



Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique
Spécialité : Automatique

Présenté par :

Asmaa BENDAHMANE et Sarah HAMDI CHERIF

Thème

**Les techniques avancées du diagnostic
et détection de défauts**

Soutenu à huis clos, le 11 / 07 / 2021, devant le jury composé de :

Dr Fouad BOUKLI HACENE	MCA	ESSA. Tlemcen	Président
Dr Mustapha A. BRAHAMI	MCB	ESSA. Tlemcen	Directeur de mémoire
Dr Sidi Mohammed ABDI	MCB	ESSA. Tlemcen	Co-Directeur de mémoire
Pr Brahim CHERKI	PR	ESSA. Tlemcen	Examineur 1
Dr Ghouti ABDELLAOUI	MCB	ESSA. Tlemcen	Examineur 2
Dr Fouad MALIKI	MCB	ESSA. Tlemcen	Invité 1
Dr Sidi Mohammed KHEFIF	MCB	ESSA. Tlemcen	Invité 2

Année universitaire : 2020 / 2021

Dédicaces

*À nos parents,
À Wassim et Chahrazed,
À Naila et Abderahmane,
À nos amis et à notre promotion,
À Meriem et Meriem (elles se reconnaîtront),
Sans oublier Chouquette, Pichu et Kirby.*

Remerciements

Nous tenons à exprimer notre gratitude envers nos encadrants, M. Mustapha Anwar BRAHAMI et M. Sidi Mohammed ABDI pour leur patience et leur indulgence. Nous les remercions aussi pour l'aide qu'ils nous ont fourni, les connaissances qu'ils nous ont transmises et surtout pour leur constante disponibilité.

Nous remercions également M. Sidi Mohammed KHEFIF pour son aide et ses précieux conseils.

Nous remercions les membres du jury : Pr B. CHERKI, Dr F. BOUKLI-HACENE, et Dr G. ABDELLAOUI d'avoir accepté d'évaluer notre travail.

Pour finir, nous ne saurons oublier de trop remercier nos familles qui ont su nous encourager par leur enthousiasme pour ce projet même si ils n'y comprenaient pas grand chose.

Table des matières

Liste des abréviations

Introduction	1
1 Introduction au diagnostic	3
1 Introduction	3
2 Généralités sur le Diagnostic	3
2.1 Terminologie	4
2.2 Types de défauts	4
3 Notions sur la diagnosticabilité	5
3.1 Les méthodologies de diagnostic	5
4 Le diagnostic par l'approche FDI (fault detection and isolation)	6
4.1 Diagnostic à base de modèle	7
4.2 Diagnostic à base de modèle quantitatif	7
4.2.1 Les approches par espace de parité :	7
4.2.2 Les approches par estimation paramétrique :	7
4.2.3 Les approches basées sur les observateurs (estimation d'état) :	8
5 Diagnostic sans modèle	8
6 Conclusion	9
2 Méthodes d'apprentissage automatique	11
1 Introduction	11
2 Généralités sur l'apprentissage :	11
3 Les réseaux de neurones artificiels (RNAs)	12
4 Systèmes neuro-flous (Réseaux de neurones flous)	13
4.1 La logique floue :	13
4.2 Architecture d'ANFIS (Adaptive Neuro-Fuzzy Inference System)	14
5 Les machines à vecteur de support	15
5.1 SVM linéaire :	15
5.2 SVM non-linéaire :	16
6 Conclusion	17
3 Optimisation et Métaheuristiques	18
1 Introduction	18
2 Généralités sur l'optimisation	18
2.1 Définition	18
2.2 Terminologie	18
2.3 Types de problèmes d'Optimisation	19
2.4 Classes de difficulté	19
2.5 Types de méthodes d'optimisation	19
2.6 Les méthodes exactes	20

2.7	Les méthodes approchées	20
3	Les Métaheuristiques	21
3.1	Caractéristiques	21
3.2	Classification	21
4	Les métaheuristiques à solution unique	22
4.1	Recuit-simulé (Simulated Annealing)	22
4.2	La recherche avec Tabous (Tabu search)	24
5	Les métaheuristiques à population	27
5.1	Les essaims particuliers (PSO)	27
5.2	Les Algorithmes Génétiques (AG)	29
6	Conclusion	32
4	Application pour la détection de fuites	33
1	Introduction	33
2	Exemples d'applications pour la détection des fuites	33
3	Cas étudié	36
3.1	Introduction	36
4	Description du système	36
5	Description de l'Algorithme Génétique	37
5.1	Codage des solutions	37
5.2	Fonction Objectif	37
5.3	Opérateurs génétiques adoptés	37
5.4	Procédure de vérification de contraintes et de réparation des solutions infaisables	39
6	Expériences numériques et interprétations	39
6.1	Génération d'instances	39
6.2	Résultats obtenus	40
6.3	Interprétation des résultats :	41
6.4	Conclusion de l'analyse	42
7	Conclusion	45
	Conclusion générale	46

Liste des tableaux

4.1	Exemple des applications de l'apprentissage et de l'optimisation au diagnostic .	34
4.2	Exemple des applications de l'apprentissage et de l'optimisation au diagnostic (2)	35
4.3	Résultats obtenus par l'AG pour les petites distances	40
4.4	Résultats obtenus par l'AG pour les grandes distances	41

Table des figures

1.1	Différents types de défauts d'un système physique [32]	5
1.2	Une classification générale des méthodes de diagnostic [32]	6
2.1	Réprésentation biologique d'un neurone [?]	12
2.2	Modèle mathématique d'un neurone [?]	13
2.3	L'architecture ANFIS [?]	15
2.4	Cas où les données sont linéairement séparables [?]	16
2.5	Cas où les données sont non-linéairement séparables [?]	16
3.1	Classification des méthodes d'optimisation [25]	20
3.2	illustration des techniques du recuit et de la trempe [41]	22
3.3	Organigramme du Recuit-simulé [41]	24
3.4	Organigramme de la Liste taboue [41]	26
3.5	exemples de deux topologies des relations entre composés [41]	28
3.6	Stratégie de déplacement d'une particule [6]	28
3.7	Organigramme des Algorithmes Génétiques [41]	31
4.1	Évolution du temps d'exécution par rapport à la longueur des canalisations . . .	42
4.2	Pour une canalisation de longueur $L = 1km$ le temps de convergence est $t_{conv} = 6min$	43
4.3	Pour une canalisation de longueur $L = 1,5km$ le temps de convergence est $t_{conv} = 6min$	43
4.4	Pour une canalisation de longueur $L = 4km$ le temps de convergence est $t_{conv} = 13min$	44
4.5	Pour une canalisation de longueur $L = 8km$ le temps de convergence est $t_{conv} = 10min$	44
4.6	Pour une canalisation de longueur $L = 12km$ le temps de convergence est $t_{conv} = 23min$	45

Liste des abréviations

AG	:	Algorithme génétique
ANFIS	:	Adaptive Neuro-Fuzzy Inference System
EMV	:	Estimation du maximum de vraisemblance
MFP	:	Matched field processing
MH	:	Métaheuristiques
PSO	:	Particle Swarm Optimization
RE	:	Recherche exhaustive
RNA	:	Réseau de Neurones Artificiels
RNF	:	Réseau de Neurones Flous
RP	:	Régression Polynomiale
SVM	:	Support Vector Machine

Introduction

A correct diagnosis is three-fourths the
remedy.

Mahatma Ghandi

Avec l'avancée majeure de l'automatisation des processus industriels, il devient de plus en plus critique de veiller à leur bon fonctionnement. Les différents aspects de surveillance prennent plus d'importance, il est essentiel de détecter les défauts le plus rapidement possible, afin d'éviter une aggravation du dysfonctionnement et donc une panne des processus ; qui selon l'application peut résulter en des pertes monétaires, troubles environnementaux voire sanitaires. Il est aussi essentiel d'être précis dans son diagnostic, afin de ne pas malencontreusement empirer la situation et surtout de résoudre le problème dans un délai acceptable.

Le diagnostic automatique requiert la capacité d'identifier automatiquement les symptômes et de les relier correctement à leurs causes. Certaines méthodes peuvent s'avérer adaptées à certains systèmes alors qu'elles sont totalement inappropriées pour d'autres ; cela dépend souvent des données disponibles sur le système, de la complexité dudit processus et des objectifs fixés.

Avec l'avènement de l'intelligence artificielle, des nouvelles méthodes de détection de défauts sont mises au point. Prenant racine dans les principes de l'apprentissage et de l'optimisation, elles présentent un mode opératoire plus intuitif, intelligent et convenant à plusieurs niveaux de complexité. Là où les techniques conventionnelles du diagnostic peuvent se trouver limitées, incapable de s'adapter face à de nouvelles situations ; les méthodes avancées simulent des stratégies puissantes de résolution déjà existantes dans la nature, comme la plasticité neuronale pour les réseaux de neurones artificiels, la théorie de l'évolution pour les algorithmes évolutionnaires ou encore *la stigmurgie* pour l'intelligence à essaim. [11]

Jusqu'à présent, les essais et tentatives qui ont utilisé ces méthodes ont prouvé que le diagnostic des défauts peut grandement en bénéficier ; qu'elles soient utilisées comme méthode principale ou en coopération avec d'autres méthodes afin de les affiner. Les réseaux neuronaux peuvent faciliter l'identification des défauts par la mise en correspondance des modèles et l'apprentissage de nouveaux symptômes. La logique floue peut améliorer la prise de décision en prenant compte l'incertitude inhérente aux informations de diagnostic. Les algorithmes génétiques et autres métaheuristiques sont capables d'optimiser les modèles de diagnostic ainsi que le processus de diagnostic lui-même et présentent souvent un gain de temps important, ce qui reste un objectif essentiel dans le cadre d'un diagnostic. [38]

Ce travail a pour but d'exposer les mécanismes intelligents de ces méthodes, et de montrer leur intérêt dans le cadre de la détection des défauts. Pour ce faire, nous présenterons les techniques avancées du diagnostic les plus utilisées en ce moment, divisées en deux classes, celles qui se basent sur l'apprentissage et celles qui se basent sur l'optimisation. Puis, nous nous concentre-

rons sur un problème en particulier : la détection des fuites dans une canalisation d'eau. [\[44\]](#), [\[43\]](#)

Dans le premier chapitre, nous présentons des notions du diagnostic avec un intérêt particulier pour le diagnostic à base de modèle. Le deuxième chapitre est dédié aux méthodes d'apprentissage utilisées fréquemment pour la détection de fautes, suivi par le troisième chapitre qui concerne les métaheuristiques. Finalement, dans le quatrième chapitre, nous présentons l'optimisation d'une méthode de détection de fuites via l'utilisation d'un algorithme génétique et exposons les résultats que nous avons obtenus.

Chapitre 1

Introduction au diagnostic

1 Introduction

Au cours des deux dernières décennies, la recherche sur le diagnostic a gagné en considération dans le monde entier. Cette évolution a été et est toujours principalement stimulée par la tendance de l'automatisation des systèmes complexes.

Le diagnostic des pannes est en train de passer d'un processus manuel, dans lequel les opérateurs humains vérifient et évaluent régulièrement l'état d'un système, vers un processus automatisé, dans lequel des algorithmes avancés sont utilisés pour détecter et identifier les défauts du système en fonction de données collectées en temps réel. [39]

Ce chapitre a pour but d'introduire des notions importantes du diagnostic.

2 Généralités sur le Diagnostic

Les définitions du diagnostic et de la détection des fautes sont nombreuses, nous en avons choisi une qui nous semble complète :

« Comme dans son acceptation médicale, le diagnostic des systèmes automatiques consiste à remonter des symptômes perçus vers les causes. Il est également défini comme l'ensemble des actions mises en œuvre en vue de détecter, localiser et identifier tout phénomène anormal sur un système à partir de la connaissance disponible sur ce dernier. La connaissance disponible inclut, non seulement, les sorties du contrôleur (qui a pour but d'améliorer les performances du système), les mesures fournies par les capteurs mais aussi, le plus souvent, des informations sur la structure et/ou un modèle du système. » [29]

Comme cité plus haut, les principales étapes suivies pour un bon diagnostic sont donc la détection, la localisation puis l'identification de tout phénomène anormal. On entend par ce dernier terme les différents dysfonctionnements qui peuvent apparaître durant le cycle de vie d'un processus : des défauts de fabrications en passant par les défaillances abruptes jusqu'aux usures causées par la vieillesse des matériaux ou le mauvais entretien.

2.1 Terminologie

À travers la littérature un certain nombre de terminologies différentes sont utilisées. Dans ce rapport nous userons de celles trouvées dans [19].

Défaut : est un écart entre la caractéristique observée et la caractéristique de référence. Un défaut peut conduire à une défaillance du système tout comme il peut exister sans altérer son fonctionnement général.

Défaillance : est une conséquence possible d'une altération importante des caractéristiques. Elle correspond à l'incapacité du système à réaliser la fonction souhaitée. Elle peut être passagère ou permanente.

Panne : est l'interruption permanente du système à réaliser la fonction souhaitée.

Résidus : Indicateurs d'anomalie. Expriment la différence entre les valeurs théoriques et celles mesurées. Présentent une valeur nulle lors du fonctionnement normal.

Monitoring : c'est la surveillance en temps réel du système, il permet la détection précoce des défauts.

Supervision : c'est la surveillance du système et la prise de décisions rapide dans le cas de l'existence d'un défaut.

Détection de défaut : il s'agit de détecter rapidement les anomalies et de prendre une décision rapide quant à l'existence ou non d'un défaut dans le système.

Isolation des défauts : il s'agit de déterminer le composant qui cause problème, par exemple, quel capteur ou actionneur est devenu défectueux.

Identification du défaut : il s'agit d'estimer la taille, le type ou la nature du défaut.

Diagnostic : La détermination du type, de la taille, de la localisation et de l'instant d'occurrence d'un défaut ; il suit la détection de défauts et inclut l'isolation et l'identification.

2.2 Types de défauts

Dans ce qui suit, nous décrirons les différentes sources de défauts dans un système [34].

Défauts sur les capteurs :

Les capteurs représentent l'interface de sortie d'un système, et transmettent des informations sur le comportement de ce dernier. Par conséquent, les défauts du capteur peuvent entraîner une dégradation des performances du processus et compromettre l'intégrité des données ce qui revient à fausser la prise de décisions.

Les capteurs sont utilisés soit pour mesurer directement les états du système, soit pour générer des estimations d'état pour la loi de contrôle. Ainsi, la présence de défauts dans les capteurs peut détériorer les estimations d'état et par conséquent entraîner un diagnostic inexact.

Défaut sur les actionneurs :

Dans de nombreux systèmes les signaux de commande du contrôleur ne peuvent pas être directement appliqués au système. Des actionneurs sont nécessaires pour les transformer en signaux d'actionnement appropriés. Cependant, l'apparition d'anomalies dans cet organe peut amener à une perte de contrôle totale.

Défauts sur le processus :

Le défaut sur le processus survient généralement lorsque certaines conditions changent dans le système, rendant invalide l'équation dynamique nominale du système.

La figure ci-dessous représente un récapitulatif des types de défauts présentés.

