [19] N BRISSON et al. « An overview of the crop model stics ». In : *European Journal of Agronomy* 18.3 (2003). Modelling Cropping Systems : Science, Software and Applications, p. 309-332. ISSN : 1161-0301. DOI : [https://doi.org/10.1016/S1161-0301\(02\)00110-7](https://doi.org/10.1016/S1161-0301(02)00110-7). URL : <http://www.sciencedirect.com/science/article/pii/S1161030102001107>.
- [20] Michele CALVELLO et Richard J FINNO. « Selecting parameters to optimize in model calibration by inverse analysis ». In : *Computers and Geotechnics* 31.5 (2004), p. 410-424. ISSN : 0266-352X. DOI : <https://doi.org/10.1016/j.compgeo.2004.03.004>. URL : <http://www.sciencedirect.com/science/article/pii/S0266352X04000540>.
- [21] Yan CAO et al. « Graph edge coloring : A survey ». In : *Graphs and Combinatorics* 35.1 (2019), p. 33-66.

- [22] Girish CHANDRASHEKAR et Ferat SAHIN. « A survey on feature selection methods ». In : *Computers and Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, p. 16-28. ISSN : 0045-7906. DOI : <https://doi.org/10.1016/j.compeleceng.2013.11.024>. URL : <http://www.sciencedirect.com/science/article/pii/S0045790613003066>.
- [23] Dachun CHEN et Yingjie MA. « Optimized algorithm for estimating parameters by solving Van Genuchten equation based on stochastic particle swarm optimization ». In : *Nongye Gongcheng Xuebao(Transactions of the Chinese Society of Agricultural Engineering)* 22.12 (2006), p. 82-85.
- [24] Han CHEN, Jinhui Jeanne HUANG et Edward McBEAN. « Partitioning of daily evapotranspiration using a modified shuttleworth-wallace model, random Forest and support vector regression, for a cabbage farmland ». In : *Agricultural Water Management* 228 (2020), p. 105923. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2019.105923>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377419313034>.
- [25] Carlton CHU et al. « Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images ». In : *Neuroimage* 60.1 (2012), p. 59-70.
- [26] U.S department COMMERCE. *Official gazette of the United States patent and trademark office*. Sous la dir. d'Authority of CONGRESS. T. 103. 1981.
- [27] Alessandro CRESPI et Auke Jan IJSPEERT. « Online optimization of swimming and crawling in an amphibious snake robot ». In : *IEEE Transactions on Robotics* 24.1 (2008), p. 75-87.
- [28] *Crop life 2019 Precision Agriculture Dealership Survey : More Moves Toward Decision Agriculture*. <https://www.croplife.com/management/2019-precision-agriculture-dealership-survey-more-moves-toward-decision-agriculture/>. published : 2019-08-21.
- [29] Conseil d'ÉTAT. *Rapport annuel 2010 du Conseil d'État (données consommations)*. 2010.
- [30] T.M. DARWISH et al. « Nitrogen and water use efficiency of fertigated processing potato ». In : *Agricultural Water Management* 85.1 (2006), p. 95-104. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2006.03.012>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377406001028>.
- [31] Kenneth Alan DE JONG. « An Analysis of the Behavior of a Class of Genetic Adaptive Systems. » Thèse de doct. University of Michigan, 1975.

