



UNIVERSITE MOHAMED BOUDIAF - M'SILA

**FACULTE DES MATHEMATIQUES ET
DE L'INFORMATIQUE**



DEPARTEMENT D'INFORMATIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique Décisionnelle et Optimisation

Par : HASSANI HADDA et ALLAM HIBA

SUJET

**L'application des algorithmes de
colonies de fourmis pour le diagnostic
des systèmes dynamiques et complexes**

Soutenu publiquement le : 09 / 06 /2026 devant le jury composé de :

Dr. Loucif Hamza

Université de M'sila Président

Dr. Mouhoub Nasser Eddine

Université de M'sila Rapporteur

Dr. Chikouche Soumia

Université de M'sila Examineur

Promotion : 2025 /2026

الاهداء

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
(وَقُلْ اَعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ)

إلى أمي يامنة...

إلى من كانت السر الحقيقي وراء كل خطوة وصلت إليها، إلى من تعبت لأجل أن أرتاح، وسهرت لأجل أن أنام، وضحت بصمت دون أن تنتظر مقابلاً.

إلى من صنعت مني إنسانة أقوى ورفعتني بدعواتها وحنانها وصبرها، فكان وصولي هذا بفضل الله ثم بفضلها، أنت الحلم الذي أعيش لأجله، والنجاح الذي لا يكتمل إلا برضاك.

إلى أبي محمد...

إلى من علمني معنى المسؤولية قبل أن أتعلم معنى الطريق، إلى من زرع في الصبر والاعتماد على النفس، بفضلك تعلمت كيف أقف وحدي، وكيف أواصل رغم الصعاب دون أن أنهار.

إلى أخوالي العزيزين اللذين رحلا عن هذه الدنيا...إلى من تركوا في القلب أثراً لا يمحي، وذكرى لا تغيب. رحمكم الله رحمة واسعة، وجعل الجنة داركم، فغيا بكم حاضر في دعائي دائماً، وفي كل نجاح أفتقد حضوركم وفرحتكم.

إلى إخوتي وأخواتي...

إلى من وقفوا معي في الأوقات الصعبة، وكانوا سنداً حقيقياً حين ضاقت بي الدنيا، فخففوا عني الحمل بكلماتهم أنتم الدفء الذي يخفف عني تعب الطريق، واليد التي تمتد حين أحتاج الدعم دون أن أطلب.

إلى عائلتي...

إلى كل من دعم بكلمة أو دعاء أو موقف فكان له أثر جميل في هذا الطريق.

إلى نفسي...

إلى تلك الروح التي لم تستسلم رغم التعب، والتي واصلت الطريق رغم الصعوبات، حتى وصلت إلى هذه اللحظة.

وأخيراً... هذا العمل لكل من آمن بي.

إلى صديقاتي رفيقات الدرب الدراسي، شكراً على الأيام الجميلة والذكريات التي ستبقى دائماً. أتمنى لكن مستقبلاً مليئاً بالنجاح والتألق.

حساني حدة

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
(وَأَنْ لَّيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى)

إلى والديّ...

إلى من كانا سبب نجاحي بعد الله، إلى أبي عبد المجيد فخري واعتزازي، الذي علّمني المسؤولية والقوة،
وكان سنداً لي في مختلف مراحل حياتي، وإلى أمي سامية التي حملت من اسمها نصيباً، فكانت سامية
الخلق، سامية المقام.

إلى إخوتي محمد وإبراهيم ويوسف...

إلى سندي الحقيقي، الذين وقفوا معي في الأوقات الصعبة وكانوا معي دائماً دون تردد.
إلى زوجة أخي...

من كانت بكلماتها الطيبة ودعمها الصادق مصدراً لراحة وتشجيع.

إلى بنات أخي...

إلى توأم قلبي ندى ونور، إلى القلوب الصغيرة المليئة بالبراءة ونور العائلة.

إلى صديقاتي...

إلى من شاركني الطريق وكنّ دعماً لي بكلمة طيبة ووفاء حقيقي، فخففن عني الكثير.

إلى نفسي...

إلى من لم تستسلم رغم التعب، وواصلت الطريق حتى وصلت إلى هذه اللحظة.

وأخيراً...

إلى كل من آمن بي ووقف بجانبني، هذا النجاح لكم أيضاً.

علام هبة

Remerciements

Nous rendons grâce à Dieu Tout-Puissant qui nous a accordé la force, la patience et la volonté nécessaires pour mener à bien ce travail et surmonter les difficultés rencontrées tout au long de notre parcours académique.

Nous exprimons nos sincères remerciements à notre encadreur **Dr. Nasseredinne Mouhoub** pour ses précieux conseils, ses orientations pertinentes ainsi que son suivi constant, qui ont grandement contribué à la réalisation de ce mémoire.

Nous adressons également nos remerciements aux membres du jury d'avoir accepté d'évaluer ce travail, ainsi qu'à l'ensemble des enseignants du département qui ont contribué à notre formation académique et scientifique tout au long de nos études.

Nous tenons à exprimer notre profonde gratitude à nos familles pour leur soutien inconditionnel, leur patience et leurs encouragements permanents, qui ont été un véritable pilier durant tout notre parcours.

Enfin, nous remercions toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail, ne serait-ce que par un conseil ou un mot d'encouragement.

Contenu

Liste des figures	vii
Liste des tableaux	vii
Introduction	1
Chapitre 1 Généralités sur les systèmes.....	2
1.1. Introduction	2
1.2. Définition d'un système	2
1.3. Les Caractéristiques des systèmes.....	3
1.4. Classification des systèmes	5
1.4.1. Les systèmes ouverts et fermés	5
1.4.2. Les systèmes simples et complexes	6
1.4.3. Les systèmes dynamiques et statiques.....	6
1.4.4. Les systèmes naturels et artificiels	6
1.5. Les systèmes dynamiques et complexes.....	6
1.5.1. Les systèmes dynamiques.....	6
1.5.2. Les systèmes complexes	7
1.5.3. Lien entre dynamique et complexité	7
1.6. Modélisation des systèmes	7
1.7. Problématique du diagnostic des systèmes	8
1.8. Conclusion	9
Chapitre 2 Diagnostic par reconnaissance de formes.....	10
2.1. Introduction	10
2.2. Définitions.....	11
2.3. Les différentes approches de diagnostic	12
2.4. Le diagnostic par reconnaissance de formes	13
2.4.1. Le principe de base	13

2.4.2.	Analyse des observations	14
2.4.3.	Réduction de la dimension de l'espace de représentation	14
2.4.4.	L'espace de décision	15
2.4.5.	La procédure de décision.....	17
2.4.6.	Phase d'exploitation	18
2.5.	Outils d'intelligence artificielle pour le diagnostic	19
2.5.1.	Réseaux de neurones.....	19
2.5.2.	L'arbre de décision	20
2.6.	Conclusion.....	21
Chapitre 3 Les colonies de fourmis		23
3.1.	Introduction	23
3.2.	Les fourmis réelles	23
3.2.1.	Principe de base	23
3.2.2.	Comportement collectif	24
3.2.3.	Stigmergie et phéromones	25
3.3.	De la fourmi réelle à la fourmi artificielle.....	26
3.3.1.	Principe de base	26
3.3.2.	La fourmi artificielle.....	26
3.4.	Algorithme de colonie de fourmis (ACO)	26
3.4.1.	Principe general	26
3.4.2.	Paramètres de l'algorithme	27
3.4.3.	Variantes principales de l'ACO (AS, ACS, MMAS)	28
3.5.	Application au diagnostic des systèmes.....	29
3.5.1.	Lien avec la reconnaissance de formes	29
3.5.2.	Apport de l'ACO au diagnostic des systèmes.....	29
3.6.	Méthode alternative de diagnostic (Arbre de décision)	30
3.6.1.	Présentation générale	30
3.6.2.	Exemple d'application.....	31

3.6.3. Comparaison avec l'ACO	31
3.7. Conclusion	33
Chapitre 4 Application et réalisation	34
4.1. Introduction	34
4.2. Environnement de travail	34
4.2.1. Langage et outils utilisés.....	35
4.2.2. Architecture générale de l'application	36
4.3. Présentation des systèmes étudiés	39
4.3.1. Réseau informatique	39
4.3.2. Système d'énergie industrielle	40
4.4. Implémentation de l'algorithme ACO	41
4.4.1. Structure générale de l'algorithme	42
4.4.2. Paramètres utilisés	43
4.4.3. Processus de détection des défauts.....	44
4.5. Résultats et discussion.....	45
4.5.1. Visualisation des résultats	45
4.5.2. Analyse des performances	46
4.5.3. Limites et perspectives.....	46
4.6. Conclusion	47
Conclusion	48
Références.....	50

Liste des figures

Figure 1.1: La forme générale du système	3
Figure 2.1: Le processus de la reconnaissance de formes [33]	14
Figure 2.2: Structure du réseau de type RBF pour le diagnostic [8]	20
Figure 2.3: Exemple d'un arbre de décision pour le diagnostic	21
Figure 3.1: Principe de recherche du plus court chemin dans l'algorithme ACO.....	24
Figure 4.1: Logo de python.....	35
Figure 4.2: L'architecture générale de l'application	37
Figure 4.3: Écran d'accueil (Splash Screen) du projet avec verrouillage d'accès.....	38
Figure 4.4: L'interface principale.....	38
Figure 4.5: Topologie du réseau informatique	40
Figure 4.6: Topologie du système d'énergie industrielle	41
Figure 4.7: La structure générale de l'algorithme	43

Liste des tableaux

Tableau 2.1: Tableau comparatif des méthodes de classification	17
Tableau 3.1: Comparaison des caractéristiques de l'ACO et de l'arbre de décision	32
Tableau 4.1: Principaux paramètres influençant le fonctionnement de l'ACO	44

LES ACRONYMES

ACO : Ant Colony Optimization (Optimisation par Colonie de Fourmis)

RDF : Reconnaissance de Formes (Pattern Recognition)

Introduction

Face à la complexité croissante des systèmes modernes, qu'ils soient informatiques, énergétiques ou industriels, la question du diagnostic fiable et rapide des défaillances est devenue un enjeu majeur. En effet, la moindre perturbation non détectée à temps peut engendrer des conséquences graves, allant de la simple dégradation des performances jusqu'à l'arrêt total du système. C'est dans ce contexte que les méthodes intelligentes de diagnostic trouvent toute leur pertinence.