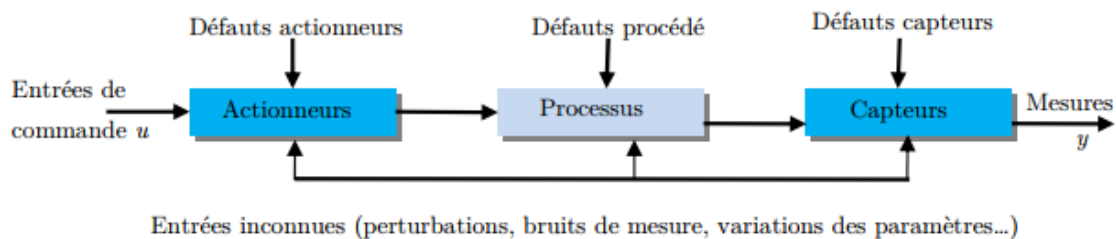


FIGURE 1.1 – Différents types de défauts d'un système physique [27]

3 Notions sur la diagnosticabilité

La notion de diagnosticabilité exprime un problème fondamental lors de la conception d'un système de diagnostic. Il s'agit de s'assurer qu'en fonctionnement, le diagnostiqueur sera toujours capable de diagnostiquer sans aucune ambiguïté un défaut préalablement répertorié. Cela signifie d'une part qu'il sera toujours capable de détecter un tel défaut préalablement répertorié lorsqu'il apparaît, et d'autre part qu'il sera toujours capable de l'identifier sans ambiguïté par rapport aux autres défauts préalablement répertoriés [2].

3.1 Les méthodologies de diagnostic

En ce qui concerne les méthodes utilisées, une distinction est faite entre l'approche basée sur un modèle, l'approche sans modèle et l'approche hybride.

Les méthodes avec modèles [20] (utilisées par la communauté *Fault detection and diagnosis* (FDI) s'appuient sur une description qualitative ou quantitative des relations entre les données de surveillance et la santé du système, tandis que les méthodes sans modèle utilisent des données et des techniques issues de l'apprentissage automatique ou de la reconnaissance de formes, quant à l'approche hybrides, elles utilisent une combinaison des méthodes mentionnées.

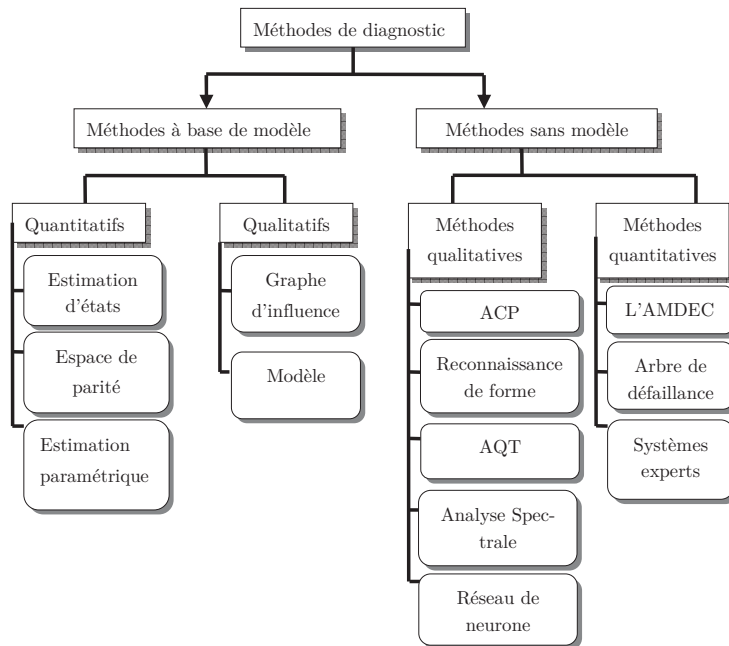


FIGURE 1.2 – Une classification générale des méthodes de diagnostic [27]

4 Le diagnostic par l’approche FDI (fault detection and isolation)

Plusieurs méthodes ont été développées pour résoudre les différents problèmes de diagnostic de défaut telle que l’approche de la détection et isolation des fautes (FDI), et celle de la tolérance aux fautes.

La philosophie derrière l’approche FDI stipule qu’un défaut entraînera des modifications de certains paramètres physiques et mesures, ce qui à son tour entraînera une modification au niveau du modèle ou des états [39].

En surveillant les paramètres, il est alors possible de détecter et d’isoler chaque faute. Ce qui revient à dire que l’objectif de la FDI est d’identifier les défauts naissants le plus tôt possible afin de prévenir les défaillances, à travers la génération de résidus et l’isolement des défauts.

Dans l’approche FDI, la détection des défauts se fait en trois étapes [27] :

- Détecter l’existence d’un défaut.
- Localiser la faute.
- Caractériser l’amplitude de la faute (l’identifier).

4.1 Diagnostic à base de modèle

L'approche principale consiste à modéliser les défauts comme des écarts entre les paramètres.

Pour illustrer le concept, considérons un système avec un modèle $M(\theta)$, où θ est un paramètre ayant la valeur nominale 0 (c'est-à-dire sans défaut). En utilisant des techniques générales d'estimation des paramètres, une estimation θ peut être formée puis comparée à θ_0 . Si θ s'écarte trop de θ_0 , alors la conclusion est qu'un défaut s'est produit.

La limitation est que lorsque le nombre de défauts diagnostiqués augmente, le vecteur de paramètre augmente en dimension ce qui donne un problème complexe.

En règle générale ces méthodes sont réparties selon le type de modèle, nous pouvons donc discerner deux branches : Les méthodes basées sur un modèle quantitatif où on retrouve les observateurs (estimateurs, filtres), l'espace de parité, et l'identification et estimation des paramètres ; et les méthodes basées sur un modèle qualitative constituées de bond graphes et de graphes biparties.

4.2 Diagnostic à base de modèle quantitatif

Selon P.Frank [12], ces méthodes sont divisées en trois groupes : les approches par espace de parité, les approches par estimation paramétrique et les approches basées sur les observateurs.

Ces méthodes reposent sur une comparaison du comportement du système avec le comportement du modèle quantitatif établi [37].

4.2.1 Les approches par espace de parité :

Le but de cette méthode est d'assurer la compatibilité entre le modèle du processus et les mesures des capteurs et entrées connues (signal de commande, consigne).

Les équations de parité offrent une approche simple et directe pour construire des résidus qui indiquent si des systèmes sont défectueux. Basée sur les écarts des états estimés du système, cette méthode vérifie la cohérence entre les relations mathématiques du système à surveiller et les mesures des capteurs et des entrées connues. Cette approche consiste à réaliser une redondance analytique qui peut être appliquée aux systèmes déterministes, stochastiques ainsi qu'aux cas dynamiques ou statiques. L'ensemble des valeurs que peuvent prendre les résidus forme l'espace de parité [27].

4.2.2 Les approches par estimation paramétrique :

Cette approche permet d'estimer les paramètres du système à l'aide des mesures d'entrées et de sorties. Elle est développée sur la base des techniques d'identification du système [20].

L'idée est d'estimer les paramètres du processus réel à plusieurs reprises en utilisant des méthodes d'estimation bien connues (par exemple : EMV), les résultats sont alors comparés aux paramètres du modèle de référence, obtenus initialement dans le cas sans défaut.

Il convient de souligner qu'une limitation des approches basées sur l'estimation des paramètres réside en la quantité d'informations que le signal d'entrée transmet afin d'estimer avec

efficacité les paramètres.

L'avantage est que les estimations des paramètres sont en fait liées à ce que l'on appelle les coefficients de processus qui sont à leur tour liés aux défauts. Des exemples de coefficients de processus incluent : rigidité, longueur, masse, résistance, vitesse, . . . etc. Une fois estimés, ils sont ensuite comparés à des valeurs de référence prédéfinies (par exemple un seuil fixe). L'isolement des défauts est ensuite mis en œuvre sur la base de la relation entre les défauts et les coefficients de processus.

La procédure de base pour réaliser une estimation des paramètres est la suivante :

- Établir le modèle de processus à l'aide des relations physiques.
- Déterminer la relation entre les coefficients du modèle et les paramètres physiques du procédé.
- Estimer les coefficients du modèle.
- Calculer les paramètres physiques nominaux du processus.
- Déterminer les changements de paramètres qui se produisent pour les différents cas de faute.

4.2.3 Les approches basées sur les observateurs (estimation d'état) :

Les approches basées sur les observateurs ou les filtres nécessitent un modèle pour la détection et le diagnostic des défauts. Les sorties du système sont estimées à partir des mesures en utilisant soit l'observateur de Luenberger dans le cadre déterministe ou le filtre de Kalman dans un cadre stochastique. Ensuite, l'erreur d'estimation est assimilée à un résidu.

La principale préoccupation de l'approche basés sur les observateurs est la génération d'un ensemble de résidus qui détectent et identifient les différents défauts.

L'idée de base est que dans un cas de fonctionnement normal, les observateurs suivent le processus. Si un défaut survient, ils s'écarteront du processus ainsi un vecteur résidu est construit, ce dernier agit comme un indicateur de comportement du système et définit l'écart entre la sortie estimée et la sortie mesurée. [27].

L'un des inconvénients de l'approche par observateur est que la connaissance du processus est requise. Cependant l'observateur fournit une estimation, une convergence et une détection de défauts assez précise.

5 Diagnostic sans modèle

Le diagnostic sans modèle intervient lorsqu'aucun modèle explicite n'est disponible. La connaissance du système se résume à des mesures en temps réel, éventuellement complétées par l'historique du processus.

Nous pouvons discerner deux classes de ce type d'approche : Les méthodes quantitatives et les méthodes qualitatives [27].

Concernant *les méthodes quantitatives*, les systèmes experts sont les plus populaires.

Ce sont des systèmes très spécialisés qui résolvent des problèmes dans un domaine d'expertise restreint. Ils sont généralement constitués d'une base de données et d'un moteur d'inférence.

Opérant à partir d'un ensemble de symptômes, ils en déduisent toutes les conclusions possibles, élaborent de nouvelles hypothèses et approfondissent leur diagnostic, en exploitant des informations supplémentaires collectées sur le système à diagnostiquer. [39].

En somme, les systèmes experts sont des systèmes informatiques qui cherchent à émuler la capacité de la prise de décision humaine.

Les méthodes qualitatives à leur tour exploitent une base de données et requièrent la connaissance des divers modes de fonctionnement du processus.

Deux stratégies principales de cette approche peuvent être adoptées :

La première est la classification

Elle implique la construction des classes à partir d'une base de données, soit de manière supervisée (c'est-à-dire avec l'aide d'un expert), ou de manière non supervisée. Un classificateur est ensuite entraîné pour effectuer la classification des variables nouvellement mesurées comme représentatives d'une ou plusieurs défaillances.

Cette approche nécessite un nombre assez conséquent d'exemples de fonctionnement nominal et défaillant du système, pour permettre au réseau d'apprendre.

L'apprentissage est réalisé comme suit : les exemples sont considérés comme des entrées et les diagnostics correspondants comme des sorties, ainsi le réseau apprendra à relier les exemples au diagnostic adéquat. Après cette étape, le réseau acquiert une certaine robustesse par rapport aux déformations des signaux par le bruit.

Plusieurs algorithmes intelligents, comme les réseaux de neurones, sont utilisés pour cette tâche, une présentation plus détaillée sera faite dans le chapitre suivant.

La deuxième stratégie est la régression

Il s'agit de construire un modèle statistique qui utilise la redondance de l'historique du processus afin de prédire les valeurs des nouvelles variables et de générer des résidus en comparant les prédictions aux valeurs mesurées.

La Régression polynomiale (RP) est une méthode d'approximation fréquemment employée dans le diagnostic et la détection de défauts.

Son principe est le suivant : en considérant un ensemble de données d'apprentissage d'entrée-sortie, l'estimateur est supposé être une fonction polynomiale d'un certain degré. Les coefficients polynomiaux (paramètres du modèle) sont estimés par la méthode des moindres carrés ; la somme des erreurs quadratiques (entre les données et les prédictions du modèle) est minimisée afin de trouver les meilleurs paramètres [1].

6 Conclusion

Dans ce chapitre nous avons introduit d'importantes notions du diagnostic et de la détection de défauts, tout en insistant sur les méthodes à base de modèle.

Ensuite nous avons brièvement passé en revue quelques approches basées sur l'apprentissage automatique (Machine Learning). Ces méthodes sont considérées plus attrayantes et plus efficaces pour acquérir des connaissances sur le système à partir d'une grande quantité de données empiriques.

Une des exigences fonctionnelles d'un diagnostiqueur est **la rapidité du diagnostic**, d'où la nécessité d'utiliser un algorithme d'optimisation de type méta-heuristique.

Chapitre 2

Méthodes d'apprentissage automatique

1 Introduction

Dans la littérature, il existe deux approches principales pour le diagnostic et la détection des défauts : les approches basées sur un modèle analytique ou physique et les approches qui s'appuient uniquement sur les observations du système. L'utilisation des techniques de diagnostic avec modèles semble être une tâche ardue et coûteuse, car il existe plusieurs applications industrielles où un modèle est difficile à obtenir en raison d'une complexité accrue du processus.

Une approche qui utilise un modèle physique est basée sur la connaissance de l'ingénierie ou des considérations théoriques, par exemple le bilan énergétique, ou les équations de transport de chaleur et de masse. En revanche, une approche basée sur les données statistiques dépend des quantités d'informations fournies par l'historique du processus ou d'un modèle empirique. Par conséquent, dans ce chapitre nous passerons en revue les techniques d'apprentissage et de classification les plus importantes et les plus populaires du diagnostic sans modèle. Parmi ces techniques, on retrouve les machines à vecteur de support (SVM), les réseaux de neurones artificiels (RNN) et les systèmes neuro-flous (RNF).

2 Généralités sur l'apprentissage :

L'apprentissage est la capacité de généraliser une situation/modèle à partir d'observations particulières.

L'apprentissage automatique (Machine learning) : est une technique de programmation informatique qui utilise des probabilités statistiques pour donner aux ordinateurs la capacité d'apprendre par eux-mêmes sans programmation explicite.

Classification : C'est le processus de reconnaissances des classes à partir des caractéristiques des objets observés.

Types d'apprentissages : L'IA repose sur des algorithmes d'apprentissage automatique, qui à leur tour sont fondamentalement ancrés dans l'optimisation [8] :

L'apprentissage supervisé : l'algorithme est présenté avec un ensemble de données d'entraînement. Ainsi, des exemples d'entrée et de sortie d'un modèle sont explicitement donnés, et des méthodes de régression sont utilisées pour trouver le meilleur modèle pour les données

étiquetées, via l'optimisation. Ce modèle est ensuite utilisé pour la prédiction et la classification.

L'apprentissage semi-supervisé : dans lequel une information incomplète est donnée de sorte qu'une partie des entrées/sorties manquent.

L'apprentissage non supervisé : aucune étiquette n'est donnée pour ce type d'algorithme. Ainsi, il doit trouver des schémas/structures (patterns) dans les données afin de déterminer comment regrouper ces dernière et générer des étiquettes pour prédire et classer de nouvelles données, le but de ce type d'apprentissage est de permettre a la machine d'inventer ses propre classes.

3 Les réseaux de neurones artificiels (RNAs)

Les RNAs sont des techniques d'apprentissage basées sur un certain nombre de neurones interconnectés, ce qui veut dire que la sortie d'un neurone peut être l'entrée d'un autre, les poids w sont obtenus par une procédure d'apprentissage qui peut s'effectuer de trois manières : supervisé, non supervisé, ou semi-supervisé [8], ainsi la capacité de généralisation du réseau est testée une fois que l'étape de l'apprentissage est achevée.

En règle générale, les RNAs se présentent sous cette forme - avec trois composants- : une couche d'entrée, une couche cachée et une couche de sortie. Les unités de la couche cachée sont appelées ainsi car leurs valeurs ne sont pas observées.

L'équivalence entre un neurone biologique et un neurone artificiel est représentée dans la figure 2.2 qui suit (a noter que cela reste une approximation de la cellule neuronal, le but des RNAs n'étant pas de décrire le fonctionnement du cerveau).