- [32] A DELAGE et H LAGATU. « Sur les résultats obtenus par l'observation des terres arables en plaques minces ». In : *Compt. Rend.* 140 (1905), p. 1555.
- [33] Dilini DELGODA et al. « Root zone soil moisture prediction models based on system identification : Formulation of the theory and validation using field and AQUACROP data ». In : *Agricultural Water Management* 163 (2016), p. 344-353. ISSN : 0378-3774.
- [34] Stanley N DEMING, Lloyd R PARKER JR et M BONNER DENTON. « A review of simplex optimization in analytical chemistry ». In : *CRC Critical Reviews in Analytical Chemistry* 7.3 (1978), p. 187-202.
- [35] Harris DRUCKER et al. « Support vector regression machines ». In : *Advances in neural information processing systems*. 1997, p. 155-161.
- [36] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Improving Multi-Modal Optimization Restart Strategy Through Multi-Armed Bandit ». In : *IEEE ICMLA 2018 : 17th IEEE International Conference On Machine Learning And Applications*. 2018.
- [37] Amaury DUBOIS, Julien DEHOS et Fabien TEYTAUD. « Upper Confidence Tree for planning restart strategies in Multi-Modal Optimization ». In : *Soft Computing* (juill. 2020). DOI : 10.1007/s00500-020-05196-w.
- [38] Amaury DUBOIS et al. « Automatic calibration of a farm irrigation model : a multi-modal optimization approach ». In : *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2019, p. 192-204.
- [39] Michael D DUKES, Lincoln ZOTARELLI et Kelly T MORGAN. « Use of irrigation technologies for vegetable crops in Florida ». In : *HortTechnology* 20.1 (2010), p. 133-142.
- [40] EOS. *soil moisture control is an essential farming constituent*. 2019. URL : <https://eos.com/blog/soil-moisture-control-is-an-essential-farming-constituent/>.
- [41] Donald C ERBACH. « Measurement of soil bulk density and moisture ». In : *Transactions of the ASAE* 30.4 (1987), p. 922-0931.
- [42] Michel Crétenet ERIC JALLAS. « Aide à la décision en agriculture et gestion des ressources naturelles : note introductive. Savanes africaines : des espaces en mutation, des acteurs face à de nouveaux défis. » In : *HAL* hal-00143508 (2003). URL : <https://hal.archives-ouvertes.fr/hal-00143508/document>.
- [43] FAO. *FAO kc*. Accessed : 2020-05-12. URL : <http://www.fao.org/3/x0490e/x0490e0b.htm>.

- [44] FAO. *FAO Penman-Monteith equation*. Accessed : 2020-05-12. URL : <http://www.fao.org/3/x0490e/x0490e06.htm#chapter%20%20%2020fao%20penman%20monteith%20equation>.
- [45] FAO. *Sprinkler irrigation*. URL : <http://www.fao.org/3/S8684E/s8684e06.htm>.
- [46] Siyuan FENG et al. « 3D walking based on online optimization ». In : *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2013, p. 21-27.
- [47] Roberto FILGUEIRAS et al. « Soil water content and actual evapotranspiration predictions using regression algorithms and remote sensing data ». In : *Agricultural Water Management* 241 (2020), p. 106346. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2020.106346>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377420303097>.
- [48] A Richer de FORGES et al. « Perdus dans le triangle des textures ». In : *Etudes et Gestion des Sols* 15.2 (2008), p. 97-111.
- [49] Jérôme G FORTIN et al. « A neural network experiment on the site-specific simulation of potato tuber growth in Eastern Canada ». In : *Computers and Electronics in Agriculture* 73.2 (2010), p. 126-132.
- [50] Montesano FRANCESCO et al. *Modern technologies, strategies and tools for sustainable irrigation management and governance in Mediterranean Agriculture*. Sept. 2015. ISBN : 2-85352-549-X.
- [51] Michel VALLANCE FRANÇOIS COLAS-BELCOUR Roland RENOULT. *Rapport n° 14061 : Synthèse Eau et Agriculture*. Rapp. tech. Ministère de l'agriculture de l'agroalimentaire et de la forêt, 2015.
- [52] Sylvain GELLY et David SILVER. « Combining Online and Offline Knowledge in UCT ». In : *International Conference of Machine Learning*. 2007.
- [53] David E GOLDBERG, Jon RICHARDSON et al. « Genetic algorithms with sharing for multimodal function optimization ». In : *Genetic algorithms and their applications : Proceedings of the Second International Conference on Genetic Algorithms*. 1987, p. 41-49.
- [54] Pablo M. GRANITTO et al. « Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products ». In : *Chemometrics and Intelligent Laboratory Systems* 83.2 (2006), p. 83-90. ISSN : 0169-7439. DOI : <https://doi.org/10.1016/j.chemolab.2006.01.007>. URL : <http://www.sciencedirect.com/science/article/pii/S0169743906000232>.