Les systèmes dynamiques et complexes se distinguent par leur évolution continue dans le temps, leurs interactions non linéaires et leur sensibilité aux conditions initiales, ce qui rend leur modélisation mathématique classique particulièrement difficile, voire impossible dans certains cas. Ces caractéristiques imposent le recours à des approches alternatives, plus flexibles et capables de traiter des données volumineuses et incertaines.

Parmi ces approches, le diagnostic par reconnaissance de formes s'est imposé comme une méthodologie efficace, permettant de comparer l'état courant d'un système à des états de référence afin d'identifier automatiquement toute anomalie. Combinée aux techniques d'intelligence artificielle, cette approche offre des capacités d'apprentissage, d'adaptation et de généralisation particulièrement adaptées aux systèmes complexes.

C'est dans ce cadre que s'inscrit le présent travail, qui propose d'appliquer l'algorithme de colonie de fourmis (Ant Colony Optimization — ACO) au diagnostic de systèmes dynamiques et complexes. Inspiré du comportement collectif des fourmis réelles, cet algorithme exploite le mécanisme des phéromones pour explorer efficacement l'espace de recherche et identifier les composants défaillants de manière progressive et adaptative.

Ce mémoire est structuré en quatre chapitres complémentaires. Le premier chapitre pose les bases conceptuelles relatives aux systèmes, à leur classification, à leur modélisation et à la problématique de leur diagnostic. Le deuxième chapitre présente le diagnostic par reconnaissance de formes, en détaillant ses différentes étapes et les méthodes de classification associées. Le troisième chapitre est consacré aux algorithmes de colonie de fourmis, en exposant leurs fondements biologiques, leurs principes de fonctionnement et leur application au diagnostic, avec une comparaison théorique avec les arbres de décision.

Enfin, le quatrième chapitre présente l'implémentation pratique de l'application développée ainsi que les résultats expérimentaux obtenus sur deux systèmes distincts.

Chapitre1 Généralités sur les systèmes

1.1. Introduction

Les systèmes constituent un cadre scientifique global pour l'étude des phénomènes complexes et interconnectés. Ils sont appréhendés comme des unités intégrées composées d'éléments interdépendants et en interaction, évoluant dans un environnement donné et produisant des comportements globaux qui ne peuvent être réduits à leurs composants isolés. L'approche systémique permet ainsi une compréhension approfondie des relations internes et externes, et met en évidence l'importance de l'organisation, de l'équilibre et de la rétroaction dans le maintien de la performance et l'adaptation aux changements.

Qu'ils soient naturels, comme les organismes vivants, industriels, comme les organisations et les entreprises, ou sociaux, comme les communautés humaines, les systèmes se distinguent par leur capacité à évoluer et à manifester des propriétés nouvelles issues des interactions internes et des liens avec leur environnement. L'étude des systèmes ne se limite donc pas à la description de leurs éléments, mais vise à analyser leurs dynamiques, à comprendre leurs mécanismes et à expliquer leur comportement global.

Ce chapitre introductif a pour objectif de poser les bases scientifiques générales de la compréhension des systèmes, en mettant en lumière leurs caractéristiques essentielles et leur rôle dans la recherche scientifique. Il prépare ainsi le terrain pour l'exploration des dynamiques, de la complexité et des méthodes d'optimisation qui seront développées dans les chapitres suivants

1.2. Définition d'un système

Un système est généralement défini comme un ensemble d'éléments matériels et immatériels en interaction mutuelle, reliés par des mécanismes de contrôle qui assurent leur orientation vers un objectif déterminé. Autrement dit, il s'agit d'une structure organisée qui reçoit des entrées de son environnement, les traite à travers des processus internes, puis produit des sorties qui influencent à leur tour cet environnement [1].

Parmi les définitions les plus complètes dans ce cadre, un système peut être décrit comme un cadre global et intégré qui coordonne l'utilisation des ressources disponibles énergie, informations et matières en vue d'atteindre un ensemble d'objectifs interdépendants [1]. Ce qui

distingue cette définition, c'est qu'elle met l'accent sur la dimension fonctionnelle du système, et non sur la simple description de ses composants.

Sur la base de ce qui précède, on peut affirmer que toute entité naturelle ou artificielle peut être considérée comme un système, dès lors qu'elle interagit avec son environnement, l'influence et en est influencée. Un système est, en définitive, un ensemble d'éléments en interaction qui doivent maintenir un certain équilibre pour atteindre une finalité donnée [1].

En résumé, tout système se compose fondamentalement d'entrées et de sorties, séparées par un processus de transformation qui assure le passage des premières aux secondes et c'est ce principe qui guidera notre analyse dans les chapitres suivants.

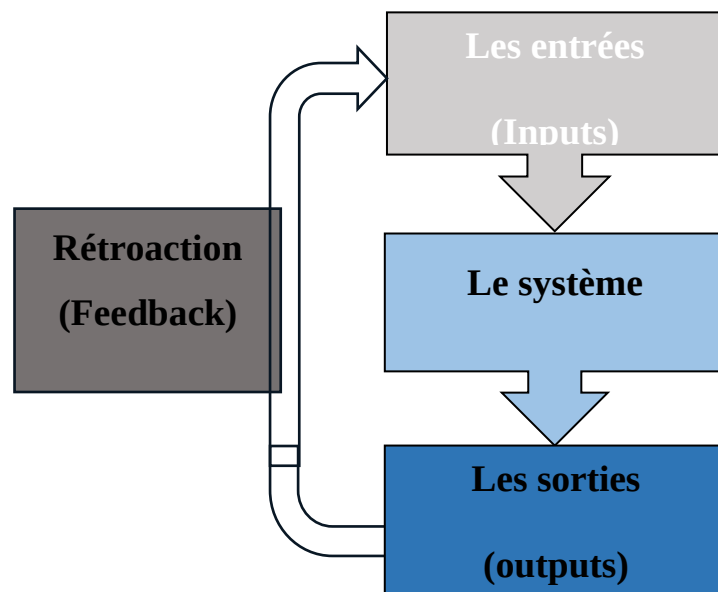


Figure 1.1: La forme générale du système.

1.3. Les Caractéristiques des systèmes

Les caractéristiques des systèmes constituent des éléments fondamentaux pour comprendre leur nature et leur comportement, car elles permettent d'analyser la manière dont les composants d'un système interagissent entre eux et avec l'environnement externe. La théorie générale des systèmes, développée par Ludwig von Bertalanffy, s'est particulièrement intéressée à l'étude de ces caractéristiques afin de proposer une vision globale et intégrée des systèmes, quels que soient leurs types [2].

- **Interdépendance**

Les systèmes se caractérisent par l'existence de fortes relations d'interdépendance entre leurs composants, de sorte qu'aucun élément ne peut être étudié indépendamment des autres. Toute modification affectant un composant entraîne nécessairement des effets directs ou indirects sur les autres composants. Cela fait du système une entité interconnectée reposant sur l'interaction continue entre ses parties.

- **Totalité / Holisme**

L'une des caractéristiques les plus importantes des systèmes est qu'ils doivent être étudiés comme un tout intégré, et non comme une simple juxtaposition d'éléments séparés. Un système possède des propriétés globales qui ne peuvent être comprises uniquement par l'analyse de ses parties, mais plutôt par l'étude des relations qui les relient entre elles. Ce principe est souvent exprimé par l'idée que « le tout est supérieur à la somme de ses parties ».

- **Finalité / Objectif**

Tout système est conçu pour atteindre un objectif ou un ensemble d'objectifs prédéfinis. L'ensemble des composants et des activités du système est orienté vers la réalisation de cet objectif, qu'il s'agisse d'un système naturel ou artificiel. L'objectif constitue un élément essentiel pour déterminer le comportement du système et évaluer ses performances.

- **Frontières**

Chaque système possède des frontières qui le séparent de son environnement externe. Ces frontières définissent ce qui appartient au système et ce qui n'en fait pas partie. Elles peuvent être physiques ou abstraites, mais elles jouent un rôle essentiel dans l'organisation des interactions entre le système et son environnement, ainsi que dans la délimitation de son champ d'action.

- **Interaction avec l'environnement**

Un système ne fonctionne pas de manière isolée, mais interagit en permanence avec l'environnement externe qui l'entoure. Il reçoit des entrées provenant de cet environnement (telles que l'information ou l'énergie) et produit des sorties qui influencent à leur tour cet environnement. Cette interaction est essentielle pour assurer la continuité du système et son adaptation aux changements.

- **Rétroaction (Feedback)**

La rétroaction constitue l'un des mécanismes de contrôle les plus importants au sein des systèmes, car elle permet au système de surveiller ses résultats et de corriger ses écarts. Le système réutilise les informations issues de ses sorties afin d'améliorer ses performances et de garantir l'atteinte de ses objectifs de manière plus efficace.