Dans ce modèle, un neurone reçoit l'échantillon d'information d'entrée x , avec x_i indiquant le ième élément du vecteur x , les neurones sont connectés par l'intermédiaire de leurs axones, les synapses contiennent les poids w_i à appliquer aux entrées. Puis, grâce aux dendrites, les contributions des différentes entrées sont transmises au corps cellulaire où la somme pondérée des entrées est calculée (en plus de la valeur du biais b qui est ajoutée), enfin la fonction d'activation f est appliquée à la somme afin d'avoir le résultat.

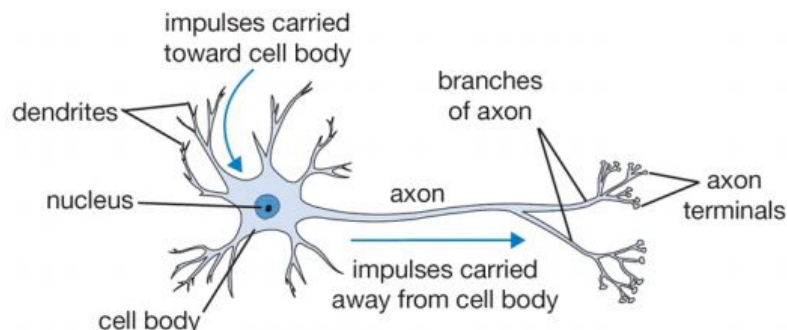


FIGURE 2.1 – Représentation biologique d'un neurone [31]

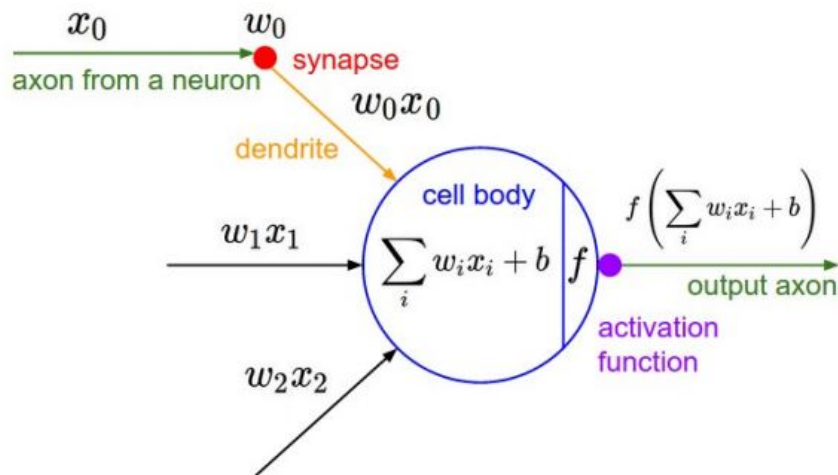


FIGURE 2.2 – Modèle mathématique d'un neurone [31]

L'objectif des RNAs est d'implémenter des algorithmes qui tentent d'atteindre une performance neurologique, comme l'apprentissage et la généralisation à partir de situations déjà rencontrées. Dans un problème de diagnostic, ils sont employés soit comme :

- Un modèle du système à surveiller qui génère un résidu d'erreur entre les observations et les prédictions.
- Classificateur.
- Evalueur de résidus.

La capacité des RNAs à modéliser des problèmes complexes non linéaires avec un haut degré de précision les ont rendus applicables aux problèmes de FDI. Cependant, leur aspect boîte noire rend difficile l'interprétation du modèle. De plus, ils sont sujets au sur-apprentissage et nécessitent des bases de données diversifiées pour éviter ce problème.

Leur implémentation dans de grands systèmes reste discutable, il est préférable de les utiliser comme aide pour d'autres techniques dans la génération des erreurs.

4 Systèmes neuro-flous (Réseaux de neurones flous)

Nous commençons par introduire des concepts précurseurs qui sont nécessaires à la compréhension des systèmes neuro-flous.

4.1 La logique floue :

Les problèmes de la vie réelle ont souvent tendance à ne pas être de nature discrète mais continue. Parfois, définir une catégorie bien précise pour un objet est impossible. Pour résoudre ce problème, en 1965 Lofti Zadeh a introduit la théorie des ensembles flous, où l'appartenance à un élément n'est plus un état binaire mais une valeur continue. Par exemple, au lieu de dire que \mathbf{a} est un élément de l'ensemble \mathbf{A} et non un élément de \mathbf{B} , on peut dire que \mathbf{a} est un élément d'un ensemble flou par 0,66 degré de vérité (ou d'appartenance) et est un élément de \mathbf{B} par 0,31 degré de vérité.

En somme la logique floue utilise des ensembles flous pour représenter des valeurs de vérité et par conséquent elle fournit un moyen efficace de modélisation de la connaissance humaine

dans un langage mathématique en utilisant des règles qualitatives qui suivent cette forme :

Si <prémisse> alors <conséquence>.

Un modèle neuro-flou vient de la combinaison de la logique floue avec des réseaux de neurones pour donner un système de données et d'inférences afin de décrire un objet ou un processus. Il existe différentes méthodes de combinaison des réseaux des neurones et de la logique floue on trouve :

- L'utilisation des techniques d'apprentissages supervisés ou non supervisés pour construire un modèle flou à base de règles maquette.
- L'exploitation des technique d'apprentissage pour partitionner l'espace d'entrée.

Dans le contexte du diagnostic (FDI), la logique floue a été appliquée principalement aux tâches suivantes :

- La détection de défauts : Les procédés industriels sont caractérisés par des variables continues (symptômes) ces dernières sont sujettes aux erreurs de mesure et au bruit. Par conséquent, des mesures fiables sont difficiles à définir. La logique floue fournit une bonne solution à ce problème, en représentant le signal avec des variables linguistiques.
- Le diagnostic des défauts : Le diagnostic des défauts dans les processus dynamiques est toujours approximatif, car les valeurs du signal mesurées ne sont connues qu'avec un certain degré de précision donc un système basé sur une inférence floue avec des règles SI-ALORS peut fournir une solution à ce problème.

4.2 Architecture d'ANFIS (Adaptive Neuro-Fuzzy Inference System)

L'un des premiers systèmes neuro-fuzzy a été introduit par Jang en 1993, cette architecture utilise le système d'inférence Takagi-Sugeno-Kang.

La figure 2.3 montre l'architecture ANFIS qui est composée de six couches ou x et y désignent les variables d'entrée et z désigne la variable de sortie. Chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. La première couche contient deux nœuds avec pour entrée x et y , la deuxième couche lie les valeurs d'entrée aux fonctions d'appartenance. Les nœuds de la troisième couche correspondent aux règles floues, leurs valeurs de sortie sont appelées les poids normalisés tandis que les sorties de la quatrième couche déterminent les paramètres de la partie conséquence des règles. La défuzzification se produit dans la cinquième couche et les nœuds de sortie de cette couche sont additionnés lors du passage à la sixième couche [40].

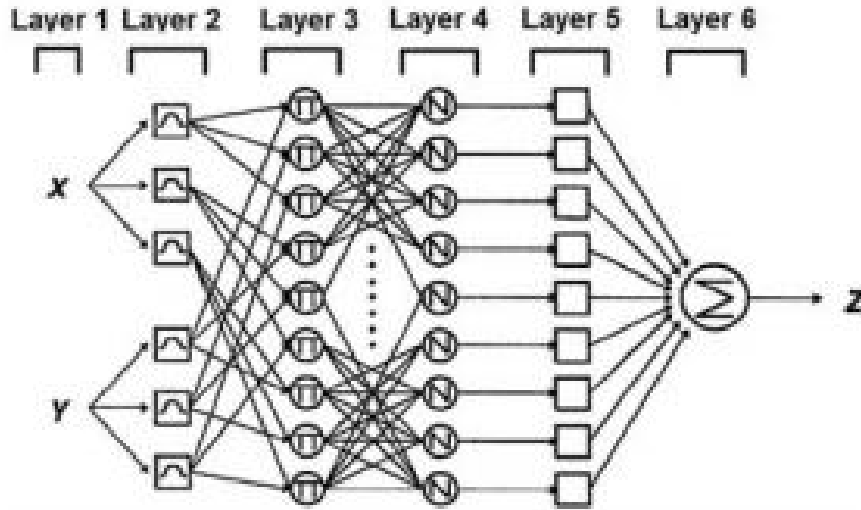


FIGURE 2.3 – L'architecture ANFIS [40]

L'apprentissage itératif de l'ANFIS est composé de deux étapes. Dans la première étape, les paramètres des fonctions liées à la cinquième couche sont réglés par la méthode des moindres carrés.

Au cours de la deuxième étape les paramètres des fonctions de prémisse (située dans la deuxième couche) sont ajustés par une rétropropagation. Ces deux étapes sont répétées itérativement afin de former le système.

5 Les machines à vecteur de support

Inventé par Vladimir N. Vapnik et Chervonenkis en 1963, l'algorithme des machines à vecteur de support a évolué à partir des méthodes d'apprentissage statistiques ; il reprend le principe des réseaux de neurones et peut être considéré comme une extension du perceptron (Rosenblatt en 1958), ce dernier garantit l'existence d'un hyperplan quelconque tandis que l'algorithme SVM construit des hyperplans optimaux. En règle générale il existe des SVM linéaires conçus pour des problèmes linéairement séparables et des SVM non linéaires qui règlent des problèmes non linéairement séparables.

5.1 SVM linéaire :

L'idée de cette méthode est de construire un hyperplan tel que :

$$w \cdot x + b = 0 \quad (2.1)$$

Où w est un vecteur et b est une constante.

La figure 2.4 montre deux potentiels hyperplans divisant un ensemble de données. Chacun a une valeur différente de w et de la constante b . L'objectif de SVM est de non seulement optimiser une ligne de décision qui fait le moins d'erreurs d'étiquetage des données, mais aussi de maximiser également la marge entre les données. Les vecteurs qui déterminent les limites de cette dernière sont appelées les vecteurs supports(ceux qui se rapprochent de la region grise, figure 2.4) [8].

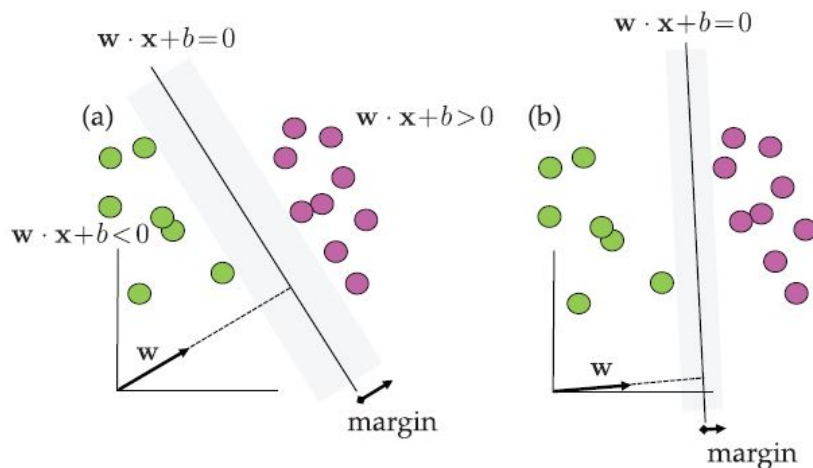


FIGURE 2.4 – Cas où les données sont linéairement séparables [8]

5.2 SVM non-linéaire :

Le principe des SVM linéaires est repris, l'astuce du noyau (kernel trick) est introduite pour pallier à la non-linéarité.

Les hyperplans sont construits grâce à des fonctions à noyaux (kernel functions). Ces dernières sont définies dans l'espace d'entrée et fournissent un moyen de manipuler les données comme si elles étaient projetées dans une dimension supérieure (aussi appelé espace des caractéristiques) mais en opérant dans leur espace d'origine (comme illustrer dans la figure 3.7).

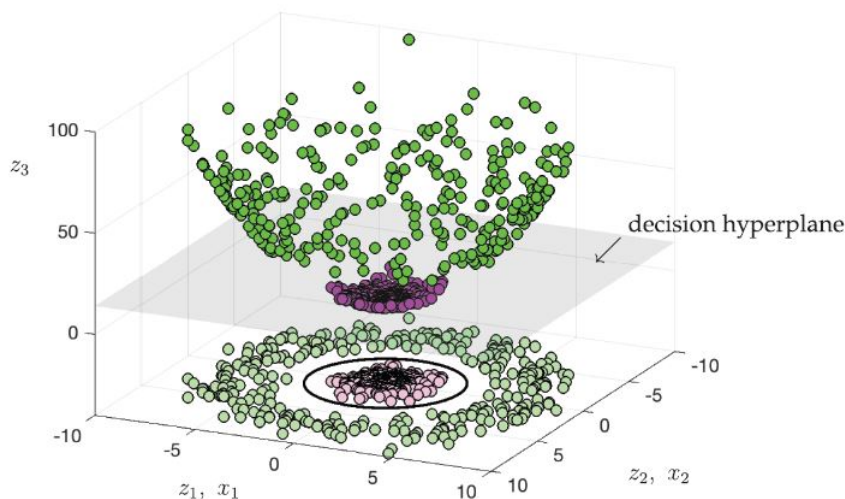


FIGURE 2.5 – Cas où les données sont non-linéairement séparables [8]

En conséquent ces fonctions constituent la richesse des SVM, car on peut toujours chercher de nouveaux noyaux qui peuvent être mieux adaptés a la tâche qu'on veut accomplir.

A noter que leurs applications au diagnostic ne diffèrent pas vraiment de celles présentées dans la section des réseaux de neurones. Pour la détection des défauts, ce classificateur n'aura pas besoin qu'on lui donne des règles car il construira automatiquement une frontière de décision à partir d'une base d'apprentissage constituée des données du fonctionnement normal et défectueux, lui permettant par la suite de décider si une classe représente un défaut ou pas.

6 Conclusion

Une fois mises en œuvre, les approches classiques du diagnostic résultent en un niveau de performance fixe, c'est-à-dire qu'il est impossible d'améliorer ces méthodes en utilisant des expériences antérieures. Cependant les approches d'apprentissage automatique présentées dans ce chapitre ont pour but d'améliorer continuellement les performances du système à l'aide des différentes données qui remplace le modèle mathématique utilisée lors de l'apprentissage.

Toutefois une base de données mal conçue ou qui présente des informations manquantes peut fausser les résultats et conduire a une étude erronée.

Chapitre 3

Optimisation et Métaheuristiques

1 Introduction

Les modèles mathématiques ou physiques utilisés pour le diagnostic des défauts sont souvent incomplets et imprécis ; résultant en des problèmes difficiles qui ne peuvent pas être résolus par des méthodes classiques, et rarement en un temps raisonnable. Les métaheuristiques sont des méthodes d'optimisation non exactes qui utilisent des stratégies stochastiques afin de parvenir à un résultat approché de la solution optimale en contrepartie d'un temps de calcul moins excessif.

Dans ce chapitre, nous introduisons d'importantes notions d'optimisation, puis nous présentons quatre métaheuristiques et leurs principes de fonctionnements afin de se familiariser avec les stratégies utilisées par ces méthodes pour résoudre les problèmes d'optimisation.

2 Généralités sur l'optimisation

L'optimisation est une branche des mathématiques. Il s'agit de modéliser un problème concret, l'analyser puis le résoudre de façon analytique ou numérique.

2.1 Définition

On appelle problème d'optimisation en dimension fini, tout problème de recherche des points x' dans \mathbb{R}^n qui vérifient :

$$f(x') = \min_{x \in C} f(x)$$

ou

$$f(x') = \max_{x \in C} f(x)$$

C'est donc la recherche des points optimaux, qu'ils soient minimums ou maximums, locaux ou globaux selon le problème considéré et les contraintes qui lui sont propres.