- [55] Genetha Anne GRAY et al. « Optimizing an empirical scoring function for transmembrane protein structure determination ». In : *INFORMS Journal on Computing* 16.4 (2004), p. 406-418.
- [56] Hoshin V GUPTA, Keith J BEVEN et Thorsten WAGENER. « Model calibration and uncertainty estimation ». In : *Encyclopedia of hydrological sciences* (2006).
- [57] Isabelle GUYON et André ELISSEEFF. « An introduction to variable and feature selection ». In : *Journal of machine learning research* 3.Mar (2003), p. 1157-1182.
- [58] Jeongwoo HAN, Michael KOKKOLARAS et Panos Y PAPALAMBROS. « Optimal design of hybrid fuel cell vehicles ». In : *Journal of fuel cell science and technology* 5.4 (2008).
- [59] Nikolaus HANSEN. « The CMA Evolution Strategy : A Comparing Review ». In : t. 192. Juin 2007, p. 75-102. DOI : 10.1007/3-540-32494-1_4.
- [60] Nikolaus HANSEN et Stefan KERN. « Evaluating the CMA evolution strategy on multimodal test functions ». In : *International Conference on Parallel Problem Solving from Nature*. Springer. 2004, p. 282-291.
- [61] Nikolaus HANSEN, Sibylle D MÜLLER et Petros KOUMOUTSAKOS. « Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES) ». In : *Evolutionary computation* 11.1 (2003), p. 1-18.
- [62] Tin Kam Ho. « Random decision forests ». In : *Proceedings of 3rd international conference on document analysis and recognition*. T. 1. IEEE. 1995, p. 278-282.
- [63] Terry HOWELL et Steve EVETT. « The Penman-Monteith Method ». In : (jan. 2004).
- [64] David HUNTER et al. « Selection of proper neural network sizes and architectures A comparative study ». In : *IEEE Transactions on Industrial Informatics* 8.2 (2012), p. 228-240.
- [65] INSEE. *Insee Références, édition 2016 - Fiches - Marché du travail*. 2016.
- [66] *itk Mission Intelligence Artificielle*. <http://www.itk.fr/wp-content/uploads/2018/04/ITK-Intelligence-Artificielle-Agriculture.pdf>. published : 2017-11-28.
- [67] J. L. JEREZ et al. « Embedded Online Optimization for Model Predictive Control at Megahertz Rates ». In : *IEEE Transactions on Automatic Control* 59.12 (2014), p. 3238-3251.

- [68] Yufeng JIA et Shichun CHI. « Back-analysis of soil parameters of the Malutang II concrete face rockfill dam using parallel mutation particle swarm optimization ». In : *Computers and Geotechnics* 65 (2015), p. 87-96. ISSN : 0266-352X. DOI : <https://doi.org/10.1016/j.compgeo.2014.11.013>. URL : <http://www.sciencedirect.com/science/article/pii/S0266352X14002183>.
- [69] K.B. JOHNSON, S.B. JOHNSON et P.S. TENG. « Development of a simple potato growth model for use in crop-pest management ». In : *Agricultural Systems* 19.3 (1986), p. 189-209. ISSN : 0308-521X. DOI : [https://doi.org/10.1016/0308-521X\(86\)90052-1](https://doi.org/10.1016/0308-521X(86)90052-1). URL : <http://www.sciencedirect.com/science/article/pii/0308521X86900521>.
- [70] Serdar KADIOGLU, Meinolf SELLMANN et Markus WAGNER. « Learning a Reactive Restart Strategy to Improve Stochastic Search ». In : *International Conference on Learning and Intelligent Optimization*. Springer, 2017, p. 109-123.
- [71] Ali KAMRAN et Liang GUOZHU. « Design and Optimization of 3D Radial Slot Grain Configuration ». In : *Chinese Journal of Aeronautics* 23.4 (2010), p. 409-414. ISSN : 1000-9361. DOI : [https://doi.org/10.1016/S1000-9361\(09\)60235-1](https://doi.org/10.1016/S1000-9361(09)60235-1). URL : <http://www.sciencedirect.com/science/article/pii/S1000936109602351>.
- [72] Kassu Tadesse KASSAYE et al. « Monitoring soil water content for decision supporting in agricultural water management based on critical threshold values adopted for Andosol in the temperate monsoon climate ». In : *Agricultural Water Management* 229 (2020), p. 105930. ISSN : 0378-3774. DOI : <https://doi.org/10.1016/j.agwat.2019.105930>. URL : <http://www.sciencedirect.com/science/article/pii/S0378377419316312>.
- [73] Valentin KIENY. *Température & agriculture*. http://infoclimat.org/MeteoALecole/ressources_scientifiques/. Accessed : 2020.
- [74] L. KOCSIS et C. SZEPESVÁRI. « Bandit based Monte-Carlo planning ». In : *Machine Learning : ECML 2006*. Springer, 2006, p. 282-293.
- [75] M KUMAR et al. « Estimating evapotranspiration using artificial neural network ». In : *Journal of Irrigation and Drainage Engineering* 128.4 (2002), p. 224-233.
- [76] Benjamin LACROIX, Lee A CHRISTIE et John A.W. MCALL. « Interpolated Continuous Optimisation Problems with Tunable Landscape Features ». In : *Proc. of GECCO 17 Companion*. 2017.