- **Dynamisme et évolution**

De nombreux systèmes, en particulier les systèmes complexes, sont caractérisés par leur nature dynamique, c'est-à-dire qu'ils évoluent et changent au cours du temps. Le comportement de ces systèmes dépend d'un ensemble de facteurs internes et externes, ce qui leur permet de s'adapter à différentes conditions.

D'après ce qui précède, il apparaît que les systèmes ne sont pas de simples ensembles d'éléments, mais des entités complexes caractérisées par un ensemble de propriétés qui régissent leur comportement et leurs performances. L'étude de ces caractéristiques permet de mieux comprendre le fonctionnement des systèmes et de les analyser, notamment dans le cadre des systèmes complexes et dynamiques qui constituent l'objet principal de ce travail.

1.4. Classification des systèmes

Les systèmes peuvent être classés selon plusieurs critères, chaque classification reflétant la nature des relations entre les éléments du système et la manière dont il interagit avec son environnement [1][2] :

1.4.1. Les systèmes ouverts et fermés

Un système ouvert interagit avec son environnement externe par l'échange de matière, d'énergie ou d'information. Exemple : une entreprise économique qui reçoit des intrants (matières premières, informations du marché) et produit des extrants (biens, services, données). Ce type de système se distingue par sa capacité d'adaptation aux changements environnementaux.

Un système fermé, au contraire, n'échange rien avec son environnement. Il est rare dans la réalité et utilisé surtout comme modèle théorique pour simplifier l'analyse. Exemple hypothétique : une expérience chimique dans un récipient parfaitement isolé.

1.4.2. Les systèmes simples et complexes

Dans Un système simple est constitué d'un nombre limité d'éléments et de relations, ce qui rend son comportement prévisible et facile à analyser.

Un système complexe comprend un grand nombre d'éléments interconnectés par des relations non linéaires et présente des comportements émergents (Emergent behavior) qui ne peuvent être déduits directement des propriétés des éléments individuels. Exemple : les réseaux sociaux ou les écosystèmes, où les interactions entre individus ou organismes produisent de nouveaux schémas.

1.4.3. Les systèmes dynamiques et statiques

Un système dynamique évolue dans le temps selon des lois ou des équations mathématiques, comme le mouvement des planètes ou l'évolution des marchés financiers.

Un système statique reste inchangé tant que les conditions externes ne varient pas, comme une structure architecturale ou une base de données non mise à jour.

1.4.4. Les systèmes naturels et artificiels

Les systèmes naturels existent dans la nature, tels que les organismes vivants et les écosystèmes, et sont souvent complexes et non linéaires.

Les systèmes artificiels sont créés par l'homme, comme les systèmes industriels ou informatiques, et sont conçus pour atteindre des objectifs précis.

1.5. Les systèmes dynamiques et complexes

Les systèmes dynamiques et complexes représentent une catégorie particulière d'une grande importance dans l'étude et le diagnostic [1].

1.5.1. Les systèmes dynamiques

Un système dynamique est caractérisé par une évolution dans le temps selon des lois ou des équations mathématiques. Son état change continuellement en fonction des variables internes et des influences externes. Exemples : la trajectoire des planètes, les circuits électriques ou les modèles économiques. Ces systèmes nécessitent une modélisation précise afin de comprendre leur comportement et anticiper leurs évolutions.

1.5.2. Les systèmes complexes

Un système complexe se distingue par un grand nombre d'éléments interconnectés, des interactions non linéaires et une forte sensibilité aux conditions initiales. Il présente des comportements émergents qui ne peuvent être prédits uniquement à partir des propriétés des composants individuels. Exemples : les réseaux neuronaux, les écosystèmes ou Internet. L'analyse de ces systèmes est difficile car ils combinent incertitude, diversité et interdépendance.

1.5.3. Lien entre dynamique et complexité

Dans la réalité, un système complexe est généralement dynamique : son évolution dans le temps rend son diagnostic particulièrement délicat. C'est précisément dans ce contexte que les méthodes d'intelligence artificielle, telles que l'optimisation par colonies de fourmis, trouvent leur pertinence pour analyser, modéliser et diagnostiquer ces systèmes.

1.6. Modélisation des systèmes

Lorsqu'un chercheur ou un ingénieur est confronté à un système complexe, il se trouve face à un défi fondamental : comment comprendre ce qu'il ne peut ni toucher ni maîtriser directement ? La réponse réside dans la modélisation. Un modèle n'est pas simplement un outil mathématique, c'est avant tout une façon de penser qui permet de transformer une réalité complexe en un langage analysable, testable et prédictible [3].

La modélisation est, dans son essence, un processus de simplification réfléchi et non une simplification naïve. Elle ne vise pas à réduire la réalité, mais à en extraire ce qui est fondamental. C'est ainsi qu'un bon modèle devient un miroir fidèle de la dynamique et du comportement du système, qu'il s'agisse de suivre l'évolution de ses variables dans le temps, d'analyser sa réponse aux sollicitations extérieures, ou d'anticiper ses dérives avant qu'elles ne s'aggravent [3][4].

Les formes de modèles sont aussi variées que la nature des systèmes eux-mêmes. On distingue notamment les modèles mathématiques analytiques, fondés sur des équations différentielles et des représentations d'état, qui sont les plus précis et les plus répandus pour décrire les systèmes dynamiques. Il existe également les modèles Bond Graph, qui offrent un langage graphique unifié permettant de représenter dans un même cadre des systèmes électriques, mécaniques et thermiques de natures différentes. À cela s'ajoutent les modèles de simulation informatiquetels que ceux développés sous MATLAB et Simulink qui permettent

de reproduire le comportement du système dans un environnement virtuel sécurisé avant tout déploiement réel. Des langages de programmation comme Python sont également utilisés, offrant une grande flexibilité dans la construction de modèles de simulation et l'analyse des données des systèmes [4][5].

Cependant, la modélisation n'est pas sans difficultés réelles. Le concepteur se trouve constamment confronté à une tension entre simplicité et précision : un modèle trop simplifié peut omettre des phénomènes cruciaux, tandis qu'un modèle trop détaillé peut devenir lourd et difficilement exploitable. De plus, les systèmes non linéaires présentent une sensibilité excessive aux conditions initiales, où une légère erreur dans les données peut engendrer un écart considérable dans le comportement global du modèle [5].

En définitive, ce que la modélisation nous enseigne, c'est que la compréhension du système est une condition sine qua non à toute tentative de diagnostic ou de contrôle. Plus le modèle est précis et réaliste, plus le système de diagnostic qui s'appuie sur lui sera fiable et performant ce que les chapitres suivants de ce travail s'attacheront à démontrer [3][4].

1.7. Problématique du diagnostic des systèmes

Le diagnostic des systèmes constitue un sujet essentiel dans l'étude des systèmes dynamiques et complexes, puisqu'il vise à détecter les pannes et les déviations susceptibles d'affecter la performance et la stabilité du système [28]. Le processus de diagnostic repose sur la surveillance du comportement du système et l'analyse des données et informations générées, afin d'identifier la source de la défaillance et de prendre les mesures appropriées pour éviter son impact sur le système [28].

Avec le développement technologique et l'augmentation de la complexité des systèmes modernes, le diagnostic est devenu plus difficile en raison du grand nombre d'éléments interagissant au sein du système et de la diversité des relations entre eux, ainsi que de l'évolution continue du comportement des systèmes dynamiques [5]. De plus, la présence de relations non linéaires et d'interactions imbriquées rend la prédiction du comportement du système lors de l'apparition de pannes ou de déviations particulièrement complexe [5].

Les méthodes traditionnelles de diagnostic présentent plusieurs limites lorsqu'il s'agit de traiter des systèmes complexes, notamment en ce qui concerne la rapidité de détection des pannes et la précision dans l'identification de leurs causes. Cela a conduit à l'émergence de techniques intelligentes basées sur l'intelligence artificielle, la reconnaissance de formes et les algorithmes d'optimisation, afin d'améliorer la performance des systèmes de diagnostic [9].

Parmi les approches modernes les plus utilisées dans ce domaine, on trouve l'algorithme de la colonie de fourmis (*Ant Colony Optimization*), qui a démontré son efficacité dans le traitement des problèmes de recherche et d'optimisation au sein des systèmes complexes, grâce à sa capacité à trouver des solutions efficaces dans un temps raisonnable [6]. Pour cette raison, cet algorithme est largement employé dans les applications de diagnostic intelligent des systèmes dynamiques et complexes.

1.8. Conclusion

Ce chapitre a posé les fondements conceptuels indispensables à toute démarche d'analyse et de diagnostic des systèmes. En partant d'une définition rigoureuse du système, nous avons progressivement mis en lumière ses propriétés essentielles interdépendance, holisme, rétroaction, dynamisme qui font de lui bien plus qu'un simple assemblage d'éléments : une entité vivante, évolutive et complexe dans ses interactions.

L'exploration des différentes catégories de systèmes a permis de saisir la diversité des comportements possibles, et de comprendre pourquoi les systèmes dynamiques et complexes occupent une place centrale dans la recherche scientifique contemporaine. Leur évolution continue dans le temps, combinée à la non-linéarité de leurs interactions internes, en fait des objets d'étude particulièrement délicats, où l'intuition seule ne suffit plus.