2.2 Terminologie

$f(x)$: fonction objectif à optimiser.

x' : solution du problème.

C : ensemble des contraintes. S'il n'existe pas de contraintes alors $C = \mathbb{R}^n$.

2.3 Types de problèmes d'Optimisation

Il existe deux grandes familles : l'optimisation continue et l'optimisation discrète. Cette classification dépend de la nature des variables impliquées dans la fonction objectif.

- Optimisation continue : si l'ensemble des valeurs acceptables des variables est continu et donc infini.
- Optimisation discrète : Si l'ensemble des valeurs acceptables des variables est fini ou dénombrable.

A noter qu'il est possible d'avoir des problèmes mixtes puisque la limitation des variables est souvent associée aux contraintes imposées par le problème. [33]

2.4 Classes de difficulté

Les méthodes d'optimisation sont nombreuses et diverses ; chacune s'adaptant à un niveau de difficulté. Il existe différentes façons de classifier la complexité des problèmes d'optimisation, nous nous contenterons de présenter celles concernant les problèmes de décision. [35]

Un problème d'optimisation peut être assimilé à un problème de décision qui tranche si oui ou non il existe une meilleure solution que l'actuelle. Ils partagent donc la même complexité, qui est déterminée par les classes suivantes :

Classe P regroupe les problèmes solvables par un algorithme déterministe de résolution polynomiale.

Classe NP regroupe les problèmes solvables par un algorithme non-déterministe de résolution polynomiale. S'il est possible de résoudre lesdits problèmes dans un temps polynomial on parle de NP-complets, dans le cas inverse, on parle de problèmes NP-difficiles.

A savoir que les problèmes NP-difficiles sont NP-complets mais la réciproque est fausse.

2.5 Types de méthodes d'optimisation

Il existe plusieurs classifications, celle de Dréo [21] suit cette hiérarchie :

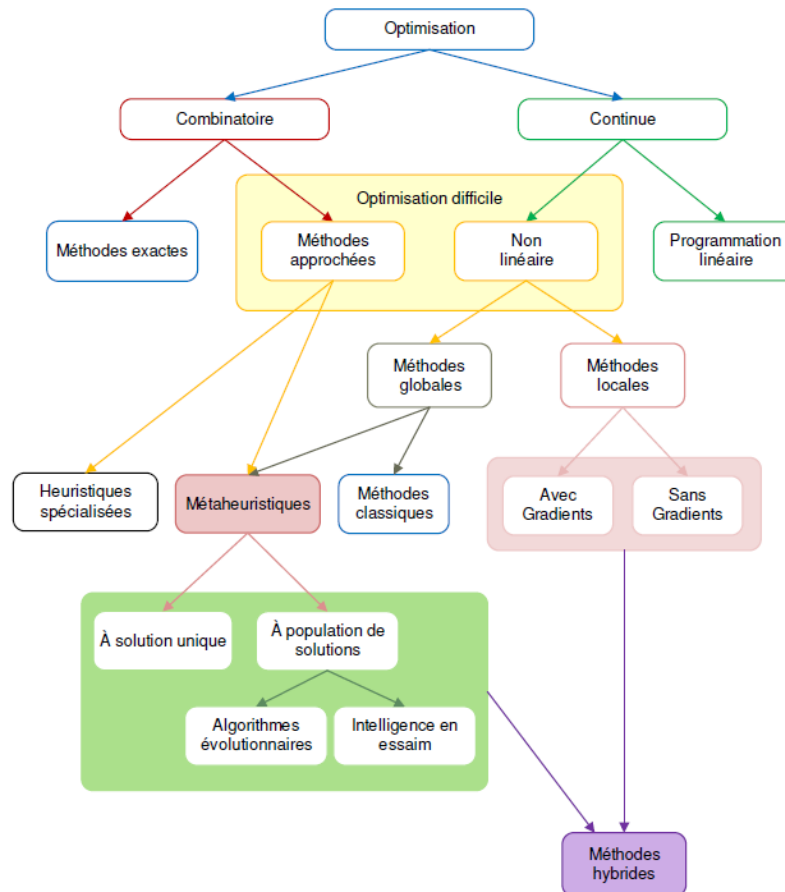


FIGURE 3.1 – Classification des méthodes d’optimisation [21]

Les méthodes de l’optimisation continue sont composées de la programmation linéaire et non linéaire; tandis que les méthodes de l’optimisation discrète (combinatoire) sont divisées entre les méthodes exactes et les méthodes approchées. On trouve en commun entre les deux branches, les métaheuristiques.

Nous détaillons ci-dessous les techniques de l’optimisation combinatoire.

2.6 Les méthodes exactes

Ces méthodes procèdent à l’exploration complète de l’espace des solutions. Elles finissent toujours par trouver l’optimum ou bien par réfuter son existence. Ces méthodes risquent de se heurter au problème de l’explosion combinatoire dans le cas où la taille de l’espace de recherche est trop importante; autrement dit, le temps de résolution devient excessivement long.

2.7 Les méthodes approchées

Afin de contrer l’explosion combinatoire, ces méthodes explorent l’espace de recherche de façon intuitive et pas forcément en son entièreté. Il est possible qu’en négligeant une partie des solutions, elles ratent l’optimum; cependant, elles arrivent souvent à un résultat qui est suffisamment proche et cela en un temps raisonnable. Elles sont divisées en deux familles : les heuristiques et les métaheuristiques.

Les Heuristiques d'origine grec, le mot heuristique signifie « je trouve », connu plus largement grâce au fameux « Eureka » d'Archimède. Une heuristique est un algorithme qui repère une solution acceptable en un temps raisonnable – donc, polynomial – mais qui ne garantit pas l'optimalité de ladite solution. Sa façon d'explorer l'espace de recherche est construite selon les contraintes du problème étudié, elle lui est donc propre et ne peut pas être généralisée pour d'autres cas. [5]

Les Métaheuristiques c'est là qu'interviennent les métaheuristiques. Méta signifie « au-delà » en grec, et effectivement ces algorithmes ont une structure plus globale que les heuristiques. Au lieu de suivre des contraintes spécifiques, la diversification des solutions est assurée par des « opérateurs » inspirés de phénomènes naturels tels que la théorie de l'évolution (algorithmes évolutionnaires), la métallurgie (recuit-simulé) ou encore l'éthologie (PSO, algorithme des chauves-souris). [5]

3 Les Métaheuristiques

Les métaheuristiques (MH) sont des algorithmes d'optimisation stochastiques et itératifs capables de faire une exploration intelligente de l'espace des solutions en utilisant des approches statistiques ne nécessitant pas le calcul des dérivés de la fonction coût. [32]

Un des points intéressants des MH est qu'elles sont capables de s'adapter aux problèmes discrets et aussi aux problèmes continus. La caractéristique intelligente des métaheuristiques est représenté par leur capacité à diversifier et intensifier leur manière d'exploiter l'espace de solutions.

3.1 Caractéristiques

- Elles sont, au moins en partie, stochastiques : ce qui permet de faire face à l'explosion combinatoire des solutions possibles.
- Généralement d'origine discrète, elles ont l'avantage, décisif dans le cas continu d'être directes (ne nécessitent pas le calcul des dérivés.)
- Inspirées par des analogies, avec la physique (recuit simulé), la biologie (algorithmes évolutionnaires, recherche tabou) ou avec l'éthologie (colonie de fourmis, colonie d'abeilles.)
- Leur inconvénient majeur est la difficulté de réglage des paramètres. [33]

3.2 Classification

Il existe plusieurs façons de classer les métaheuristiques, l'une d'entre elle se base sur la stratégie d'évolution des solutions.

- *Métaheuristiques à solution unique* : aussi appelées méthodes de trajectoire ; elles manipulent une seule solution initiale et la font évoluer à travers les itérations, créant ainsi une trajectoire vers la meilleure solution. Des exemples de ces métaheuristiques sont la méthode de descente, le recuit simulé [23] et la recherche par tabous [13].

- *Métaheuristique à population* : manipulent un groupe de solutions initiales et les font évoluer à travers les itérations. La taille de la population est fixée par l'utilisateur et ne subit pas de modifications. Il faut noter qu'à la fin, nous retrouvons une seule solution : la meilleure. Ces méthodes sont divisées en deux grandes familles :
 - Les algorithmes évolutionnaires (les algorithmes génétiques [14], les stratégies d'évolution, la programmation évolutive, et la programmation génétique).
 - Les algorithmes d'intelligence par essaim (l'optimisation par colonie d'abeilles, les systèmes immunitaires artificiels, l'optimisation par essais particuliers...)

4 Les métaheuristiques à solution unique

4.1 Recuit-simulé (Simulated Annealing)

Origine : Cet algorithme est inspiré d'un processus de métallurgie : Le recuit thermique, fondée sur les lois de thermodynamiques énoncées par Boltzmann. Le terme « recuit » correspond à un réchauffement suivi d'un refroidissement. Si l'on refroidit un métal fondu rapidement (technique de la trempe), il va se stabiliser dans un état métastable « désordonné » correspondant à un minimum local de l'énergie. Par contre, si on abaisse la température progressivement, le métal atteindra un état cristallin correspondant à au minimum global de l'énergie.

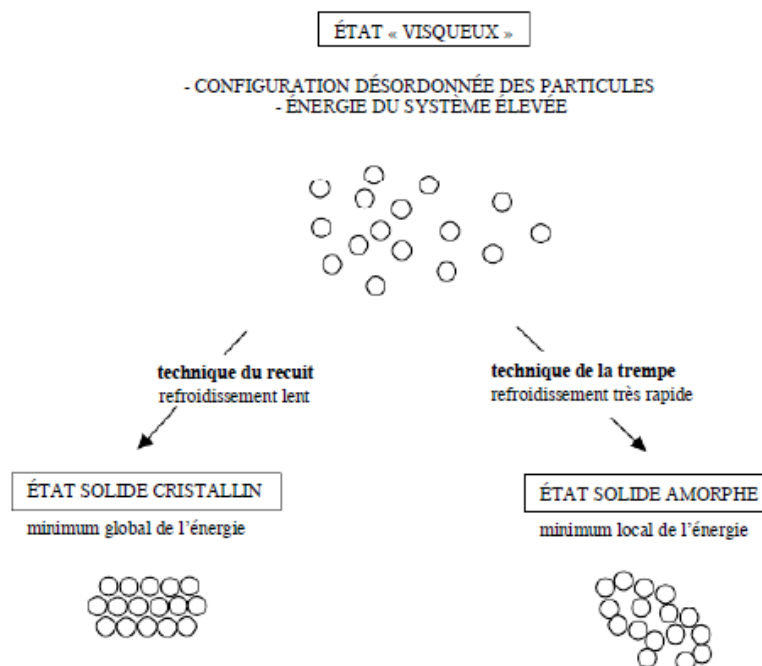


FIGURE 3.2 – illustration des techniques du recuit et de la trempe [33]

Analogie avec le système physique : Cette méthode est transposée en optimisation pour trouver les extrema d'une fonction : la fonction objectif, assimilée à l'énergie d'un matériau, est alors minimisée, moyennant l'introduction d'une température fictive, qui est contrôlée par une

fonction décroissante définissant un schéma de refroidissement.

Principe : Mit au point par Kirkpatrick et son équipe en 1983 [23] et indépendamment par Cerny en 1985 [9]. Elle consiste à introduire, un paramètre de contrôle, qui joue le rôle de la température. La “température” du système à optimiser doit avoir le même effet que celle du système physique : elle doit conditionner le nombre d’états accessibles et conduire vers l’état optimal si elle est abaissée de façon lente et bien contrôlée (technique du recuit) et vers un minimum local si elle est abaissée brutalement (technique de la trempe). [33]

La progression lente de la température est associée à de légères modifications locales dans les variables de la fonction objectif. En d’autres termes, il s’agit de commencer par d’importantes variations de la solution et diminuer leurs importances avec le temps afin de pouvoir se stabiliser dans un optimum global (ou s’en approchant).

Le recuit simulé est considéré comme une version améliorée de l’algorithme de Hill Climbing, car elle accepte certaines solutions « mauvaises » avec une certaine probabilité définie afin de s’échapper des optimums locaux. (Règles de Metropolis.) [33]

Paramètres :

- La température initiale T_0 .
- Le seuil de précision $\epsilon > 0$
- Le nombre maximum M d’états successifs qui n’arrivent pas à minimiser l’énergie au-delà du seuil de précision.
- Le nombre maximum K d’états successifs de température sans remplacer l’état minimal courant.
- Le critère d’arrêt du programme.

Etapas de l’algorithme : [4]

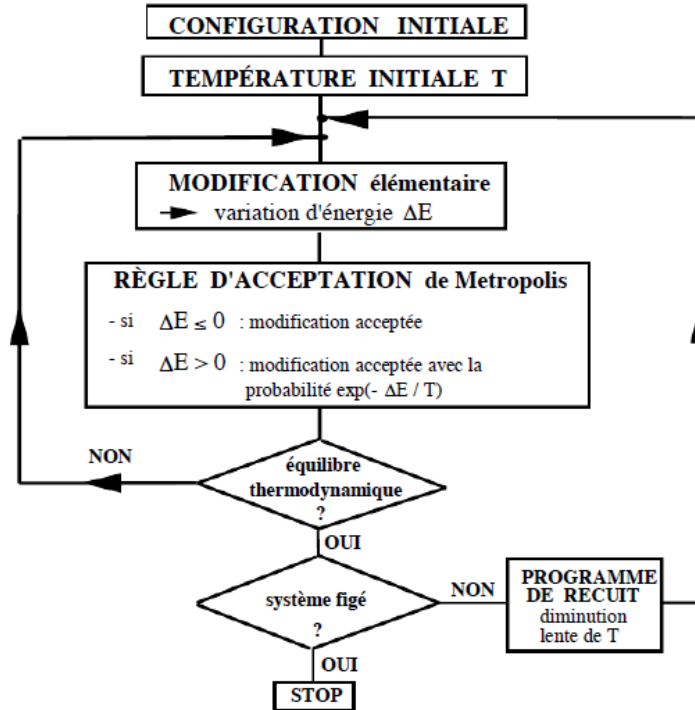


FIGURE 3.3 – Organigramme du Recuit-simulé [33]

Etape 1 : choisir, aléatoirement, une solution initiale x et évaluer sa fitness $f(x)$.

Etape 2 : choisir une « température » T initiale élevée.

Etape 3 : perturber cette solution pour obtenir une nouvelle solution $x' = x + \Delta x$.

Etape 4 : calculer $\Delta E = f(x') - f(x)$;

Etape 5 : accepter ou refuser la solution x' selon une règle d'acceptation. (règle de Metropolis)

Etape 6 : sauvegarder le meilleur point rencontré.

Etape 7 : si l'équilibre thermodynamique à T est atteint, alors abaisser légèrement la température sinon revenir à l'étape 3.

Etape 8 : si le système est figé ($T < T_{seuil}$) alors aller à l'étape 9 sinon aller à l'étape 3.

Etape 9 : arrêt du programme.

4.2 La recherche avec Tabous (Tabu search)

Origine : cette méthode s'inspire des mécanismes de la mémoire humaine.

Principe : La méthode de recherche avec tabous, ou simplement recherche tabou (TS :Tabu Search) a été formalisée par Fred Glover en 1986 [13]. Elle utilise explicitement l'historique de la recherche, à la fois pour échapper aux minima locaux et pour mettre en œuvre une stratégie d'exploration. De ce point de vue, la méthode taboue prend un chemin opposé à celui du recuit simulé, totalement dépourvu de mémoire, et donc incapable de tirer les leçons du passé. [5]

Types de listes taboues :

Mémoire à court terme : version la plus simple d'une liste taboue. C'est une liste d'une longueur l qui contient un certain nombre de mouvements interdits pour l'algorithme. Le principe est de vérifier toutes les solutions possibles dans le voisinage de la solution actuelle x . Ensuite, de prendre la meilleure parmi celles accessibles, et ce même si elle est de plus mauvaise qualité que x_i .