- [77] Tze Leung LAI et Herbert ROBBINS. « Asymptotically efficient adaptive allocation rules ». In : *Advances in applied mathematics* 6.1 (1985), p. 4-22.
- [78] Élise Deniel LAURENCE MENET Solenn Leplay. *Économiser leau en changeant les pratiques agricoles : retours d'expériences en Europe*. Rapp. tech. Ministère de l'agriculture et de l'alimentaire, 2018.
- [79] scikit LEARN. *Feature Selection*. 2020. URL : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html.
- [80] scikit LEARN. *random forest regressor*. 2020. URL : <https://scikit-learn.org/stable/modules/ensemble.html>.
- [81] scikit LEARN. *Recursive Feature Elimination*. 2020. URL : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.
- [82] Claude LEMARÉCHAL. *Optimisation continue*. published : 2002-03-10. URL : <https://www.techniques-ingenieur.fr/base-documentaire/automatique-robotique-th16/automatique-sequentielle-42395210/optimisation-continue-s7210/>.
- [83] Hal LEMMON. « COMAX : An expert system for cotton crop management ». In : *Science* 233.4759 (1986), p. 29-33.
- [84] Florian LEPRÊTRE, Fabien TEYTAUD et Julien DEHOS. « Multi-armed bandit for stratified sampling : Application to numerical integration ». In : *TAAI 2017*. Taipei, Taiwan, 2017.
- [85] Florian LEPRÊTRE et al. « Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on SIALAC benchmark ». In : *Applied Soft Computing* 85 (2019), p. 105869.
- [86] Jérôme LEROY. *site web weenat*. Accessed : 2020-05-12. URL : <https://weenat.com/>.
- [87] *Les aides Le site d'information des CCI sur les aides aux d'entreprises*. <https://les-aides.fr/fiche/a5Z1DH1GwPjMB3ZQ/conseil-regional/aide-aux-investissements-dans-les-infrastructures-liees-a-l-irrigation-agricole.html>. Accessed : 2020-05-06.

- [88] T. Warren LIAO. « Two hybrid differential evolution algorithms for engineering design optimization ». In : *Applied Soft Computing* 10.4 (2010). Optimisation Methods Applications in Decision-Making Processes, p. 1188-1199. ISSN : 1568-4946. DOI : <https://doi.org/10.1016/j.asoc.2010.05.007>. URL : <http://www.sciencedirect.com/science/article/pii/S1568494610001080>.
- [89] C. D. van LOON. « The effect of water stress on potato growth, development, and yield ». In : *American Potato Journal* 58 (1981). URL : <https://doi.org/10.1007/BF02855380>.
- [90] Fabian Löw et al. « Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using support vector machines ». In : *ISPRS journal of photogrammetry and remote sensing* 85 (2013), p. 102-119.
- [91] Manuel LOZANO, Daniel MOLINA et Francisco HERRERA. « Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems ». In : *Soft Computing* 15.11 (2011), p. 2085-2087.
- [92] Samir W MAHFOUD. « Niching methods for genetic algorithms ». In : *Urbana* 51.95001 (1995), p. 62-94.
- [93] Alison L MARSDEN, Jeffrey A FEINSTEIN et Charles A TAYLOR. « A computational framework for derivative-free optimization of cardiovascular geometries ». In : *Computer methods in applied mechanics and engineering* 197.21-24 (2008), p. 1890-1905.
- [94] C MATHIEU et al. « Physical Analysis of Soils : Selected Methods ». In : *Lavoisier TEC & DOC, PARIS ed* (1998).
- [95] Uwe MEIER. *Entwicklungsstadien mono- und dikotyle Pflanz*en : BBCH Monografie. Quedlinburg : Open Agrar Repository, 2018. ISBN : 978-3-95547-070-8. DOI : 10.5073/20180906-075119. URL : https://www.openagrar.de/rsc/thumbnail/openagrar_mods_00042352.png.
- [96] Ole J MENGSHOEL et David E GOLDBERG. « Probabilistic crowding : Deterministic crowding with probabilistic replacement ». In : *Proc. of GECCO conference*. 1999, p. 409-416.
- [97] *Ministere de l'Agriculture et de l'Alimentation Économiser leau pour irrigation par les changements de pratiques agricoles*. <https://agriculture.gouv.fr/economiser-leau-pour-irrigation-par-les-changements-de-pratiques-agricoles>. published : 2018-07-03.