Face à cette complexité, la modélisation s'est imposée comme l'outil incontournable qui permet de transformer le réel en un langage analysable et prédictible. Qu'il s'agisse de modèles mathématiques, de Bond Graphs ou de simulations informatiques, chaque approche constitue une tentative de saisir l'essentiel du comportement d'un système avec ses forces et ses limites inhérentes.

Enfin, la problématique du diagnostic est venue révéler les véritables enjeux de ce travail : les méthodes traditionnelles montrent leurs limites face à des systèmes toujours plus complexes, et c'est précisément là que les approches intelligentes trouvent leur raison d'être.

Fort de ces bases, le chapitre suivant s'attachera à explorer le diagnostic par reconnaissance des formes une approche qui place l'intelligence au cœur du processus de détection et d'identification des défaillances, ouvrant ainsi la voie vers des systèmes de diagnostic plus robustes, plus précis et mieux adaptés aux exigences des systèmes industriels modernes.

Chapitre 2 Diagnostic par reconnaissance de formes

2.1. Introduction

Les systèmes complexes et dynamiques sont parmi les plus difficiles à modéliser et à analyser, en raison de leur nature non linéaire et de leur instabilité durant le fonctionnement. Ces systèmes sont généralement influencés par un grand nombre de variables qui déterminent leur état, ce qui rend leur représentation précise à l'aide de modèles mathématiques classiques très compliquée, voire impossible. Ainsi, il est devenu nécessaire de développer des méthodologies alternatives plus flexibles, parmi lesquelles le diagnostic basé sur la reconnaissance de formes (Pattern Recognition) occupe une place importante.

La fonction de surveillance dans ces systèmes consiste essentiellement à vérifier l'état de fonctionnement. Elle se divise en deux étapes principales : la détection et le diagnostic. La détection vise à déterminer si le système fonctionne normalement ou présente une anomalie, tandis que le diagnostic cherche à identifier les composants défaillants et à en déterminer les causes à partir d'un ensemble de symptômes observés. D'un point de vue général, cette démarche peut être assimilée à un problème de classification, puisqu'il s'agit de comparer un vecteur de données représentant l'état actuel du système avec des vecteurs de référence traduisant l'historique de son fonctionnement.

De nombreux chercheurs ont mis en évidence l'importance de la reconnaissance de formes dans divers domaines où les systèmes sont de nature complexe, tels que la reconnaissance d'images, la reconnaissance de l'écriture manuscrite, le diagnostic médical et le diagnostic industriel. La force de cette approche réside dans plusieurs avantages : rapidité de traitement lors de la reconnaissance d'une nouvelle information, capacité à gérer des données incertaines ou imprécises, ainsi que possibilité de suivre l'évolution du fonctionnement du système dans le temps.

Sur le plan pratique, le diagnostic par reconnaissance de formes constitue un outil puissant, car il combine l'analyse statistique et l'expérience accumulée, permettant de prendre des décisions plus précises et objectives. Par exemple, dans le domaine médical, cette technique est utilisée pour analyser des images radiologiques ou des IRM afin de détecter des pathologies. Dans le domaine industriel, elle permet de surveiller et diagnostiquer les pannes dans des machines complexes en comparant les données actuelles avec des données historiques. Enfin,

dans le domaine de l'intelligence artificielle, la reconnaissance de formes est devenue une base incontournable pour des applications telles que la reconnaissance faciale, vocale ou textuelle.

En outre, la reconnaissance de formes est une méthodologie capable de traiter des données massives (Big Data), en extrayant les relations internes entre les éléments et en les transformant en informations interprétables. Cela en fait un outil essentiel à l'ère de la transformation numérique, où le besoin d'avoir des systèmes intelligents capables de diagnostiquer rapidement et avec précision ne cesse de croître.

Cependant, malgré ses nombreux avantages, cette approche présente certaines limites. Elle requiert souvent de grandes quantités de données pour l'apprentissage des systèmes, ce qui peut constituer un défi. De plus, la présence de données bruitées ou incomplètes complique le processus de diagnostic. Enfin, l'intervention d'experts humains reste indispensable pour interpréter les résultats et garantir leur fiabilité. Néanmoins, les progrès constants dans le domaine de l'intelligence artificielle et de l'apprentissage automatique contribuent progressivement à surmonter ces obstacles.

2.2. Définitions

Cette section présente les principaux concepts liés au diagnostic par reconnaissance de formes qui seront utilisés dans ce travail [7].

- **Le diagnostic** est une opération visant à déterminer les défauts pouvant apparaître dans un système, à travers leur détection, leur analyse et l'identification de leur nature.
- **La détection** elle représente la première étape du diagnostic et consiste à détecter l'existence d'un défaut ou d'un comportement anormal dans le système.
- **L'identification** elle consiste à caractériser précisément le défaut détecté et à déterminer ses propriétés.
- **La localisation** c'est une opération qui permet de déterminer l'emplacement du défaut après sa détection.
- **La défaillance** elle correspond à la perte de la capacité d'un système à assurer correctement la fonction demandée.
- **Le défaut** il représente un écart entre le comportement réel du système et le comportement de référence ou attendu.
- **La panne** elle désigne l'arrêt ou l'incapacité d'un système à accomplir sa fonction à cause d'un défaut ou d'une défaillance.

- **La reconnaissance de formes (RdF)** regroupe un ensemble de techniques et d'algorithmes permettant de classifier des données ou des formes à partir de leurs caractéristiques et de leur comparaison avec des modèles de référence.
- **Le vecteur forme** il s'agit d'un ensemble de caractéristiques ou de paramètres représentant une forme ou un état donné. Ces paramètres peuvent être numériques ou symboliques.
- **Le prototype** le prototype est un modèle représentatif d'une classe défini à partir de valeurs caractéristiques.
- **La classe** une classe représente un ensemble d'éléments partageant les mêmes caractéristiques ou comportements.
- **La classification d'un objet** elle consiste à affecter un objet ou une observation à une classe donnée en fonction de ses caractéristiques.

2.3. Les différentes approches de diagnostic

Le thème des **différentes approches de diagnostic** constitue l'un des sujets fondamentaux dans le domaine de la surveillance des systèmes complexes. Le diagnostic vise à établir une relation entre les symptômes observés dans le système, les défaillances et leurs causes réelles, afin d'identifier l'origine du dysfonctionnement et de prendre les décisions appropriées pour assurer le bon fonctionnement du système [8]. Avec l'augmentation de la complexité des systèmes et la multiplicité de leurs composants, plusieurs approches de diagnostic ont été développées en fonction de la nature des informations disponibles [8]. Ces approches peuvent être classées en trois grandes catégories.

La première catégorie concerne les approches à base de modèles (approches à base de modèles), qui consistent à construire un modèle mathématique ou logique représentant le comportement normal du système, puis à comparer ce modèle au comportement réel afin de détecter toute déviation indiquant une défaillance. Cette approche se caractérise par une grande précision et une capacité d'interprétation des résultats, mais elle nécessite une connaissance approfondie de la structure du système, ce qui peut être difficile dans le cas des systèmes complexes [8].

La deuxième catégorie regroupe les approches à base de connaissances (approches à base de connaissances), qui reposent sur l'expertise humaine et les bases de connaissances telles que les systèmes experts et l'analyse des modes de défaillance. Dans ce cas, les défaillances sont déduites à partir de règles logiques construites à partir de l'expérience. Cette approche est

relativement simple à mettre en œuvre, mais elle reste limitée par la disponibilité et la fiabilité des connaissances [8].

La troisième catégorie correspond aux approches à base de données (approches à base de données), qui exploitent les données historiques du système à l'aide de techniques d'intelligence artificielle et d'apprentissage automatique, telles que les réseaux de neurones et les algorithmes de classification, afin d'extraire des patterns permettant d'identifier les défaillances et de localiser leurs causes. Cette approche s'est révélée particulièrement efficace pour les systèmes difficiles à modéliser mathématiquement [9]. Par ailleurs, le diagnostic des systèmes peut être considéré comme un problème de classification, où l'état courant du système est comparé à des états de référence afin de déterminer le type de défaillance et ses causes. Cela nécessite le choix de paramètres pertinents permettant de distinguer les différentes conditions de fonctionnement, ce qui contribue à améliorer la précision et l'efficacité du diagnostic [8][11].

Compte tenu des limites de chaque approche prise isolément, les travaux de recherche actuels s'orientent vers l'utilisation de méthodes hybrides combinant les modèles, les connaissances et les données, afin d'obtenir un diagnostic plus précis et plus fiable, adapté à la complexité des systèmes modernes [9].

2.4. Le diagnostic par reconnaissance de formes

2.4.1. Le principe de base

Un problème de diagnostic peut se définir comme un problème de reconnaissance de formes. Il a pour but la reconnaissance d'une forme parmi différentes possibilités à partir d'observations bruitées. Il existe plusieurs approches : la reconnaissance de forme structurelle ou syntaxique qui exploite les relations entre les composants de la forme et la reconnaissance de formes de type numérique (statistique, floue, etc.) qui exploite des modélisations de formes probabilistes ou floues. Une forme est représentée par un ensemble de paramètres ou attributs caractéristiques qui peuvent être numériques ou symboliques. Un prototype est défini par des valeurs précises de l'ensemble des paramètres caractéristiques d'une situation particulière (défaillance dans notre contexte). Une classe est définie par un ensemble possible de valeurs des attributs et est représentée par son prototype. La classification d'un objet (la reconnaissance d'une forme) parmi M classes est la décision d'affecter l'objet à une classe particulière. Les classes peuvent être définies de façon probabiliste ou encore comme des classes floues. Dans le cadre du diagnostic, le prototype 1 peut être lié au fonctionnement normal du procédé alors

que les prototypes 2 et 3 peuvent représenter deux modes de défauts distincts. Le diagnostic consiste à décider de quel prototype cette forme est la plus proche [10].