Une fois que la liste atteint sa limite, on commence à effacer les solutions les plus anciennes.

Mémoire à long terme : il est possible que la liste taboue ne soit pas suffisante pour éviter le phénomène de « cyclage », soit le fait de visiter cycliquement le même sous ensemble de solutions. [33] Parfois il est nécessaire de rallonger la liste afin de palier à ce problème. Il faut cependant garder à l'esprit qu'une liste trop longue peut empêcher de détecter l'optimum global.

Stratégies de convergence :

Diversification : afin d'éviter le problème de cyclage, autre que de rallonger la liste taboue, on peut aussi effectuer des sauts dans l'espace de solutions basé sur la mémoire existante. C'est-à-dire réessayer des solutions qui n'ont pas été utilisés depuis un moment. Une autre stratégie est d'accepter certaines solutions interdites en leur affectant une pénalité.

Intensification : le principe est d'exploiter plus intensivement les solutions intéressantes. Soit en rajoutant des contraintes, en revenant aux meilleures solutions et en explorant leurs voisinages de façon plus complète.

L'idée est de trouver un équilibre entre ces deux stratégies afin de converger de façon efficace et rapide vers l'optimum global. [33]

Paramètres :

- La longueur l de la liste taboue.
- Le types de mouvements possibles, selon les contraintes.
- Le nombre minimum de solutions pour un voisinage courant.
- Le seuil de précision $\epsilon > 0$.
- Le nombre maximum N d'itérations pour lesquelles la solution optimale reste inchangée.

Etapes de l'algorithme : [10]

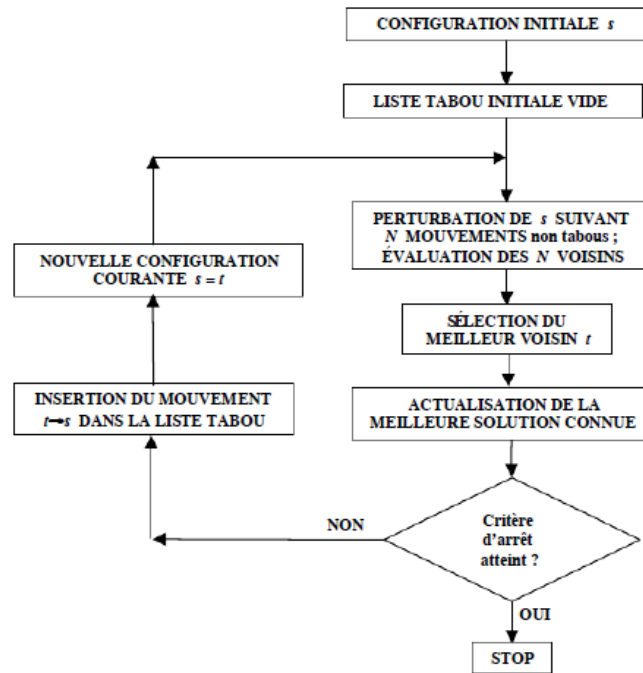


FIGURE 3.4 – Organigramme de la Liste taboue [33]

Etape 1 : choisir, aléatoirement, une solution initiale x du système à optimiser et évaluer la fitness $f(x)$.

Etape 2 : faire n fois :

- Perturber la solution x pour obtenir une nouvelle solution x' .
- Si le mouvement de x à $x' \in$ « Liste-tabou », alors effectuer une autre perturbation de x .
- Évaluer la valeur de f au point x' .
- Fin.

Etape 3 : Accepter le meilleur point x' parmi les n points voisins obtenus ; faire $x \leftarrow x'$.

Etape 4 : Mettre le mouvement inverse du mouvement effectué dans « Liste-tabou » (à la place du plus ancien élément de cette liste).

Etape 5 : Sauvegarder le meilleur point rencontré.

Etape 6 : Si pas d'amélioration depuis m itérations, alors aller à l'étape 7 ; sinon aller à l'étape 2.

Etape 7 : Arrêt du programme.

5 Les métaheuristiques à population

5.1 Les essais particuliers (PSO)

Origine et Inspiration : basée sur l'éthologie, cet algorithme reprend les principes d'auto-organisation qui permettent à un groupe d'organismes vivants (oiseaux, chauves-souris, fourmis) de communiquer et d'agir ensemble.

Inspiré du modèle simulant le déplacement des oiseaux par Craig Reynolds et proposé pour la première fois par James Kennedy et Russel Eberhart [22]. Cette métaheuristique fait partie des méthodes utilisant l'intelligence à essaim. Elle repose sur le principe de stigmergie, c'est-à-dire un groupe de particules qui ont à la fois une conscience propre et une conscience sociale ; capable de communiquer entre elles via une modification de l'environnement.

Ce phénomène est parfaitement illustré par la façon dont les fourmis opèrent. Certaines assument le rôle d'exploratrices à la recherche du chemin le plus court vers la nourriture et laissent derrière elles des phéromones détectables par le reste de la colonie afin d'indiquer la trajectoire. Les autres fourmis auront tendances à suivre le chemin contenant le plus haut taux de phéromones. [33], [5]

Principe : Selon Kennedy et Eberhart, la structure de cet algorithme est basée sur trois composants principaux. Les objets, les relations entre eux et des mécanismes de gestions.

1. *Les objets* : sont divisées en trois classes.
 - *Les exploratrices* : ce sont les particules qui vont parcourir l'espace de recherche, à chacune est attribuée une position, et une vitesse utilisée pour calculer les déplacements successifs.
 - *Les mémoriseurs* : ce sont des agents qui conservent les « bonnes » positions des exploratrices. Souvent il est question de la dernière position, mais les stratégies peuvent différer et il est intéressant de garder plusieurs points de sauvegarde afin d'éviter les minimums locaux.
 - *Les générateurs* : chargés de générer des nombres aléatoires selon une distribution uniforme afin de calculer les déplacements des exploratrices. Certaines versions de l'algorithme déterministes génèrent les nombres selon une liste prédéfinie, et de façon cyclique ; faisant passer l'algorithme de stochastique à déterministe [33]
2. *Les relations* : Dans la plupart des PSO, chaque particule exploratrice a son propre mémoriseur et se base sur les informations qu'il lui communique afin de se déplacer. Au lieu d'individualiser chaque particule, on peut considérer le corpus (exploratrice + mémoriseur) appelé « composé ».

La portée de la conscience sociale est représentée par l'échange d'informations entre les mémoriseurs. Si on considère seulement un voisinage d'une particule on parle de topologie locale. Dans le cas où toutes les particules peuvent communiquer sans frontières, on parle de topologie globale.

Il y'a évidemment plusieurs manières d'organiser la communication entre les essaims, la figure ci-dessous représente deux types de topologies possibles :

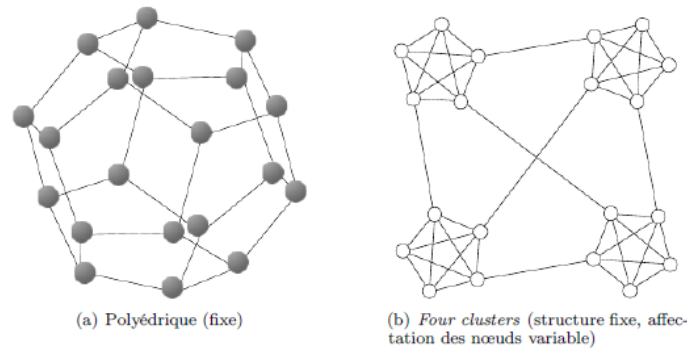


FIGURE 3.5 – exemples de deux topologies des relations entre composés [33]

3. *Les mécanismes* : Ce sont des stratégies de gestion, concernant l'initialisation des particules, les liens d'informations entre elles, les contraintes, les déplacements des particules et la gestion des paramètres.

Paramètres :

- Taille de l'essaim : N .
- Déplacement maximal en valeur absolue : V_{max} .
- Inertie : $0 < \omega < 1$.
- Coefficient de confiance cognitive $c_1 > 1$.
- Coefficient de confiance sociale : $c_2 = c_1$.

Etapes de l'algorithme [33] :

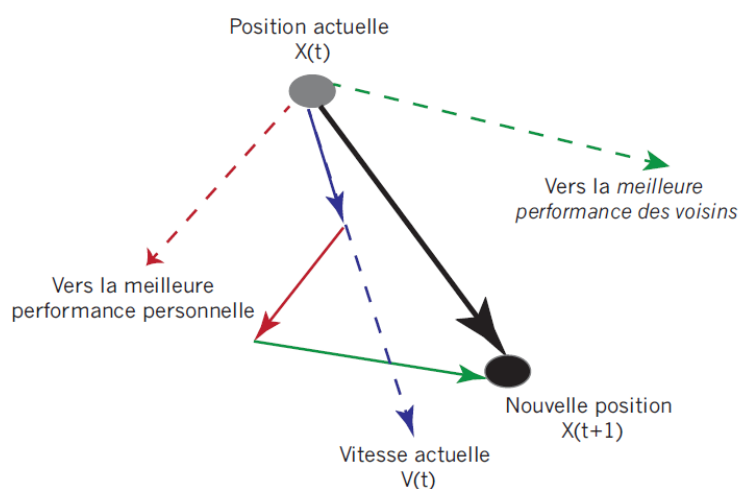


FIGURE 3.6 – Stratégie de déplacement d'une particule [5]

Etape 1 : générer un nombre N de particules ainsi que leurs vitesses de façon aléatoire.

Etape 2 : pour chaque particule i et dimension d :

- Calculer la fitness $f(x_{i,d})$
- Mettre à jour la meilleure position individuelle si nécessaire $p_{i,d}(t)$.
- Mettre à jour la meilleure position globale si nécessaire $g(t)$.

Etape 3 : Mettre à jour le paramètre d'inertie ω .

Etape 4 : pour chaque particule i et dimension d :

- Calculer la vitesse $v_{i,d}(t+1) = \omega * v_{i,d}(t) + c_1 * (p_{i,d}(t) - x_{i,d}(t)) + c_2 * (g(t) - x_{i,d}(t))$
- Calculer la position $x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$
- Mettre à jour la meilleure position globale si nécessaire.

Etape 5 : Si aucune condition d'arrêt n'est satisfaite, réitérer à partir de l'étape 2. Sinon passer à l'étape 6.

Etape 6 : Fin du programme. La solution est la meilleure position globale.
figure

5.2 Les Algorithmes Génétiques (AG)

Origine : proposé par J. Holland en 1972 puis popularisé par Goldberg en 1988 [14], les algorithmes génétiques découlent des algorithmes évolutionnaires. Leur fonctionnement est inspiré de la théorie de l'évolution de Charles Darwin, elles reprennent le principe de la sélection naturelle et du besoin d'adaptation pour la survie des espèces. [17]

Analogie avec la biologie : Les algorithmes génétiques reprennent plusieurs termes issus de la biologie mais adaptés à l'optimisation mathématique.

Un individu – ou solution – est représenté par un *chromosome*. Ce dernier est constitué de plusieurs *gènes* qui sont les valeurs de chaque solution. Chaque gène a une position précise dans le chromosome qui est appelée *locus*.

Au lieu d'itérations on parle de *générations*, et chaque génération de solution laisse place à une autre plus adaptée à son environnement. En effet, la théorie de Darwin stipule que seuls les individus les plus forts survivent et peuvent se reproduire, créant ainsi une progéniture encore plus robuste, ce qui revient à dire des solutions toujours plus proches de l'optimum.

La *force* d'un individu dans l'algorithme génétique est représentée par sa "fitness". Selon que l'on ait à faire à un problème de minimisation ou de maximisation, on définit ce que l'on entend par une meilleure fitness.

Les opérateurs de l'algorithme génétique

La fonction d'adaptation (fitness) : C'est la fonction qui permet d'évaluer les individus afin de leur attribuer un score selon qu'ils soient proches ou éloignés de la solution optimale. Elle est souvent assimilée à la fonction objectif.

Sélection : c'est un algorithme qui va sélectionner des individus - selon leur fitness - pour se reproduire et ainsi sauvegarder leur patrimoine génétique, c'est-à-dire des parents. Il existe plusieurs méthodes pour ce faire ; sélection par tournoi, sélection par probabilités et élitisme.

Croisement : Cet opérateur est inspiré du processus de reproduction biologique. Après que deux individus soient sélectionnés dans la population sélectionnée. Deux autres chromosomes seront créés à partir du mélange des gènes des parents. Le but du croisement est de diversifier la population, en produisant de nouvelles solutions à partir d'individus considérés forts.

Il existe plusieurs méthodes pour ce faire : croisement à un point, le croisement multi-points et le croisement arithmétique (ou barycentre).

Mutation : La mutation consiste à modifier aléatoirement un ou plusieurs gènes d'un chromosome selon une probabilité de mutation définie à l'avance et qui est souvent choisie très basse. En effet, tout comme dans la génétique, la mutation dans l'algorithme génétique est censée être un phénomène rare qui permet la création d'individus inédits. Il existe plusieurs méthodes : dans le cas de la représentation binaire : la méthode bit-flip, et pour la représentation réelle : la mutation uniforme et la mutation non-uniforme.

La mutation permet d'atteindre la propriété d'ergodicité, elle garantit un balayage de tout l'espace recherche ; donc plus de chances de trouver l'optimum globale. [15]

Paramètres de l'algorithme génétique

Nombre d'individus : En règle générale, ce nombre dépend de la taille de l'espace de recherche. Plus l'espace est grand plus on a besoin d'une population fournie afin d'en faire le tour.

Nombre de générations : Ou nombre d'itérations ; plus on augmente ce nombre, plus la solution a de chances pour converger. Le temps d'exécution en dépend fortement, il faut donc préférer un nombre qui réalise une solution approchée dans un temps raisonnable.

Probabilité de croisement : Souvent fixé à 0.7, on considère que plus ce taux est élevé, plus les solutions sont diversifiées et la convergence de la solution est assurée.

Probabilité de mutation : On considère que plus ce taux est petit, mieux c'est. La mutation doit rester un phénomène rare.

Dans certains cas où l'espace de recherche est très grand, on peut avoir recours à une probabilité de mutation élevée afin de pouvoir prétendre à l'*ergodicité*.

Critère d'arrêt : Si on a déjà une idée de l'optimum, alors l'algorithme ne s'arrête que lorsque ses solutions s'en approchent (ou l'égalent). Sinon, on peut soit définir un temps d'exécution maximal ou un certain nombre de générations à atteindre.

Étapes de l'Algorithme

Les fondements théoriques des AG ont été détaillés dans l'œuvre de David Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning (1989)*.

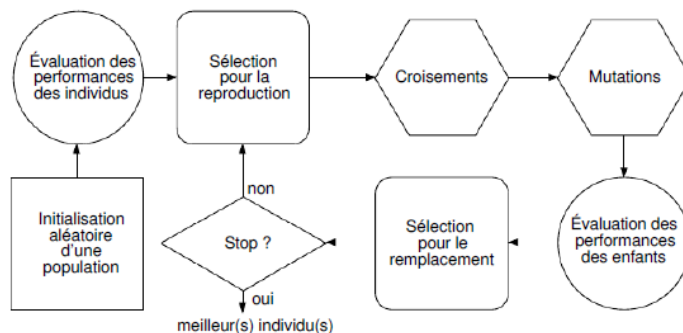


FIGURE 3.7 – Organigramme des Algorithmes Génétiques [33]

Type de codage : Avant de procéder aux étapes de l'AG, il faut tout d'abord choisir une représentation pour les chromosomes. Selon les problèmes, ça peut être une représentation binaire, réelle ou en base n .