- [98] *Ministere de l'Agriculture et de l'Alimentation* <https://agriculture.gouv.fr/afclim-agriculture-foret-climat-vers-des-strategies-dadaptation-0>. <https://agriculture.gouv.fr/economiser-leau-pour-lirrigation-par-les-changements-de-pratiques-agricoles>. published : 2019-06-16.
- [99] FRANCESCO F. MONTESANO et al. « Tensiometer-Based Irrigation Management of Subirrigated Soilless Tomato : Effects of Substrate Matric Potential Control on Crop Performance ». In : *Frontiers in Plant Science* 6 (2015), p. 1150. ISSN : 1664-462X. DOI : 10.3389/fpls.2015.01150. URL : <https://www.frontiersin.org/article/10.3389/fpls.2015.01150>.
- [100] Mathieu MUNIGLIA et al. « Massive asynchronous master-worker EA for nuclear reactor optimization : a fitness landscape perspective ». In : *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2017, p. 295-296.
- [101] Remi MUNOS. « Optimistic Optimization of Deterministic Functions without the Knowledge of its Smoothness ». In : *Advances in Neural Information Processing Systems* (2011), p. 783-791.
- [102] NITISH SRIVASTAVA AND GEOFFREY HINTON AND ALEX KRIZHEVSKY AND ILYA SUTSKEVER AND RUSLAN SALAKHUTDINOV. « Dropout : A Simple Way to Prevent Neural Networks from Overfitting ». In : *Journal of Machine Learning Research* 15 (2014), p. 1929-1958. URL : <http://jmlr.org/papers/v15/srivastava14a.html>.
- [103] Rodrigue OEUVRAY. *Trust-region methods based on radial basis functions with application to biomedical imaging*. Rapp. tech. EPFL, 2005.
- [104] PACHEPSKY, YA. A., D. TIMLIN, AND G. VARALLYAY. « Artificial Neural Networks to Estimate Soil Water Retention from Easily Measurable Data. » In : *Soil Sci. Soc. Am. J.* 60 (1998), p. 727-733. DOI : 10.2136/sssaj1996.03615995006000030007.
- [105] Neelmadhab PADHY et Suresh Chandra SATAPATHY. « Digital Advancement in AgriTech by Using Evolutionary Techniques ». In : *Smart Intelligent Computing and Applications*. Springer, 2020, p. 361-368.
- [106] Christos H PAPADIMITRIOU et Kenneth STEIGLITZ. *Combinatorial optimization : algorithms and complexity*. Courier Corporation, 1998.
- [107] Alain PÉTROWSKI. « A clearing procedure as a niching method for genetic algorithms ». In : *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. 1996, p. 798-803.