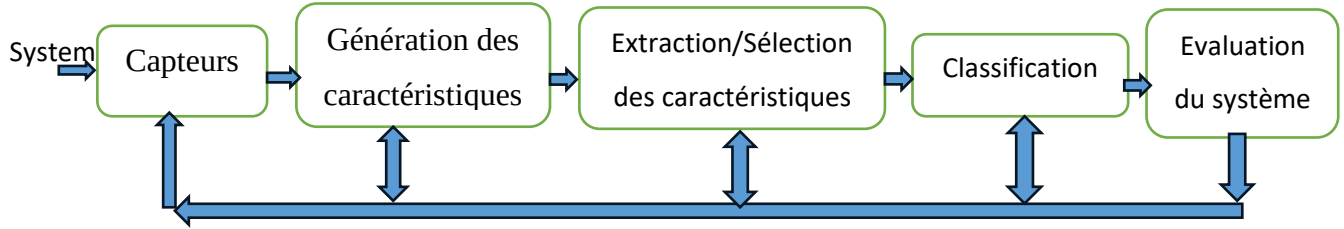


Figure 2.1: Le processus de la reconnaissance de formes [33].

2.4.2. Analyse des observations

Dans une application, si les données acquises sont sous formes de signaux, alors il est indispensable d'en extraire des paramètres numériques. Ces paramètres, qui par ailleurs constituent le vecteur forme, doivent pouvoir décrire le fonctionnement du système. Dans cette phase, on doit également définir les différentes classes qui représentent les modes de fonctionnement du système étudié. Donc, On obtient à la fin de cette phase : un ensemble de N entités regroupées en M classes qui représente les données d'apprentissage [8].

2.4.3. Réduction de la dimension de l'espace de représentation

L'efficacité des systèmes de diagnostic dépend du choix des paramètres les plus pertinents, car l'utilisation d'un grand nombre de variables ralentit le processus d'identification et dégrade les performances [8]. C'est pourquoi on recourt à des techniques de réduction de dimension visant à diminuer la taille du vecteur d'état et à ne conserver que les paramètres essentiels, ce qui améliore la rapidité et la précision du diagnostic.

Ces méthodes se divisent en deux catégories principales :

- **Méthodes d'extraction** : elles consistent à créer de nouveaux paramètres à partir des variables initiales, par combinaison ou transformation. L'exemple le plus connu est l'Analyse en Composantes Principales (ACP), qui transforme un grand nombre de variables en un petit nombre de composantes principales résumant l'essentiel de l'information, ce qui facilite le diagnostic [6].
- **Méthodes de sélection** : elles consistent à retenir un sous-ensemble des paramètres originaux les plus discriminants entre les états, en utilisant des algorithmes tels que les algorithmes génétiques [13] ou l'optimisation par colonie de fourmis (ACO) [14], afin

de choisir les variables les plus utiles pour obtenir une meilleure performance de classification.

Toute méthode de sélection de paramètres repose sur quatre éléments essentiels [18] :

- **Un ensemble de départ** représentant les paramètres initialement utilisés par la procédure de recherche. Cet ensemble peut être vide, contenir tous les paramètres ou bien un sous-ensemble aléatoire.
- **La procédure de recherche**, élément central de toute méthode de sélection, qui retourne le sous-ensemble de paramètres répondant le mieux au critère de qualité.
- **La fonction d'évaluation**, qui détermine la qualité de la classification obtenue en utilisant un sous-ensemble de paramètres.
- **Un critère d'arrêt**, utilisé pour mettre fin à la procédure de recherche, dépendant soit de la fonction d'évaluation, soit des paramètres de configuration définis par l'utilisateur.

L'objectif final de ces approches est de réduire le nombre de paramètres tout en conservant la capacité de distinguer les différents modes de fonctionnement, permettant ainsi un diagnostic plus rapide et plus efficace [18].

Nous proposons dans le deuxième chapitre une nouvelle méthode de sélection des paramètres basée sur une variante de l'algorithme de colonie de fourmis, et nous montrons par la suite que les résultats obtenus par cette approche sont comparables à ceux obtenus avec les algorithmes génétiques [6].

2.4.4. L'espace de décision

La phase de l'espace de décision constitue un pilier fondamental dans les systèmes de diagnostic basés sur la reconnaissance des formes. Les données d'apprentissage y sont réparties en groupes homogènes selon diverses méthodes de classification, puis chaque groupe est associé à un état de fonctionnement précis, qu'il s'agisse d'un régime normal ou d'une condition de défaillance. L'objectif est d'organiser les données en classes cohérentes facilitant la prise de décision et garantissant la précision du diagnostic des défauts [11][12].

La classification est définie comme un mécanisme permettant de regrouper les éléments similaires au sein d'une même classe, en s'appuyant sur des critères de ressemblance ou de dissemblance entre les données. Elle se décline en deux formes : la classification exclusive, où chaque élément n'appartient qu'à une seule classe, et la classification non exclusive, qui autorise l'appartenance d'un même élément à plusieurs classes simultanément ; une configuration

particulièrement adaptée aux systèmes complexes où les types de défauts peuvent se chevaucher [11].

Il convient également de distinguer la classification supervisée de la classification non supervisée. Dans la première, les classes sont connues à l'avance, ce qui permet au système d'apprendre à partir de données étiquetées. Dans la seconde, aucune information préalable sur les classes n'est disponible ; le système s'appuie uniquement sur les degrés de similarité ou de dissimilarité entre les données pour découvrir automatiquement la structure interne des groupes [12][13].

- **La classification hiérarchique**

La classification hiérarchique repose sur le calcul de distances entre les éléments de données afin de construire une structure arborescente mettant en évidence les relations entre eux.

Ce processus aboutit à un arbre dont les nœuds contiennent les données de la base d'apprentissage ; cet arbre est ensuite coupé à un niveau déterminé pour obtenir les groupes finaux [13].

Ce type de classification se divise en deux approches distinctes : une approche ascendante (agglomérative), où l'algorithme part d'éléments isolés et les fusionne progressivement selon leur degré de similitude pour former des groupes de plus en plus larges, et une approche descendante (divisive), qui part d'un groupe unique contenant l'ensemble des données et le subdivise en sous-groupes aussi distincts que possible, afin d'assurer une séparation nette entre eux [14].

Malgré la capacité de ces méthodes à révéler les relations internes des données, elles se heurtent à des contraintes computationnelles importantes lors du traitement de grandes bases de données, en raison d'une complexité algorithmique souvent quadratique ou supérieure [14].

- **La classification par partition**

Dans cette approche, chaque groupe représente une région homogène de l'espace des données. La plupart de ces méthodes s'appuient sur le calcul d'un centroïde pour chaque groupe, puis affectent chaque élément au centroïde le plus proche selon la mesure de distance adoptée [13].

Ces techniques visent à maximiser la cohésion interne de chaque groupe tout en minimisant la similarité entre les groupes distincts, ce qui les rend particulièrement adaptées aux

applications de diagnostic industriel et à la détection de patterns anormaux au sein de systèmes complexes [13][14].

2.4.5. La procédure de décision

Une fois la phase de classification achevée, la procédure de décision constitue l'étape finale du système de diagnostic. Elle attribue à chaque nouvelle observation son état de fonctionnement correspondant — régime normal ou type de défaillance. Deux mécanismes de rejet permettent de gérer les cas ambigus [11][15] :

- **Rejet d'ambiguïté (ω_0)** : S'applique lorsqu'une observation présente des similarités avec plusieurs classes sans appartenir clairement à aucune d'entre elles. Le système l'oriente vers une classe de réserve dans l'attente d'informations complémentaires [15].
- **Rejet de distance (ω_1)** : S'applique lorsqu'une observation est éloignée de toutes les classes connues, suggérant qu'elle appartient à une catégorie inconnue — situation fréquente dans les systèmes industriels présentant de nouveaux types de défauts [15].

Tableau 2.1 : Tableau comparatif des méthodes de classification

Méthode	Principe	Quand l'utiliser
Bayésienne	Affecte l'observation à la classe dont la probabilité conditionnelle est la plus élevée.	Données structurées avec une distribution probabiliste connue [11][14].
Parzen (noyaux)	Estime la densité de distribution à partir des points voisins selon leur proximité.	Données sans forme de distribution préalable connue [16].
k plus proches voisins (k-NN)	Classe selon la majorité parmi les k points d'apprentissage les plus proches.	Simple et flexible ; adaptée aux structures complexes [11][14].
Discriminant linéaire de Fisher	Trouve la direction qui maximise la séparation entre classes et minimise la dispersion interne.	Séparation linéaire claire entre deux classes ou plus [17].

Sur le plan méthodologique, ces procédures se divisent en deux grandes approches : une approche statistique reposant sur des modèles probabilistes (paramétriques ou non paramétriques), et une approche géométrique définissant des frontières de décision explicites entre les classes dans l'espace de décision [11][14].

2.4.6. Phase d'exploitation

- **Critères d'évaluation**

Les critères d'évaluation sont utilisés pour mesurer les performances du système de diagnostic et vérifier la qualité des résultats obtenus [18].

Pour réaliser cette évaluation, un ensemble de données de test est utilisé. Les résultats réels de ces données sont connus à l'avance afin de comparer les résultats du système avec les réponses correctes [19][20].

Plusieurs critères sont utilisés dans le domaine de la classification, notamment :

- La précision
- Le rappel,
- L'exactitude,
- Le taux d'erreur.