La première étape est la génération de la population initiale, aussi appelée la genèse. Elle peut être faite de façon aléatoire, ou si l'on a une bonne connaissance de l'espace de recherche, on peut rajouter des contraintes pour mieux orienter la naissance d'individus proches de la solution. Une fois la génération créée, la fitness de chaque solution est évaluée.

C'est à partir de *la deuxième étape* que l'on commence l'itération. Des individus sont sélectionnés pour servir de parents à la prochaine génération, pour cela, il y'a plusieurs méthodes qui le plus souvent retiennent les meilleurs éléments.

Lors de *la troisième étape*, une population d'enfants est créée à partir de la recombinaison des gènes des parents suivant une probabilité de croisement ; sinon on reprend les parents comme individus dans la nouvelle génération.

La quatrième étape est celle de la mutation. Une mutation génétique est une modification rare de l'information génétique, et peut toucher une ou plusieurs parties du génome. Dans l'algorithme génétique, sa rareté est représentée par une probabilité de mutation assez basse – bien que cela dépende des applications -. Il y'a ici aussi plusieurs méthodes pour procéder à la modification d'un gène.

A partir de là, soit une condition d'arrêt est satisfaite et on récupère la meilleure solution de la dernière génération, ou alors on réitère depuis la deuxième étape. [6]

Avantages de l'algorithme génétique

- Peut s'adapter à différent problèmes.
- Trouve des solutions approchées en un temps raisonnable.

- Est moins susceptible de bloquer dans des optimums locaux.

Inconvénients de l’algorithme génétique

- Impossibilité de s’assurer que la solution trouvée est bien l’optimum (sauf si connaissance préalable de ladite solution).
- Le réglage des paramètres de l’algorithme afin d’avoir de bons résultats peut prendre un certain temps.

6 Conclusion

Les métaheuristiques sont nombreuses et chacune a son propre fonctionnement, et ce même si la stratégie générale reste la même, trouver un équilibre entre l’intensification et l’exploitation dans la recherche de l’optimum. Cette diversité offre une multitude de méthodes afin de s’adapter à tout autant de problèmes de détection de fautes qu’il est possible de rencontrer.

Les algorithmes génétiques que nous avons détaillé dans ce chapitre sont considérés comme l’une des métaheuristiques les plus connues et les plus performantes, et sont utilisés dans différents problèmes d’optimisation. Dans le prochain chapitre, nous utilisons les algorithmes génétiques comme métaheuristique pour la résolution d’un problème célèbre de diagnostic, à savoir : la détection des fuites dans les canalisations.

Chapitre 4

Application pour la détection de fuites

1 Introduction

Après avoir présenté plusieurs méthodes, à la fois de l'apprentissage et de l'optimisation, dans les chapitres précédents ; nous allons à présent voir les différentes façons dont elles peuvent être appliquées pour le même problème. Pour ce travail, nous nous sommes intéressés à la détection des fuites dans une canalisation d'eau.

C'est un problème qui ne perd pas en importance, malgré les diverses méthodes existantes et qui ne cessent d'accroître en nombre chaque année. Les pertes qui résultent de ces fuites peuvent devenir rapidement très conséquentes si on ne dispose pas d'un diagnostic rapide et précis, sans parler des risques de contaminations puisque le liquide n'est plus isolé.

Nous exposerons ensuite notre tentative pour résoudre ce problème, grâce à un algorithme génétique.

2 Exemples d'applications pour la détection des fuites

Allant sur plusieurs kilomètres, les canalisations sont sujettes à plusieurs types de perturbations – comme le trafic routier, des travaux dans les alentours ou encore les tremblements de terre dans les zones qui y sont sensibles – qui peuvent provoquer une mauvaise pression sur les matériaux, en addition à des défauts de fabrication, des faiblesses causées par la corrosion ou par le phénomène de coup de bélier. Ce sont des constructions qui sont destinées à fonctionner dans des conditions stressantes, ce qui peut provoquer bon nombres de défauts, dont les fuites.

Nous avons fait une sélection de méthodes d'apprentissages, d'optimisation et hybrides et leurs applications dans le domaine de détection de fuites, afin de montrer la diversification et la versatilité des techniques de ces techniques.

Auteurs	Méthodes	Type	Résumé
[Vitovsky et al, 2000][46]	Algorithme génétique	Optimisation	Cet article utilise l'algorithme génétique (AG) conjointement avec la méthode transitoire inverse pour détecter les fuites et les facteurs de friction dans les systèmes de distribution d'eau.
[Zhang et al, 2018][?]	PSO	Optimisation	Une méthode de détection et de localisation des fuites est proposée, sur la base d'analyses hydrauliques thermodynamiques inverses en régime transitoire et d'une optimisation par essais de particules (PSO) améliorée
[Huang et al, 2015][?]	Recuit simulé	Optimisation	Une approche numérique, appelée détection de fuite par recuit simulé (LDSA), est développée sur la base de la combinaison de simulations d'écoulement transitoire et de recuit simulé pour détecter les fuites dans un réseau de canalisations.
[Sornmuang et al, 2014][?]	RNA + liste taboue	Apprentissage et Optimisation	Cet article présente une nouvelle méthode hybride de recherche taboue adaptative et neuronale (NATS) pour la détection des fuites dans les pipelines. Les algorithmes coopératifs proposés sont formés d'un réseau neuronal artificiel (ANN) et d'une recherche taboue adaptative (ATS).

TABLE 4.1 – Exemple des applications de l'apprentissage et de l'optimisation au diagnostic

Auteurs	Méthodes	Type	Résumé
[Mashford et al, 2012][?]	SVM	Apprentissage et Optimisation	Cet article présente une méthode d'utilisation de l'analyse SVM pour interpréter les données obtenues par un ensemble de capteurs de pression ou de surveillant un réseau de canalisations afin d'obtenir des informations sur l'emplacement et la taille des fuites dans le réseau.
[Mandal et al, 2012][?]	Algorithme des abeilles et SVM	Optimisation et Apprentissage	Cet article utilise un classificateur SVM pour détecter de nouvelles classes. Pour son apprentissage, cet article utilise une technique d'intelligence en essaim : l'algorithme de la colonie d'abeilles artificielle (ABC), qui imite le comportement intelligent de recherche de nourriture des abeilles à miel.
[Wachla et al, 2015][?]	Reseaux de neurones flous	Apprentissage	Dans la méthode proposée, l'emplacement de la fuite est déterminé à l'aide du groupe de classificateurs neuro-flous. Le nombre de classificateurs correspond au nombre de zones dans lesquelles le réseau est divisé. La tâche de chaque classificateur est de changer l'état de sa sortie en cas de fuite dans le réseau associé à ce classificateur.

TABLE 4.2 – Exemple des applications de l'apprentissage et de l'optimisation au diagnostic (2)

3 Cas étudié

3.1 Introduction

Dans cette partie, nous présentons une amélioration d'une méthode déjà existante de la détection des fuites grâce à une métaheuristique : l'Algorithme Génétique.

Nous nous sommes basés sur les travaux de Wang et Ghidaoui [48] [47] où la détection des fuites a été effectuée grâce à une hybridation de méthodes hydrauliques et statistiques.

L'idée principale est de créer un modèle mathématique de la canalisation qui considère chaque point comme une fuite potentielle. Puis grâce à une technique connue, qui est l'étude de la réflexion d'une onde hydraulique transitoire dans la canalisation, des mesures de la pression manométrique (via hydrophones) sont prises puis comparées à ceux du modèle.

La fonction objectif est obtenue grâce à la méthode du matched field processing (MFP) dans un premier travail [48] puis grâce à l'estimateur de maximum de vraisemblance (EMV) dans un travail suivant [47], une notion introduite par les auteurs afin de prendre en compte le bruit et d'assurer de la robustesse à l'identification.

La différence entre les deux méthodes est la contrainte de la localisation des fuites proches. MFP permet de détecter deux fuites si la distance entre elles est supérieure à $0.5\lambda_{min}$, mais EMV – étant une méthode de super résolution - permet de détecter deux fuites si la distance entre elles est supérieure à $0.5\lambda_{min}$.

Finalement, la résolution du problème est effectuée grâce à la maximisation de la fonction objectif qui a pour entrées un couple de fuites potentielles. La localisation de l'optimum est réalisée grâce à un algorithme de recherche exhaustive (RE) qui a pour principe d'évaluer chaque cas possible.

Problématique : Le problème considéré est NP-difficile. La taille de l'espace des solutions est proportionnelle à la taille de la canalisation étudiée. Dans le cas de grandes distances, la recherche exhaustive – qui est une méthode exacte – devient sujette à l'explosion combinatoire et cela cause un temps de diagnostic non polynomial.

Objectif : comme vu dans le chapitre précédent, les métaheuristicques sont des méthodes de choix lorsqu'il s'agit de trouver des solutions approchées dans un temps raisonnable.

Dans ce travail, nous cherchons à remplacer la recherche exhaustive par un algorithme génétique afin d'avoir un diagnostic fiable et rapide.

4 Description du système

L'étude portera sur une canalisation horizontale connectée à deux réservoirs dont les coordonnées sont $x = x^U = 0$ et $x = x^D$ et leurs pressions manométriques respectives $H1 = 25m$ et $H2 = 20m$.

Le facteur de friction Darcy-Weisbach est $f = 0.02$, le débit en régime permanent est $Q_0^L = 0.0153m^3/s$, et la vitesse de l'onde est $a = 1200m/s$. H_0^L représente la pression manométrique en régime permanent. Il y'a trois capteurs (des hydrophones) situés stratégiquement au début et à la fin de la canalisation. Leurs coordonnées sont représentées par x_M .

On considère x^{L_n} les positions réelles des fuites et x^{L_n} les positions potentielles (avec $n = 1, \dots, N$). La taille de la fuite est donnée par $s^L = C^d A^L$ avec C^d le coefficient de débit de la fuite et A^L la surface de l'orifice.

5 Description de l'Algorithme Génétique

5.1 Codage des solutions

Parmi les trois types de codage existants pour l'Algorithme Génétique, le codage réel s'est imposé puisque nous avons seulement deux paramètres exprimés en mètre et représentés par des réels. Le chromosome est composé de deux gènes : la position de la première fuite (x1) et celle de la deuxième (x2).

x1	x2
----	----

La valeur du premier gène est choisie suivant un vecteur de valeurs équidistantes distribuées sur le long de l'intervalle imposé par la longueur de la canalisation.

Celle du deuxième gène est choisie aléatoirement entre deux valeurs : cette position doit être supérieure à la première avec une distance suivant la limite de l'estimateur par maximum de vraisemblance ($0.15\lambda_{min}$), et inférieure à la position du premier hydrophone en aval de la canalisation.

Remarque : La valeur de λ_{min} est proportionnelle à la longueur de la canalisation.

5.2 Fonction Objectif

Comme ce fut expliqué dans le chapitre précédent, nous avons pris comme fonction coût l'estimation d'un couple de fuites par maximum de vraisemblance comme détaillé dans l'article de Wang et Ghidaoui. [47]

Elle prend comme entrées deux fuites potentielles et retourne un coût. Notre objectif est trouver le couple de positions pour lequel la fonction est maximale.

$$\begin{aligned} \hat{x}^L &= \underset{x^L}{\operatorname{argmin}} \|\Delta h - G(x^L)(G^H(x^L)G(x^L))^{-1}G^H(x^L)\Delta h\|^2 \\ &= \underset{x^L}{\operatorname{argmax}} (\Delta h^H G(x^L)(G^H(x^L)G(x^L))^{-1}G^H(x^L)\Delta h) \end{aligned} \quad (4.1)$$

avec

$$x^L = (x_1, x_2)$$

5.3 Opérateurs génétiques adoptés

Comme expliqué dans le chapitre précédent, l'AG comporte trois types d'opérateurs qui permettent aux générations d'évoluer et de se diversifier à travers les itérations.

L'opérateur de sélection : permet de sélectionner les individus qui vont se faire l'objet de l'opération de croisement pour donner naissance à la prochaine génération. Dans notre AG, nous avons utilisé la sélection binaire par tournoi.

la sélection binaire par tournoi : Soit N le nombre d'individus de la génération, il faut en sélectionner $\frac{N}{2}$ puisque chaque paire de parents va engendrer deux enfants.

Dans le cas de la sélection binaire, deux individus sont choisis au hasard et seul celui qui présente la meilleure fitness peut accéder à la génération intermédiaire.

Cette étape est répétée jusqu'à ce que la génération intermédiaire soit remplie ($\frac{N}{2}$ composants). Il est tout à fait possible que les mêmes individus participent à plusieurs tournois : s'ils gagnent plusieurs fois, ils auront donc le droit d'être autant de fois dans la génération intermédiaire, ce qui favorisera la pérennité de leurs gènes.

L'opérateur de croisement : permet de créer de nouveaux individus à partir de la combinaison des gènes des parents sélectionnés en amont (population intermédiaire). Nous avons utilisé le croisement arithmétique (barycentre).

Croisement arithmétique : Si les croisements en un point/multi-points peuvent être implémentés pour n'importe quel choix de codage. Il est plus intéressant de procéder au croisement arithmétique lorsqu'il s'agit d'une représentation réelle.[31]

On définit un paramètre de pondération aléatoire alpha, souvent pris entre $[0 \ 1]$. Puis l'on sélectionne des gènes $p1(i)$ et $p2(i)$ dans chaque parent au même locus i et on crée deux nouveaux gènes par combinaison linéaire.

L'opérateur de mutation : permet de modifier les valeurs d'un ou de plusieurs gènes, juste suffisamment pour avoir une nouvelle solution. Nous avons utilisé une mutation non uniforme, avec une fonction définissant l'écart de mutation présentée dans un article de Cordon et Herrera en 2001. [36].

La mutation non uniforme : Après avoir tiré au sort quels gène(s) vont être modifiés, un taux de mutation est calculé ce taux sera à son maximum durant les premières itérations et selon le moment où l'on veut stopper la mutation, il décroîtra jusqu'à devenir nul. [8]

La fonction définissant la mutation non uniforme est présentée comme suit :

$$\Delta(t, y) = y * (1 - r^{(1-t/T)^\beta}) \quad (4.2)$$

$$x'_k = \begin{cases} x_k + \Delta(t, x_k^{max} - x_k) & si \\ rand = 0 \\ x_k - \Delta(t, x_k - x_k^{min}) & si \\ rand = 1 \end{cases} \quad (4.3)$$

T : l'indice de la génération pour laquelle on cesse la mutation.

t : l'indice de la génération en cours de traitement.
 r : un nombre aléatoire dans l'intervalle $[0, 1]$.
 β : paramètre de l'opérateur de mutation.
 y : l'écart entre la position à muter et la limite inférieure/supérieure de l'intervalle de recherche selon la décision binaire prise. $rand$: un nombre aléatoire dans l'intervalle $[0, 1]$.

5.4 Procédure de vérification de contraintes et de réparation des solutions infaisables

Les contraintes qui sont imposées par le système et par la méthode de EMV sont :

- 1) La distance entre deux fuites doit être au minimum $0.15\lambda_{min}$ pour que l'estimateur donne un résultat fiable.
- 2) Les solutions proposées doivent rester dans l'intervalle imposé, ou bien si les positions se trouvent à la fin de la canalisation : par la position du premier hydrophone.

La première contrainte est prise en compte lors de la création de la population initiale. Avec une condition qui permet de fixer une distance minimum entre deux fuites.

Elle est également vérifiée après l'opération de mutation.