- [108] Khanh PHAM et al. « Analysis of neural network based pedotransfer function for predicting soil water characteristic curve ». In : *Geoderma* 351 (2019), p. 92-102. ISSN : 0016-7061. DOI : <https://doi.org/10.1016/j.geoderma.2019.05.013>. URL : <http://www.sciencedirect.com/science/article/pii/S0016706118315179>.
- [109] Philippe PREUX, Remi MUNOS et Michal VALKO. « Bandits attack function optimization ». In : *IEEE Congress on Evolutionary Computation*. 2014.
- [110] Dirk RAES et al. « AquaCrop—the FAO crop model to simulate yield response to water : II. Main algorithms and software description ». In : *Agronomy Journal* 101.3 (2009), p. 438-447.
- [111] Eric RAMAT et Bertrand VANDOORNE. *Plant growth model for decision making support*. Rapp. tech. Université du Littoral Côte d’Opale, et ISA Lille, 2002.
- [112] J. RAPIN et O. TEYTAUD. *Nevergrad - A gradient-free optimization platform*. <https://GitHub.com/FacebookResearch/Nevergrad>. 2018.
- [113] Ingo RECHENBERG. « Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution ». In : (1973).
- [114] David G ROSSITER. « ALES : a framework for land evaluation using a microcomputer ». In : *Soil use and management* 6.1 (1990), p. 7-20.
- [115] Sebastian RUDER. « An overview of gradient descent optimization algorithms ». In : *arXiv preprint arXiv:1609.04747* (2016).
- [116] AA SABZIPARVAR et H TABARI. « Comparison of artificial neural network models and non-linear regression methods for estimation of potato crop evapotranspiration in a semi-arid region of Iran ». In : *The international conference on intelligent network and computing, Nov. 2010*, p. 26-28.
- [117] Reem SALIM et al. « The parameter identification of the Nexa 1 2 kW PEMFC’s model using particle swarm optimization ». In : *Renewable Energy* 82 (2015). International Conference on Renewable Energy : Generation and Applications (ICREGA 2014), p. 26-34. ISSN : 0960-1481. DOI : <https://doi.org/10.1016/j.renene.2014.10.012>. URL : <http://www.sciencedirect.com/science/article/pii/S0960148114006399>.
- [118] SCHAAP, M. G., F. J. LEIJ, AND M. TH. VAN GENUCHTEN. « Neural Network Analysis for Hierarchical Prediction of Soil Hydraulic Properties. » In : *Soil Sci. Soc. Am. J.* 62 (1998), p. 847-855. DOI : 10.2136/sssaj1998.03615995006200040001.

- [119] Marc SCHOENAUER, Fabien TEYTAUD et Olivier TEYTAUD. « A Rigorous Runtime Analysis for Quasi-Random Restarts and Decreasing Stepsize ». In : *Artificial Evolution*. Angers, France, 2011.
- [120] MA SCHUMER et Kenneth STEIGLITZ. « Adaptive step size random search ». In : *IEEE Transactions on Automatic Control* 13.3 (1968), p. 270-276.
- [121] *Service Public Restrictions d'eau : quels sont les départements concernés*. <https://www.service-public.fr/particuliers/actualites/A13500>. published : 2019-09-18. 2019.
- [122] David SILVER. « Reinforcement Learning and Simulation-Based Search ». Thèse de doct. University of Alberta, 2009.
- [123] Gulshan SINGH et Kalyanmoy DEB DR. « Comparison of multi-modal optimization algorithms based on evolutionary algorithms ». In : *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 2006, p. 1305-1312.
- [124] Soroosh SOROOSHIAN, Qingyun DUAN et Vijai Kumar GUPTA. « Calibration of rainfall-runoff models : Application of global optimization to the Sacramento Soil Moisture Accounting Model ». In : *Water Resources Research* 29.4 (1993), p. 1185-1194. DOI : 10.1029/92WR02617. eprint : <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/92WR02617>. URL : <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/92WR02617>.
- [125] Hossein TABARI et al. « Applicability of support vector machines and adaptive neurofuzzy inference system for modeling potato crop evapotranspiration ». In : *Irrigation science* 31.4 (2013), p. 575-588.
- [126] Y TANG, P REED et Thibaut WAGENER. « How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? » In : (2006).
- [127] Fabien TEYTAUD et Olivier TEYTAUD. « QR Mutations Improve Many Evolution Strategies : A Lot On Highly Multimodal Problems ». In : *Proc. of the 2016 GECCO conference*. ACM. 2016, p. 35-36.
- [128] Martin THALHEIMER. « Tensiometer modification for diminishing errors due to the fluctuating inner water column ». In : *Soil Science Society of America Journal* 67.3 (2003), p. 737-739.
- [129] Roque TORRES-SANCHEZ et al. « A Decision Support System for Irrigation Management : Analysis and Implementation of Different Learning Techniques ». In : *Water* 12.2 (2020). ISSN : 2073-4441. DOI : 10.3390/w12020548. URL : <https://www.mdpi.com/2073-4441/12/2/548>.