Ces critères permettent de connaître la capacité du système à fournir des résultats corrects et à réduire les erreurs de classification [20].

Ils sont également largement utilisés dans les systèmes de reconnaissance et de classification afin d'analyser les performances du classificateur et d'améliorer son efficacité [19][20].

Dans le cas d'une classification multi-classes, ces critères sont calculés séparément pour chaque classe puis une moyenne générale est réalisée afin d'obtenir une évaluation globale du système [20].

- **Méthodes d'évaluation**

Plusieurs méthodes sont utilisées pour évaluer les performances des systèmes de classification et estimer la qualité des résultats obtenus [18].

Parmi ces méthodes, on trouve :

- ✓ La méthode de re-substitution
- ✓ La méthode Holdout
- ✓ La validation croisée,
- ✓ La méthode Leave-one-out,
- ✓ La méthode de rééchantillonnage.

Ces méthodes se différencient par la manière d'utiliser les données et de mesurer les performances du système, mais leur objectif principal reste l'obtention d'une évaluation

générale de l'efficacité du classificateur et de la qualité des résultats de classification [18]. Les méthodes d'évaluation représentent une étape importante dans les systèmes de diagnostic et de classification, car elles permettent d'identifier les points forts et les limites du système afin d'améliorer ses performances.

2.5. Outils d'intelligence artificielle pour le diagnostic

Ces dernières années, les méthodes classiques sont devenues moins utilisées dans le domaine du diagnostic industriel. Les chercheurs se sont orientés vers les techniques d'intelligence artificielle grâce à leur efficacité dans l'analyse des systèmes industriels et la détection des défauts [21].

Ces techniques peuvent être appliquées à différents types de systèmes industriels, qu'ils soient simples, complexes ou dynamiques. Elles présentent également une importante capacité d'apprentissage même avec une quantité réduite de données. Parmi les principales méthodes utilisées, on retrouve les réseaux de neurones et l'arbre de décision [22].

2.5.1. Réseaux de neurones

Les réseaux de neurones s'inspirent du fonctionnement biologique du neurone et reposent sur des modèles mathématiques utilisant des neurones artificiels [22].

Ils possèdent une grande capacité de classification et de diagnostic des systèmes complexes. De plus, ces réseaux sont capables d'apprendre et de généraliser les connaissances à de nouvelles données inconnues. Pour cette raison, ils sont largement employés dans les applications de diagnostic industriel et les systèmes automatisés de prise de décision.

Les réseaux de neurones sont constitués de plusieurs couches interconnectées : une couche d'entrée, une ou plusieurs couches cachées, une couche de sortie.

La couche d'entrée représente les données ou observations à traiter, et le nombre de neurones dans cette couche correspond au nombre de caractéristiques des données introduites. Les couches cachées assurent le traitement des informations ainsi que l'extraction des caractéristiques importantes liées aux différentes classes. La couche de sortie fournit le résultat final de la classification en indiquant le degré d'appartenance de chaque observation à une classe donnée.

Lorsqu'une nouvelle donnée est introduite dans le réseau, celui-ci l'analyse et la compare aux modèles appris afin d'identifier la classe correspondante. Durant la phase d'exploitation, le

réseau devient capable de classifier automatiquement de nouvelles observations, tandis que les données ambiguës peuvent être rejetées.

Malgré leurs nombreux avantages dans le diagnostic industriel, les réseaux de neurones nécessitent une phase d'apprentissage souvent longue et complexe, demandant un temps de calcul important [22]. En outre, ces méthodes exigent une base de données variée représentant les différentes conditions de fonctionnement ainsi que les défauts possibles du système étudié.

Les performances du réseau peuvent diminuer lorsqu'il est confronté à des données situées hors de l'espace d'apprentissage. Enfin, l'ajout de nouvelles données nécessite généralement une nouvelle phase d'entraînement.

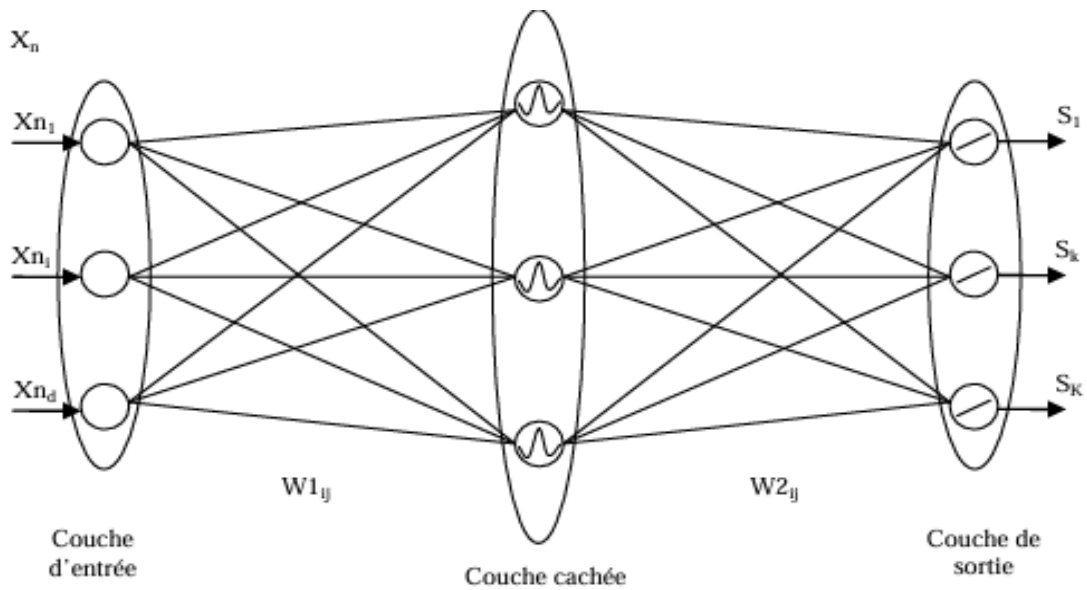


Figure 2.2: Structure du réseau de type RBF pour le diagnostic [8].

2.5.2. L'arbre de décision

L'arbre de décision est l'une des techniques d'intelligence artificielle utilisées dans le diagnostic et la prise de décision. Il repose sur une série de questions séquentielles permettant d'aboutir à un résultat ou à un diagnostic précis. Cette technique est représentée sous la forme d'un diagramme arborescent qui part d'un nœud principal et se ramifie en plusieurs branches selon les réponses possibles [32].

L'arbre de décision se distingue par sa facilité de compréhension et d'interprétation par rapport à d'autres techniques, car il repose sur un principe simple : si une condition est vérifiée, le système passe à une décision déterminée. C'est pourquoi il est utilisé dans plusieurs domaines tels que les réseaux informatiques, les systèmes médicaux, l'industrie et l'analyse de données [32].

Dans le domaine des réseaux informatiques, l'arbre de décision peut être utilisé pour diagnostiquer les pannes techniques en analysant l'état du réseau et des équipements associés, ce qui permet d'identifier la cause du problème de manière organisée et rapide. Cette technique contribue également à faciliter la maintenance et à réduire le temps nécessaire à la détection des pannes, car elle s'appuie sur des étapes ordonnées et logiques pour parvenir à la solution appropriée.

Parmi les principaux avantages de l'arbre de décision :

- ✓ Facilité de compréhension et d'application.
- ✓ Possibilité d'interpréter les résultats de manière claire et transparente.
- ✓ Rapidité dans la prise de décision.
- ✓ Applicabilité dans plusieurs domaines techniques, médicaux et industriels.

Malgré ces avantages, l'arbre de décision peut devenir complexe lorsque le nombre de conditions et de questions est trop important, ce qui entraîne une augmentation de la taille de l'arbre et des difficultés dans sa gestion [32] .

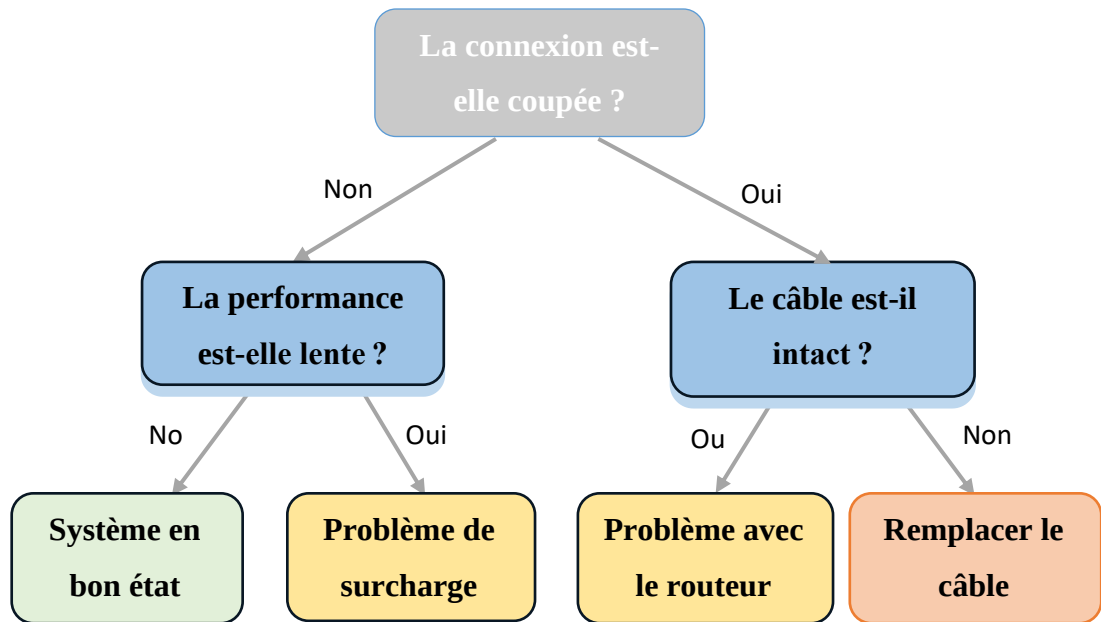


Figure 2.3: Exemple d'un arbre de décision pour le diagnostic.