La deuxième contrainte est vérifiée après la mutation de la population enfant, et donc juste avant son évaluation. La stratégie de réparation consiste à désigner le minimum entre la valeur proposée et la limite inférieure de l'intervalle, puis le maximum entre ladite valeur et la limite supérieure.

6 Expériences numériques et interprétations

Nous présentons dans cette section les résultats numériques obtenus par notre algorithme génétique.

Le code représentant le modèle de la canalisation et la fonction objectif appartient à *X. Wang et M. Ghidaoui*.

Le code original appartient à *Mostapha Kalami Heris* [20], nous l'avons modifié afin de l'adapter au problème étudié.

6.1 Générations d'instances

La génération d'instances est liée à l'adaptation des paramètres de l'algorithme génétique au problème décrit dans (Wang et Ghidaoui, 2018). Il s'agit de déterminer le nombre d'individus, le nombre d'itération et les probabilités de mutation et de croisement afin d'avoir un algorithme fonctionnel et des résultats satisfaisants.

Nous avons choisi une canalisation de 1 Km pour nous servir de référence, avec une limite imposée par EMV de 30 m de distance entre deux fuites. L'instance de 1,5 Km nous a servi de comparaison avec les cas présentés dans l'article de Wang et Ghidaoui.

Nous avons ensuite testé l'AG sur de grandes distances – à savoir 4Km, 8Km et 12Km - afin de vérifier son aptitude à fonctionner dans un grand espace de recherche.

— Les fuites ont été choisies avec une distance de $0.5\lambda_{min}$ entre elles.

- Les paramètres de l’algorithme ont dû être réadaptés au fur et à mesure de l’augmentation de l’espace de recherche.
- Le temps d’exécution de la recherche exhaustive a été limité à 100 min.

L’inconvénient majeur de cette partie de la simulation est que l’on peut difficilement vérifier les erreurs commises par l’AG sans les résultats de la recherche exhaustive.

6.2 Résultats obtenus

Paramètres de l’algorithme génétique adopté : Nous avons testé les différents paramètres afin d’optimiser la recherche pour une canalisation d’une longueur de 1 km. Nous avons procédé par tâtonnement, en commençant par des valeurs faibles et en augmentant selon les résultats. Nous sommes parvenus à cette structure :

Codage	Réel
Nombre de gènes	2
Croisement	Arithmétique avec $Pc = 0.6$
Mutation	Mutation non uniforme avec $Pm = 0.4$
Taille de la population	150
Nombre d’itérations	350

Le tableau suivant présente les résultats pour l’implémentation de l’AG sur toute la longueur des canalisations (respectivement 1 km et 1,5 km), avec comme instance des fuites proches et éloignés pour chaque cas.

Les fuites prises pour 1.5 km sont des exemples pris de l’article de Wang et Ghidaoui (2018) et qui permettent de confirmer l’efficacité de l’algorithme génétique.

N	L (Km)	Intervalle (m)	Fuites réelles (m)	Résultats (m)		Temps d’exécution (min)	
				RE	AG	RE	AG
1	1 Km	1000 m	(45 80)	(45 80)	(40 80)	25min	6min
			(180 650)	(180 650)	(180 650)	25min	6min
2	1,5 Km	1500 m	(260 350)	(260 350)	(258 351)	61min	6min
			(140 470)	(140 470)	(141 368)	61min	6min

TABLE 4.3 – Résultats obtenus par l’AG pour les petites distances

Résultats pour de grandes distances :

N	L (Km)	Intervalle (m)	Fuites réelles (m)	Résultats (m)		Temps d'exécution (min)	
				RE	AG	RE	AG
3	4 Km	4000 m	(820 1000)	(- -)	(821 1000)	>100min	15min
4	8 Km	8000 m	(4500 4820)	(- -)	(4497 4827)	>100min	17min
5	12 Km	12000 m	(6125 6550)	(- -)	(6129 6557)	>100min	20min

TABLE 4.4 – Résultats obtenus par l'AG pour les grandes distances

6.3 Interprétation des résultats :

Avant de commencer à analyser la performance de l'algorithme génétique, il faudrait éclaircir les erreurs commises par l'algorithme original (EMV + recherche exhaustive (RE)) et qui ont été reproduites par l'AG.

EMV est une méthode dont l'efficacité dépend du taux d'information qui lui est transmis par l'onde acoustique se propageant dans la canalisation, à travers les mesures prises par les hydrophones. Les auteurs ont précisé dans leurs articles que plus l'espace de recherche augmenterait, plus il serait judicieux d'augmenter le nombre de fréquences exploitables.

Puisque la distance minimale entre deux fuites est définie par λ_{\min} et que $a = \frac{\lambda_{\min}}{f}$. Augmenter les fréquences permettrait d'atténuer la croissance de λ_{\min} (avec a la vitesse d'onde et f les fréquences exploitées).

Nous avons préféré garder le nombre de fréquences utilisé par défaut dans les articles parce que ça n'influence pas la performance de l'algorithme génétique. Le résultat est une perte de précision du côté de EMV et qui est retrouvée par la recherche exhaustive et donc par l'AG.

L'AG a donné des résultats équivalents à ceux de la recherche exhaustive tant que l'intervalle sur lequel il était exécuté restait inférieur à 1500 m. A partir de là, des erreurs commencent à apparaître et bien qu'elles soient négligeables c'est le signe qu'il faut ajuster à nouveau les paramètres de l'algorithme génétique afin qu'il puisse s'adapter à de plus grands espaces de recherche.

A partir de 4 Km, le temps d'exécution de la recherche exhaustive ne nous permet plus d'avoir des résultats en moins de 100 min, il devient donc difficile d'évaluer avec précision la performance de l'AG.

Nous déduisons de l'écart dans l'estimation que la précision est inversement proportionnelle à l'augmentation de la taille de l'espace de recherche. Malgré la réadaptation des paramètres, il est difficile d'obtenir un résultat exact.

D'un autre côté, nous avons réussi à avoir une estimation suffisamment proche des fuites réelles en un temps raisonnable en comparaison avec la méthode de recherche exhaustive.

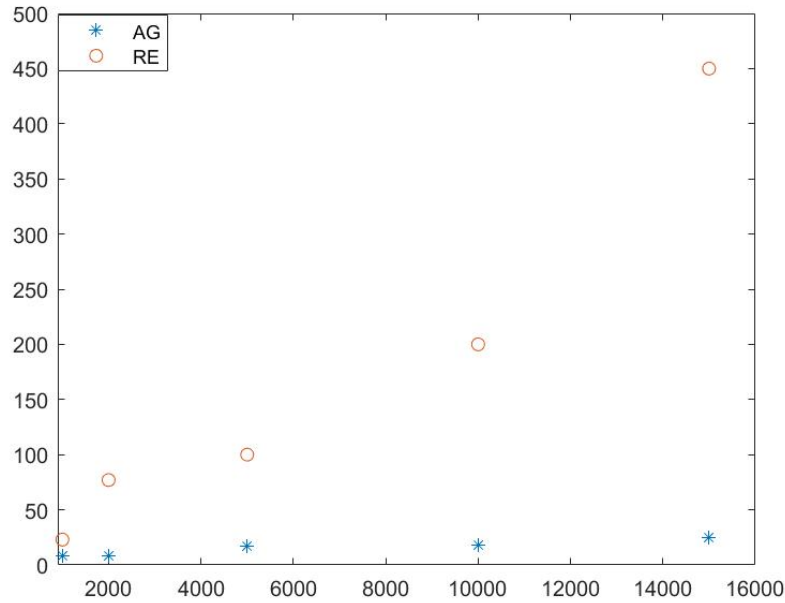


FIGURE 4.1 – Évolution du temps d'exécution par rapport à la longueur des canalisations

Puisque nous sommes en train de traiter un problème NP-difficile, nous avons limité le temps d'exécution de la recherche exhaustive à 100 min et n'avons donc pas conduit les exécutions jusqu'au bout. En comparaison, l'augmentation du temps de convergence de l'AG est plus raisonnable et nous permet de conclure que l'utilisation d'une métaheuristique est une nécessité pour les grandes instances de ce problème (grandes distances pour les canalisations).

6.4 Conclusion de l'analyse

Les métaheuristiques ont pour but de résoudre les problèmes NP-difficiles en un temps raisonnable, même si il n'y a aucune garantie sur la qualité d'optimalité de la solution trouvée.

C'est ce que nous avons observé dans cette étude. Malgré l'explosion combinatoire qui rend le temps d'exécution de la recherche exhaustive très excessif, l'AG a su exploiter l'espace de recherche de façon adéquate jusqu'à un certain point. La réadaptation de ses paramètres devient plus fréquente à chaque fois que l'on augmente la taille de la canalisation.

On considère en règle générale que le nombre de générations nécessaires pour atteindre la convergence augmente avec la taille de l'espace de recherche; mais puisque l'AG est un algorithme stochastique, son temps de convergence change à chaque exécution.

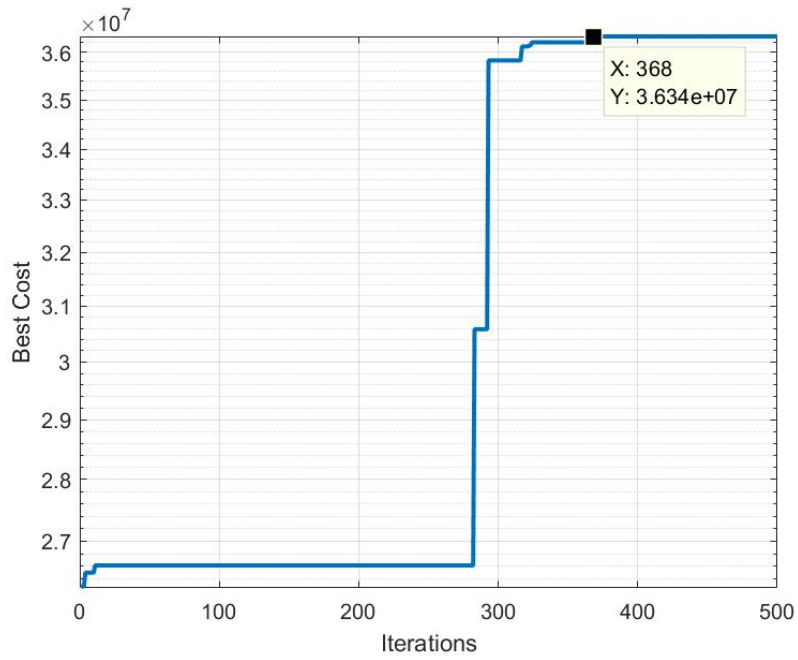


FIGURE 4.2 – Pour une canalisation de longueur $L = 1km$ le temps de convergence est $t_{conv} = 6min$

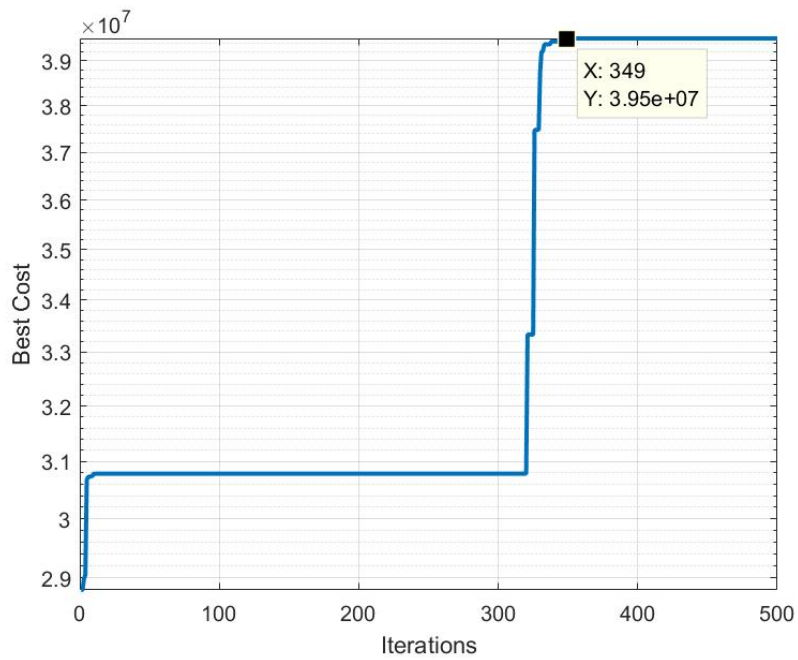


FIGURE 4.3 – Pour une canalisation de longueur $L = 1,5km$ le temps de convergence est $t_{conv} = 6min$

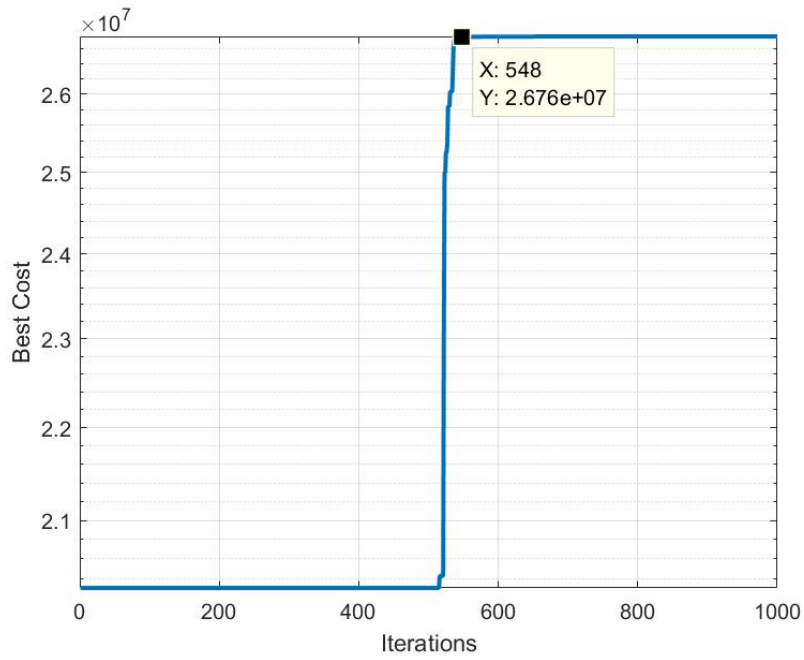


FIGURE 4.4 – Pour une canalisation de longueur $L = 4km$ le temps de convergence est $t_{conv} = 13min$

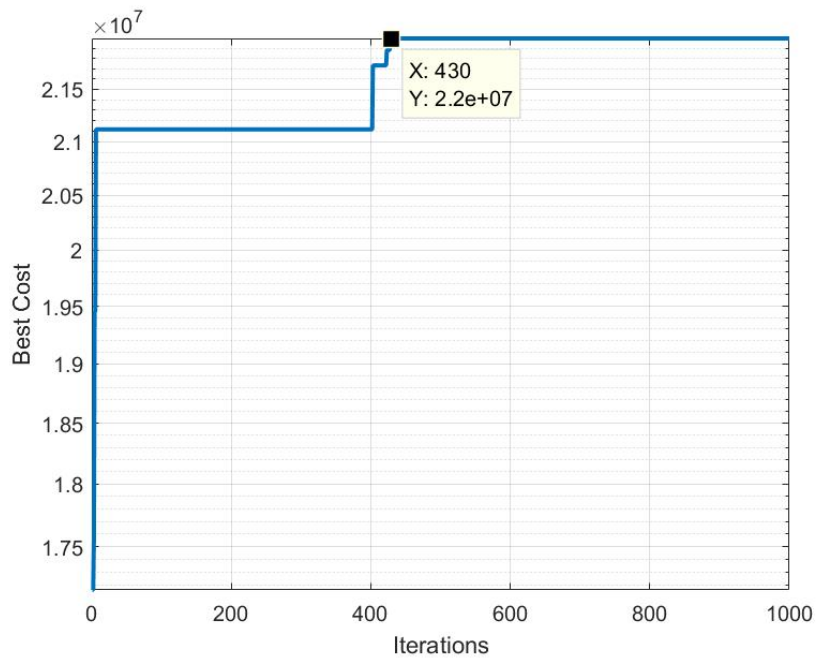


FIGURE 4.5 – Pour une canalisation de longueur $L = 8km$ le temps de convergence est $t_{conv} = 10min$