- [130] Michal VALKO, Alexandra CARPENTIER et Remi MUNOS. « Stochastic Simultaneous Optimistic Optimization ». In : *Proc. of the 30th International Conference on Machine Learning*. T. 28. 2013.
- [131] Frans VAN DEN BERGH et Andries Petrus ENGELBRECHT. « Cooperative learning in neural networks using particle swarm optimizers ». In : *South African Computer Journal* 2000.26 (2000), p. 84-90.
- [132] M Th VAN GENUCHTEN. « A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1 ». In : *Soil science society of America journal* 44.5 (1980), p. 892-898.
- [133] Vladimir N VAPNIK. « The nature of statistical learning theory ». In : (1995).
- [134] Gerhard VENTER. « Review of optimization techniques ». In : *Encyclopedia of aerospace engineering* (2010).
- [135] Y. WANG et S. BOYD. « Fast Model Predictive Control Using Online Optimization ». In : *IEEE Transactions on Control Systems Technology* 18.2 (2010), p. 267-278.
- [136] Paul WERBOS. « Beyond Regression : " New Tools for Prediction and Analysis in the Behavioral Sciences ». In : *Ph. D. dissertation, Harvard University* (1974).
- [137] FD WHISLER et al. « Crop simulation models in agronomic systems ». In : *Adv. Agron* 40.1 (1986), p. 41-208.
- [138] Sevim Seda YAMAÇ et Mladen TODOROVIC. « Estimation of daily potato crop evapotranspiration using three different machine learning algorithms and four scenarios of available meteorological data ». In : *Agricultural Water Management* 228 (2020), p. 105875.
- [139] Xiaodong YIN et Noel GERMAÏ. « A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization ». In : *Artificial neural nets and genetic algorithms*. 1993, p. 450-457.
- [140] R ZAKALUK et R Sri RANJAN. « Artificial neural network modelling of leaf water potential for potatoes using RGB digital images : a greenhouse study ». In : *Potato Research* 49.4 (2006), p. 255-272.
- [141] Fedro S ZAZUETA et Jiannong XIN. « Soil moisture sensors ». In : *Soil Sci* 73 (1994), p. 391-401.

-
- [142] Hai-xiang ZHAO et Frédéric MAGOULÈS. « Feature selection for support vector regression in the application of building energy prediction ». In : *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE. 2011, p. 219-223.

Table des matières

Résumé	xi
Remerciements	xiii
Sommaire	xv
Liste des tableaux	xvii
Table des figures	xix
Introduction	1
Contexte	2
Agriculture de précision	6
La maîtrise de l'irrigation	7
Solutions matérielles	8
Solutions logicielles	9
WEENAT	11
La société	11
Capteurs	13
WEEDRIQ	17
Architecture	18
Climat et rayonnement global	18
SWC	20
SOIL	21
WATER BALANCE	22
PLANT	23
Fonctionnement global	26
Objectifs	28

I Problèmes d'optimisation et contributions dans la résolution de problème multimodal	35
1 Problèmes d'optimisation	37
1.1 Problèmes d'optimisation	38
1.2 Typologies d'optimisation	39
1.2.1 Classes d'optimisation	40
1.2.2 Optimisation hors ligne	41
1.2.3 Optimisation dynamique	42
1.2.4 Dérivabilité de la fonction objectif	42
1.2.5 Connaissance analytique du problème	44
1.2.6 Fonction multimodale	44
1.3 Méta heuristique	45
1.3.1 Stratégies d'évolution	46
2 Calibration de modèles paramétriques	59
2.1 Étalonnage automatique de modèle paramétrique	60
2.1.1 Contexte	60
2.1.2 Travaux similaires	61
2.1.3 Définition du problème d'optimisation	63
2.1.4 Protocole expérimental	64
2.1.5 Expériences et résultats	64
2.1.6 Synthèse	69
II Problèmes d'optimisation assistée par renforcement et contributions dans la résolution de problèmes multimodaux	73
3 Problèmes d'apprentissage par renforcement	75
3.1 Apprentissage par renforcement	76
3.2 Problème du bandit manchot	77
3.2.1 Upper Confidence Bound	77
3.2.2 Upper Confidence Tree	78
4 Problèmes d'optimisation assistée par renforcement	81
4.1 Stratégie de restart assistée par UCB	82
4.1.1 Motivations	82
4.1.2 Travaux similaires	83
4.1.3 UCB Random-restarts with Decreasing Step-size (URDS)	84
4.1.4 Fonctions de test	86
4.1.5 Analyse expérimentale	89