2.6. Conclusion

Ce chapitre a pour objectif de présenter un ensemble de concepts fondamentaux qui seront utilisés dans la suite de ce mémoire. Dans un premier temps, nous avons abordé l'architecture générale d'un système de diagnostic basé sur la reconnaissance de formes, puis nous avons

présenté les principales techniques utilisées dans le diagnostic industriel tout en discutant leurs caractéristiques. Parmi ces techniques figurent la méthode bayésienne, les noyaux de Parzen, l'algorithme des k plus proches voisins, le calcul direct des frontières, le discriminateur linéaire de Fisher, les réseaux de neurones ainsi que les arbres de décision. Il ressort également que le choix de la méthode de reconnaissance dépend de la nature du système étudié, de la qualité des données utilisées et de la performance du classifieur.

À travers cette étude, nous pouvons conclure que les techniques d'intelligence artificielle représentent des outils particulièrement adaptés aux systèmes dynamiques et complexes, grâce à leur capacité à traiter des données volumineuses, imprécises et incertaines.

Dans le chapitre suivant, nous aborderons les algorithmes de colonies de fourmis en tant que techniques d'intelligence collective inspirées du comportement réel des fourmis, tout en expliquant leur utilisation dans les différentes étapes du diagnostic, notamment dans l'optimisation et la classification.

Chapitre3 Les colonies de fourmis

3.1. Introduction

Avec l'évolution des systèmes modernes et la complexité croissante de leurs structures, il est devenu nécessaire de rechercher des méthodes nouvelles et efficaces pour le diagnostic et l'analyse. Les approches classiques se révèlent souvent limitées face au volume important de données ou à la nature dynamique et complexe de ces environnements. C'est pourquoi les chercheurs se sont tournés vers des solutions inspirées de phénomènes naturels, capables de traiter des problèmes difficiles de manière collective et efficiente.

Parmi ces solutions, l'algorithme des colonies de fourmis (Ant Colony Optimization – ACO) s'est imposé comme un outil performant dans divers domaines tels que le diagnostic, la classification et l'optimisation. Sa particularité réside dans le principe de coopération entre entités simples pour produire des solutions optimales et adaptatives, ce qui le rend particulièrement pertinent pour les systèmes difficiles à modéliser par des méthodes traditionnelles.

Ce chapitre présente les bases théoriques de l'ACO, expose ses concepts essentiels et montre comment il peut être appliqué au diagnostic des systèmes, en mettant en évidence ses apports et ses limites à travers une comparaison avec d'autres méthodes de diagnostic.

3.2. Les fourmis réelles

3.2.1. Principe de base

Au départ, les fourmis se déplacent de manière aléatoire entre le nid et la source de nourriture. En chemin, elles déposent une substance chimique appelée **phéromone**, qui sert de signal pour faciliter le retour. Progressivement, lorsque plusieurs chemins existent, l'accumulation de phéromones sur l'un d'eux le rend plus attractif pour les autres fourmis. Ainsi, la colonie tend à renforcer collectivement le chemin le plus court et le plus efficace.

Il est important de noter que les phéromones ne transmettent pas uniquement une information de direction, mais peuvent également indiquer la **qualité de la source de nourriture**. Ce mécanisme permet à la colonie de s'adapter et de choisir la solution optimale de manière distribuée. Ce comportement illustre comment des interactions locales simples entre individus

peuvent générer une **intelligence collective** capable de résoudre des problèmes complexes, tels que la recherche du chemin le plus court. [6]

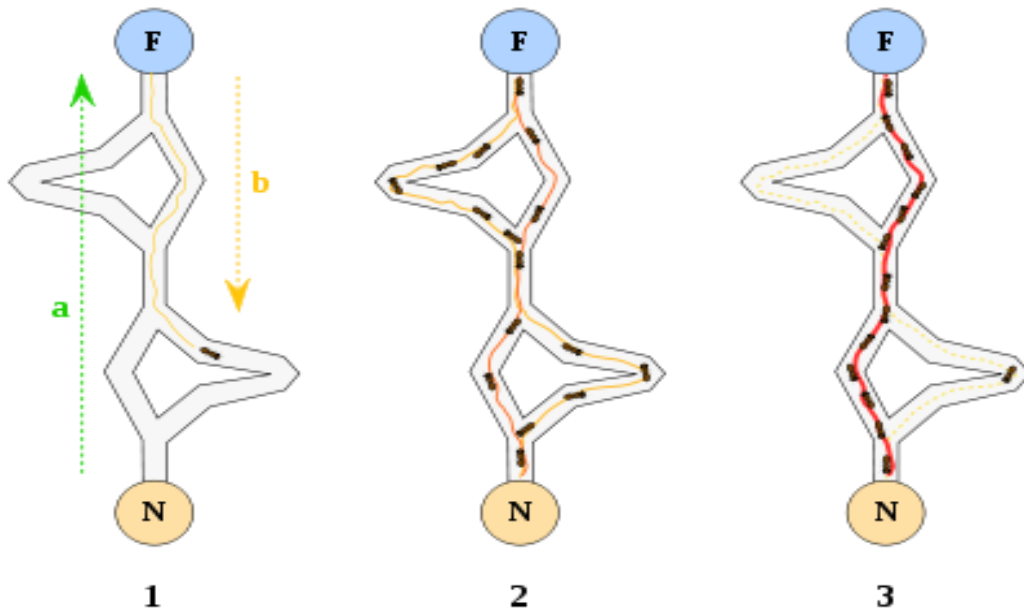


Figure 3.1: Principe de recherche du plus court chemin dans l'algorithme ACO.

La figure illustre le principe de recherche du plus court chemin par une colonie de fourmis. Dans une première étape, la première fourmi trouve la source de nourriture (F) en empruntant un chemin possible, puis retourne vers le nid (N) en déposant une trace de phéromones. Dans la deuxième étape, les fourmis utilisent aléatoirement les différents chemins disponibles, mais le chemin le plus court devient progressivement plus attractif grâce à une concentration plus élevée de phéromones. Enfin, dans la dernière étape, la majorité des fourmis emprunte le plus court chemin, tandis que les traces de phéromones présentes sur les chemins les plus longs s'évaporent progressivement, ce qui renforce le chemin optimal.

3.2.2. Comportement collectif

Les fourmis sont des insectes sociaux capables de travailler collectivement afin d'accomplir plusieurs tâches telles que la recherche de nourriture, la construction du nid et la protection de la colonie. Chaque fourmi possède des capacités simples, mais l'interaction entre les individus permet à la colonie de résoudre des problèmes complexes de manière efficace [8].

Le comportement collectif des fourmis repose principalement sur une communication indirecte basée sur les phéromones. Lorsqu'une fourmi découvre une source de nourriture, elle dépose une trace chimique sur son chemin afin de guider les autres fourmis. Plus ce chemin est

utilisé, plus la quantité de phéromones augmente, ce qui permet à la colonie de sélectionner progressivement le chemin le plus court et le plus efficace [8].

Ce phénomène d'auto-organisation a inspiré plusieurs chercheurs dans le domaine de l'intelligence artificielle, notamment pour le développement des algorithmes de colonies de fourmis (ACO), utilisés dans les problèmes d'optimisation et de diagnostic des systèmes complexes [8].

3.2.3. Stigmergie et phéromones

- **Définition de la stigmergie**

La stigmergie est définie comme un mécanisme de coordination indirecte entre les membres de la colonie, dans lequel l'information est échangée par la modification de l'environnement au lieu d'un contact direct entre les individus, ce qui permet l'émergence d'un comportement collectif organisé sans contrôle centralisé. Ce mécanisme se manifeste sous deux formes : la stigmergie quantitative, basée sur l'accumulation des traces, et la stigmergie qualitative, fondée sur la nature de la trace laissée [23].

- **Définition des phéromones**

Les phéromones sont des substances chimiques sécrétées par les fourmis lors de leurs déplacements, et elles constituent le support fondamental du mécanisme de la stigmergie. Elles agissent comme des signaux distribués dans l'environnement qui aident les autres membres de la colonie à déterminer les chemins et les directions appropriés pour atteindre la nourriture [6].

Le principe de la stigmergie est considéré comme l'un des mécanismes les plus importants de communication indirecte chez les fourmis, où la coordination entre les membres de la colonie se fait par l'interaction avec l'environnement plutôt que par un contact direct entre individus [23].

Ce comportement repose sur la substance des phéromones que les fourmis sécrètent lors de leurs déplacements sur différents chemins, et cette substance aide les autres membres de la colonie à déterminer les directions et les chemins appropriés pour atteindre la nourriture [6].

Avec la répétition du passage des fourmis sur un chemin donné, la concentration de phéromones augmente, ce qui le rend plus attractif par rapport aux autres chemins. En revanche, les phéromones s'évaporent progressivement avec le temps, ce qui permet de réduire l'importance des chemins moins efficaces et de renforcer les chemins plus courts et plus optimaux [6].