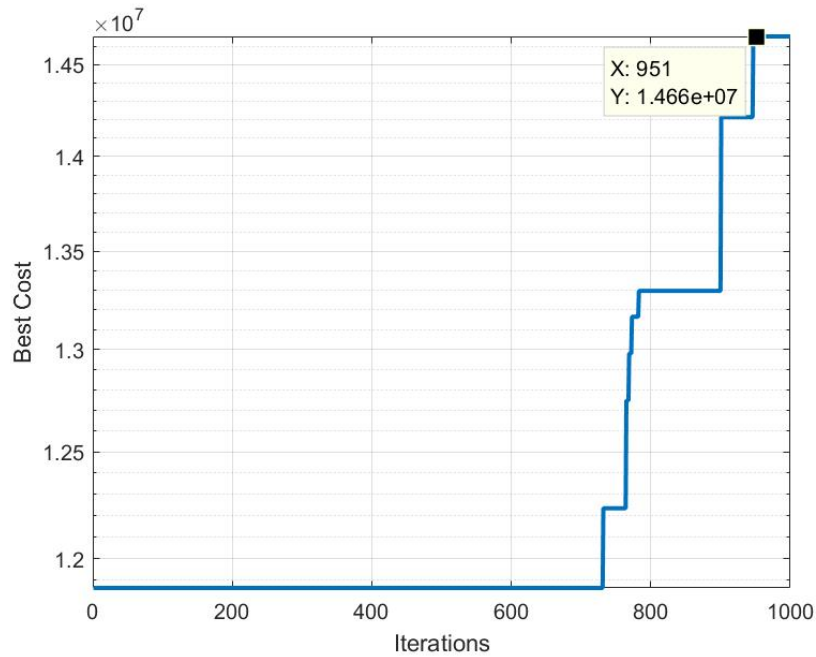


FIGURE 4.6 – Pour une canalisation de longueur $L = 12km$ le temps de convergence est $t_{conv} = 23min$

A travers les figures (4.3 4.4 4.5 4.6 4.7), on peut clairement voir la nature stochastique de l’algorithme génétique, les temps de convergences n’évoluent pas forcément proportionnellement à la taille de l’espace de recherche. Cela dépend de la qualité des solutions initialisées et de sa rapidité à se rapprocher de la solution exacte.

7 Conclusion

Nous avons présenté dans ce chapitre une sélection de méthodes de classification et d’optimisation appliquées au problème de détection de fuites.

Puis nous avons exposé notre travail, une optimisation d’un diagnostic déjà existant grâce à un algorithme génétique. L’AG que nous avons adaptée a présenté de bons résultats jusqu’à une certaine longueur de canalisation.

L’inconvénient majeur de cette méthode étant que la précision a tendance à diminuer avec l’augmentation de l’espace de recherche – la longueur de la canalisation étudiée - ; et c’est pour cette raison que nous avons été obligé de réadapter les paramètres de l’AG. Néanmoins, cela nous a permis d’avoir des résultats satisfaisants sur de très grandes distances moyennant un temps d’exécution raisonnable.

Conclusion générale

Nous avons présenté dans ce travail des méthodes de l'apprentissage et de l'optimisation fréquemment utilisées dans le diagnostic dans le but d'étudier les stratégies utilisées par ces techniques qui les rendent plus performantes que d'autres plus conventionnelles.

Les méthodes se basant sur l'apprentissage telles que les réseaux de neurones artificiels demandent beaucoup d'efforts et de données afin d'être mises en place mais le résultat est un classificateur autonome avec une capacité d'adaptation aux nouvelles situations. L'apprentissage non supervisé, bien que compliqué à atteindre, est l'une des méthodes les plus prometteuses du diagnostic automatique.

Les méthodes se basant sur l'optimisation, et principalement les métaheuristiques, lorsqu'elles sont utilisées à des fins de diagnostic de défauts, dans la grande majorité des cas, elles représentent une méthodologie de soutien pour d'autres techniques (traitement de signal, statistique), en particulier pour les tâches de réglage des paramètres. L'objectif récurrent lors de l'utilisation des métaheuristiques est d'obtenir un résultat approché en un temps raisonnable. Il existe également des approches qui les utilisent comme une technique autonome pour effectuer le diagnostic.

Nous avons illustré l'intérêt de ces méthodes pour le diagnostic avec un exemple concret qui est la détection des fuites, en utilisant un algorithme génétique pour optimiser une méthodologie déjà existante.[47],[48] Les résultats très prometteurs obtenus ont été présentés et interprétés.

Un prochain travail pourrait se pencher sur la mise au point d'une technique hybride mêlant apprentissage et optimisation.

Résumé

Ce travail présente une synthèse des méthodes d'apprentissage et d'optimisation les plus utilisées pour la détection et le diagnostic de défauts. Puis, une métaheuristique à population, plus précisément l'algorithme génétique est adapté dans le but d'optimiser une méthode déjà existante - basée sur la méthode de l'analyse inverse par onde transitoire - de détection de fuites dans une canalisation. L'objectif étant de remplacer une méthode exacte d'optimisation (coûteuse en temps et en mémoire) par une méthode approchée qui nous permettra de gagner en temps de diagnostic.

Mots-Clés : Intelligence Artificielle, Apprentissage, Optimisation, Métaheuristiques, Détection de fuites, Algorithme génétique.

Abstract

This work presents a synthesis of artificial intelligence's most used methods for the detection and diagnosis of faults. Then, a genetic algorithm is adapted in order to optimize an already existing method of detecting leaks in a pipeline - based on inverse transient theory -, the purpose being keeping the same detection precision but with a swifter diagnosis time.

Key words : Artificial Intelligence, Learning, Optimization, Metaheuristics, Leak detection, Genetic algorithm.

ملخص

يقدم هذا العمل توليفة من أساليب الذكاء الاصطناعي المستخدمة في الكشف عن الأعطال وتشخيصها. بعد ذلك، يتم تكييف الخوارزمية الجينية من أجل تحسين طريقة موجودة بالفعل لاكتشاف التسريبات في خط الأنابيب.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم، التحسين، الخصائص الوصفية، كشف التسرب، الخوارزمية الجينية.

Bibliographie

- [1] A.B. Baggeroer, W.A. Kuperman, and P.N. Mikhalevsky. An overview of matched field methods in ocean acoustics. *IEEE Journal of Oceanic Engineering*, 18(4) :401–424, 1993.
- [2] Michel Batteux. *Diagnosticabilité et diagnostic de systèmes technologiques pilotés*. Theses, Université Paris Sud - Paris XI, December 2011.
- [3] Nassim Belkharichi. *Contribution à l'étude du diagnostic et de la commande tolérante aux fautes par l'approche structurelle*. PhD thesis, Université de Toulouse, 2011.
- [4] R. Ben-Mansour, M.A. Habib, A. Khalifa, K. Youcef-Toumi, and D. Chatzigeorgiou. Computational fluid dynamic simulation of small leaks in water pipelines for direct leak pressure transduction. *Computers & Fluids*, 57 :110–123, mar 2012.
- [5] Pascal BOULANGER. *Nouvelles méthode de détection et de localisation de fuites par émission acoustique*. PhD thesis, UNIVERSITÉ DE PARIS-SUD, CENTRE D'ORSAY, 1993.
- [6] Ilhem Boussaid. *Perfectionnement de métaheuristiques pour l'optimisation continue*. PhD thesis, Université Paris-Est, 2013.
- [7] M.A Brahami. *Contributions à la conception des réseaux logistiques : Résolution des problèmes intégrés de localisation et de conception des réseaux de transport sous contraintes environnementales et de capacités*. PhD thesis, Université de Tlemcen, 2020.
- [8] Mustapha Anwar Brahami, Mohammed Dahane, Mehdi Souier, and M'hammed Sahnoun. Sustainable capacitated facility location/network design problem : a non-dominated sorting genetic algorithm based multiobjective approach. *Annals of Operations Research*, may 2020.
- [9] T. K. Chan, Cheng Siong Chin, and Xionghu Zhong. Review of current technologies and proposed intelligent methodologies for water distributed network leakage detection. *IEEE Access*, 6 :78846–78867, 2018.
- [10] M. Hanif Chaudhry. *Applied Hydraulic Transients*. Springer New York, 2014.
- [11] Andrew F. Colombo, Pedro Lee, and Bryan W. Karney. A selective literature review of transient-based leak detection methods. *Journal of Hydro-environment Research*, 2(4) :212–227, apr 2009.
- [12] D. Covas, H. Ramos, N. Graham, and C. Maksimovic. Application of hydraulic transients for leak detection in water supply systems. *Water Supply*, 4(5-6) :365–374, dec 2004.
- [13] Henrique V. da Silva, Celso K. Morooka, Ivan R. Guilherme, Tiago C. da Fonseca, and José R.P. Mendes. Leak detection in petroleum pipelines using a fuzzy system. *Journal of Petroleum Science and Engineering*, 49(3-4) :223–238, dec 2005.
- [14] B. Farley, S. R. Mounce, and J. B. Boxall. Field testing of an optimal sensor placement methodology for event detection in an urban water distribution network. *Urban Water Journal*, 7(6) :345–356, dec 2010.
- [15] Erik Frisk. Residual generation for fault diagnosis. *Linköping University*, 2001.

- [16] M. Gal, A. M. Reading, N. Rawlinson, and V. Schulte-Pelkum. Matched field processing of three-component seismic array data applied to rayleigh and love microseisms. *Journal of Geophysical Research : Solid Earth*, 123(8) :6871–6889, aug 2018.
- [17] Fred Glover. Tabu search : A tutorial. *Interfaces*, 20(4) :74–94, aug 1990.
- [18] David E. Goldberg and John H. Holland. *Machine Learning*, 3(2/3) :95–99, 1988.
- [19] Ouahib GUENOUNOU. *Méthodologie de conception de contrôleurs intelligents par l'approche génétique. Application à un bioprocédé*. PhD thesis, Université de Toulouse, 2009.
- [20] Mostapha Kalami Heris. Mostapha kalami heris, practical genetic algorithms in python and matlab – video tutoria. <https://yarpiz.com/632/ypga191215-practical-genetic-algorithms-in-python-and-matlab>), 2020.
- [21] F. Herrera, M. Lozano, and J.L. Verdegay. *Artificial Intelligence Review*, 12(4) :265–319, 1998.
- [22] Jinqiu Hu, Laibin Zhang, and Wei Liang. Detection of small leakage from long transportation pipeline with complex noise. *Journal of Loss Prevention in the Process Industries*, 24(4) :449–457, jul 2011.
- [23] Shih-Chu Huang, Wu-Wen Lin, Meng-Tsan Tsai, and Mao-Hsiung Chen. Fiber optic in-line distributed sensor for detection and localization of the pipeline leaks. *Sensors and Actuators A : Physical*, 135(2) :570–579, apr 2007.
- [24] Rolf Isermann. *Fault-Diagnosis Systems*. Springer Berlin Heidelberg, 2006.
- [25] Éli D Taillard Patrick Siarry J Dréo, A Petrowski. *Métaheuristiques pour l'optimisation difficile*. 2013.
- [26] Steven M. Kay. *Fundamentals of Statistical Signal Processing : Estimation Theory*. PRENTICE HALL SIGNAL PROCESSING SERIES, 1993.
- [27] Y. A. Khulief, A. Khalifa, R. Ben Mansour, and M. A. Habib. Acoustic detection of leaks in water pipelines using measurements inside pipe. *Journal of Pipeline Systems Engineering and Practice*, 3(2) :47–54, may 2012.
- [28] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, may 1983.
- [29] H. Krim and M. Viberg. Two decades of array signal processing research : the parametric approach. *IEEE Signal Process. Mag.*, 13 :67–94, 1996.
- [30] James A. Liggett and Li-Chung Chen. Inverse transient analysis in pipe networks. *Journal of Hydraulic Engineering*, 120(8) :934–955, aug 1994.
- [31] G. Y. Lin and C. J. Wu. Design of fuzzy logic controllers using genetic algorithms. *IEEE Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 6 :104–109, 1999.
- [32] Salowa METHNANI. *Diagnostic, reconstruction et identification des défauts capteurs et actionneurs : application aux stations d'épurations des eaux usées*. PhD thesis, École Doctorale Mathématiques et Informatique [Équipe Estimation-Commande-Diagnostic (ES-CODI)], 2012.
- [33] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1992.
- [34] Mario Alberto Coutino Minguez. Identification of room boundaries for sound field estimation. Master's thesis, TU Delft, The Netherlands, 2016.
- [35] Abdouramane Moussa Ali, Cédric Join, and Frédéric Hamelin. Diagnostic des systèmes dynamiques linéaires sans modèle explicite. *Journal Européen des Systèmes Automatisés (JESA)*, 45(Spécial : Nouvelles tendances en automatique) :339–362, January 2011.

- [36] F. Herrera O. Cordon and P. Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on Fuzzy Systems*, pages 667—674, 2001.
- [37] Didem Ozevin and James Harding. Novel leak localization in pressurized pipeline networks using acoustic emission and geometric connectivity. *International Journal of Pressure Vessels and Piping*, 92 :63–69, apr 2012.
- [38] Ranko S. Pudar and James A. Liggett. Leaks in pipe networks. *Journal of Hydraulic Engineering*, 118(7) :1031–1046, jul 1992.
- [39] Philippe Rigollet. Lecture 10 : Maximum likelihood estimation, 2016, Massachusetts Institute of Technology (MIT).
- [40] Slami Saadi. *Introduction à l'optimisation métaheuristiques*.
- [41] Patrick Siarry. *Métaheuristiques*.
- [42] Ehsan Sobhani-Tehrani and Khashayar Khorasani. *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach*. Springer US, 2009.
- [43] Christine Solnon. *Optimisation par colonies de fourmis*. 2021.
- [44] Liang Sun and Ning Chang. Integrated-signal-based leak location method for liquid pipelines. *Journal of Loss Prevention in the Process Industries*, 32 :311–318, nov 2014.
- [45] A. Tolstoy, K.V. Horoshenkov, and M.T. Bin Ali. Detecting pipe changes via acoustic matched field processing. *Applied Acoustics*, 70(5) :695–702, may 2009.
- [46] John P. Vítkovský, Angus R. Simpson, and Martin F. Lambert. Leak detection and calibration using transients and genetic algorithms. *Journal of Water Resources Planning and Management*, 126(4) :262–265, jul 2000.
- [47] Xun Wang and Mohamed S. Ghidaoui. Identification of multiple leaks in pipeline : Linearized model, maximum likelihood, and super-resolution localization. *Mechanical Systems and Signal Processing*, 107 :529–548, jul 2018.
- [48] Xun Wang and Mohamed S. Ghidaoui. Pipeline leak detection using the matched-field processing method. *Journal of Hydraulic Engineering*, 144(6) :04018030, jun 2018.
- [49] Xun Wang and Mohamed S. Ghidaoui. Identification of multiple leaks in pipeline II : Iterative beamforming and leak number estimation. *Mechanical Systems and Signal Processing*, 119 :346–362, mar 2019.
- [50] Yipeng Wu and Shuming Liu. A review of data-driven approaches for burst detection in water distribution systems. *Urban Water Journal*, 14(9) :972–983, feb 2017.