4.1.6 Synthèse	91
4.2 Stratégie de restart assistée par recherche arborescente	94
4.2.1 Motivations	94
4.2.2 Upper Confidence Tree for Random restarts with Decreasing Step-size	95
4.2.3 Analyse expérimentale	97
4.2.4 Fonctions de test	97
4.2.5 Résultats	100
4.2.6 Synthèse	104
III Apprentissage supervisé de phénomènes biologiques et application à l'agriculture	107
5 Apprentissage automatique	109
5.1 Principes	110
5.2 Algorithmes d'apprentissage supervisé	111
5.2.1 Les réseaux de neurones	113
5.2.2 Les forêts d'arbres décisionnels	116
5.2.3 Les Séparateurs à Vaste Marge (SVM)	118
5.3 Sélection de variables	119
6 Prédiction de phénomènes biologiques par apprentissage supervisé	125
6.1 Apprentissage supervisé de la teneur en eau	126
6.1.1 Motivations	126
6.1.2 Travaux similaires	128
6.1.3 Méthodologie	128
6.1.4 Préparation des données	131
6.1.5 Analyse expérimentale	133
6.1.6 Synthèse	151
Conclusion	157
Bibliographie	161
Table des matières	175

Résumé

Le sujet de la thèse porte sur une des thématiques du LISIC : la modélisation et la simulation de systèmes complexes, ainsi que sur l'optimisation et l'apprentissage automatique pour l'agronomie. Les objectifs de la thèse sont de répondre aux questions de pilotage de l'irrigation de la culture de pomme de terre par le développement d'outils d'aide à la décision à destination des exploitants agricoles. Le choix de cette culture est motivé par sa part importante dans la région des Haut-de-France.

Le manuscrit s'articule en 3 parties. La première partie traite de l'optimisation continue multimodale dans un contexte boîte noire. Il en suit une présentation d'une méthodologie d'étalonnage automatique de paramètres de modèle biologique grâce à une reformulation en un problème d'optimisation continue mono-objectif multimodale de type boîte noire. La pertinence de l'utilisation de l'analyse inverse comme méthodologie de paramétrage automatique de modèles de grandes dimensions est ensuite démontrée. La deuxième partie présente 2 nouveaux algorithmes *UCB Random with Decreasing Step-size* et *UCT Random with Decreasing Step-size*. Ce sont des algorithmes d'optimisation continue multimodale boîte noire dont le choix de la position initiale des individus est assisté par un algorithme d'apprentissage par renforcement. Les résultats montrent que ces algorithmes possèdent de meilleures performances que les algorithmes état de l'art *Quasi Random with Decreasing Step-Size*. Enfin la dernière partie est focalisée sur les principes et méthodes d'apprentissage automatique (machine learning). Une reformulation du problème de la prédiction à une semaine de la teneur en eau dans le sol en un problème d'apprentissage supervisé a permis le développement d'un nouvel outil d'aide à la décision pour répondre à la problématique du pilotage des cultures.

Mots clés : optimisation continue, optimisation boîte noire, optimisation multimodale, recherche arborescente, mcts, ucb, apprentissage par renforcement, apprentissage supervisé, sélection de caractéristiques

Abstract

The subject of this PhD concerns one of the LISIC themes: modelling and simulation of complex systems, as well as optimisation and automatic learning for agronomy. The objectives of the thesis are to answer the questions of irrigation management of the potato crop and the development of decision support tools for farmers. The choice of this crop is motivated by its important share in the Haut-de-France region. The manuscript is divided into 3 parts. The first part deals with continuous multimodal optimisation in a black box context. This is followed by a presentation of a methodology for the automatic calibration of biological model parameters through reformulation into a black box multimodal optimisation problem. The relevance of the use of inverse analysis as a methodology for automatic parameterisation of large models is then demonstrated. The second part presents 2 new algorithms *UCB Random with Decreasing Step-size* and *UCT Random with Decreasing Step-size*. these algorithms are designed for continuous multimodal black-box optimization whose choice of the position of the initial local search is assisted by a reinforcement learning algorithms. The results show that these algorithms have better performance than *(Quasi) Random with Decreasing Step-Size* algorithms. Finally, the last part focuses on machine learning principles and methods. A reformulation of the problem of predicting soil water content at one-week intervals into a supervised learning problem has enabled the development of a new decision support tool to respond to the problem of crop management.

Keywords: continuous optimization , black box optimization, multimodal optimization, tree based search, mcts, ucb, reinforcement learning, supervised learning, feature selection