Ce comportement contribue à l'émergence d'une forme d'auto-organisation et d'intelligence collective au sein de la colonie, où un groupe d'individus aux capacités limitées parvient à trouver des solutions efficaces à des problèmes complexes de manière collective et décentralisée [23][6].

3.3. De la fourmi réelle à la fourmi artificielle

3.3.1. Principe de base

En se basant sur le comportement social des fourmis réelles présenté dans la section précédente, une variété d'algorithmes d'optimisation inspirés de la nature, appelés Optimisation par Colonie de Fourmis (OCF), a été développée. Ces algorithmes sont simples et efficaces et permettent de résoudre des problèmes complexes dans différents domaines tels que l'optimisation et le diagnostic [24].

Le principe de base repose sur l'observation du comportement collectif des fourmis réelles, notamment leur capacité à coopérer et à exploiter des informations indirectes pour trouver des solutions efficaces.

3.3.2. La fourmi artificielle

Dans ce cadre, la transition vers la fourmi artificielle consiste à modéliser chaque fourmi comme un agent simple doté d'un ensemble de règles qui définissent son comportement, de manière similaire à celui de la fourmi réelle lors de la recherche de nourriture.

La fourmi artificielle évolue dans un espace de recherche combinatoire représentant la modélisation du système étudié, où chaque déplacement effectué correspond à une solution potentielle du problème.

Les solutions générées sont ensuite évaluées à l'aide d'une fonction objectif afin de déterminer la meilleure solution possible, ce qui permet de sélectionner la solution la plus efficace et la plus adaptée au système étudié [24].

3.4. Algorithme de colonie de fourmis (ACO)

3.4.1. Principe général

L'algorithme de colonie de fourmis (Ant Colony Optimization : ACO) est l'un des algorithmes d'intelligence artificielle inspirés du comportement collectif des fourmis réelles lors de la recherche de nourriture [6].

Le principe général de cet algorithme repose sur la simulation de la manière dont les fourmis choisissent le chemin le plus court entre le nid et la source de nourriture. Lorsqu'une fourmi se déplace, elle dépose une substance chimique appelée phéromone sur son trajet. Les autres fourmis ont tendance à suivre les chemins contenant une quantité plus importante de cette substance. Avec le temps, le chemin le plus court devient plus attractif puisqu'il est davantage utilisé, ce qui augmente progressivement la quantité de phéromones déposée sur ce trajet [6].

Dans l'algorithme ACO, chaque fourmi artificielle représente une solution potentielle au problème étudié. Ces fourmis explorent différents chemins dans l'espace de recherche afin de trouver la meilleure solution possible. Après chaque itération, la quantité de phéromones est mise à jour selon la qualité des solutions obtenues. Les meilleures solutions reçoivent une quantité plus importante de phéromones, ce qui augmente leur probabilité d'être sélectionnées lors des itérations suivantes [8]

L'algorithme repose principalement sur deux mécanismes : l'exploration (Exploration) et l'exploitation (Exploitation). L'exploration consiste à rechercher de nouvelles solutions variées, tandis que l'exploitation consiste à privilégier les meilleures solutions déjà découvertes. L'équilibre entre ces deux mécanismes permet d'améliorer les performances de l'algorithme et d'atteindre progressivement des solutions efficaces proches de la solution optimale [8].

Grâce à sa capacité à résoudre des problèmes complexes de manière intelligente et flexible, l'algorithme ACO est utilisé dans plusieurs domaines tels que l'optimisation des chemins, la planification, la classification et le diagnostic des systèmes complexes [6].

3.4.2. Paramètres de l'algorithme

L'efficacité de l'algorithme de colonie de fourmis repose sur un réglage précis de plusieurs paramètres qui contrôlent le comportement des fourmis artificielles et la qualité des solutions générées. Voici une description de chaque paramètre accompagné de son équation [6].

- **Quantité de phéromones (τ_{ij})**

Représente la quantité de phéromones accumulée sur l'arête reliant les nœuds i et j . Plus sa valeur est élevée, plus cette arête a de chances d'être sélectionnée [6].

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}$$

- **Coefficient des phéromones (α)**

Contrôle l'importance accordée aux phéromones lors du choix du chemin. Une valeur élevée de α renforce le suivi des traces de phéromones [6].

$$p_{ij} = [\tau_{ij}]^a \cdot [\eta_{ij}]^\beta / \sum [\tau_{iu}]^a \cdot [\eta_{iu}]^\beta$$

- **Coefficient heuristique (β)**

Contrôle l'importance de l'information heuristique η_{ij} (comme la distance ou la similarité) par rapport aux phéromones. Une valeur élevée oriente les fourmis vers les chemins localement courts [6].

$$\eta_{ij} = 1 / d_{ij}$$

- **Taux d'évaporation (ρ)**

Détermine la vitesse de diminution des phéromones dans le temps. Une valeur élevée favorise l'exploration, une valeur faible favorise l'exploitation des meilleurs chemins trouvés [6][25].

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \rho \in (0, 1]$$

- **Nombre de fourmis (m) et itérations (T)**

Le nombre de fourmis m influence la diversité des solutions explorées, tandis que le nombre d'itérations T détermine le temps de convergence. Augmenter ces deux paramètres améliore la qualité au détriment du temps de calcul [25].

3.4.3. Variantes principales de l'ACO (AS, ACS, MMAS)

Plusieurs variantes de l'algorithme de colonie de fourmis (ACO) ont été développées afin d'améliorer les performances de l'algorithme original et d'obtenir de meilleures solutions dans les problèmes d'optimisation [25].

La première variante importante est le système de fourmis (Ant System : AS), proposé par Marco Dorigo. Dans cette approche, toutes les fourmis participent à la mise à jour des phéromones après chaque itération. Cette méthode permet l'exploration de plusieurs solutions, mais elle peut conduire à une convergence lente dans certains problèmes complexes [25].

Une autre variante connue est le système de colonies de fourmis (Ant Colony System : ACS). Cette méthode introduit une règle de transition plus efficace ainsi qu'une mise à jour locale des

phéromones pendant le déplacement des fourmis. L'objectif principal de cette amélioration est d'augmenter la vitesse de convergence et d'obtenir des solutions de meilleure qualité [6].

Le système Min-Max (Max-Min Ant System : MMAS) représente également une amélioration importante de l'algorithme ACO. Dans cette variante, seules les meilleures fourmis sont autorisées à mettre à jour les phéromones. De plus, des limites minimales et maximales sont imposées sur les valeurs des phéromones afin d'éviter une convergence prématurée vers une solution non optimale [26].

Ces différentes variantes ont permis d'améliorer considérablement les performances des algorithmes ACO dans plusieurs domaines tels que l'optimisation combinatoire, la classification et le diagnostic des systèmes complexes [6].

3.5. Application au diagnostic des systèmes

3.5.1. Lien avec la reconnaissance de formes

Le diagnostic des systèmes industriels peut être considéré comme un problème de reconnaissance de formes. Cette discipline consiste à analyser un ensemble de données afin d'identifier automatiquement l'état ou la classe d'un système à partir des informations observées [14].

Dans les applications industrielles, les données provenant des capteurs, telles que les vibrations, la température, la pression ou le courant électrique, sont transformées en vecteurs de caractéristiques représentant l'état du système [27]. Ces caractéristiques permettent ensuite de distinguer les états normaux des états défectueux grâce à des méthodes de classification.

Le processus de reconnaissance de formes comprend généralement plusieurs étapes, l'acquisition des données, l'extraction des caractéristiques, la sélection des paramètres pertinents puis la classification [14]. Cette approche est largement utilisée dans le diagnostic industriel car elle facilite la détection automatique des défauts et améliore la surveillance des systèmes complexes.

Dans ce contexte, les techniques d'intelligence artificielle, notamment les algorithmes de colonies de fourmis (ACO), peuvent être utilisées pour optimiser la sélection des caractéristiques et améliorer les performances des méthodes de classification [6].

3.5.2. Apport de l'ACO au diagnostic des systèmes

Les systèmes industriels modernes sont caractérisés par leur complexité et par la grande quantité de données générées pendant leur fonctionnement. Dans ce contexte, les méthodes classiques de diagnostic peuvent devenir limitées lorsqu'il s'agit d'identifier rapidement les défauts ou de sélectionner les paramètres les plus pertinents [28].

Les algorithmes de colonies de fourmis (ACO) offrent une approche d'optimisation inspirée du comportement collectif des fourmis réelles. Ces algorithmes permettent d'explorer efficacement l'espace de recherche afin de trouver des solutions optimales ou proches de l'optimum [6].

Dans le domaine du diagnostic, l'ACO est principalement utilisé pour la sélection de caractéristiques, l'optimisation des paramètres et l'amélioration des performances des méthodes de classification [6]. Grâce au mécanisme des phéromones, les solutions les plus performantes sont progressivement favorisées, ce qui améliore la précision du diagnostic.

Par ailleurs, les méthodes ACO possèdent une bonne capacité d'exploration et d'exploitation, leur permettant d'éviter les minima locaux et d'obtenir des résultats efficaces dans les problèmes complexes de diagnostic industriel [26].

3.6. Méthode alternative de diagnostic (Arbre de décision)

En plus des algorithmes de colonies de fourmis (ACO), plusieurs méthodes d'intelligence artificielle sont utilisées dans le domaine du diagnostic des systèmes industriels. Parmi ces méthodes, les arbres de décision représentent une approche largement utilisée grâce à leur simplicité et leur efficacité dans les problèmes de classification et de prise de décision [29].

Les arbres de décision sont utilisés dans plusieurs applications industrielles pour la détection des défauts, l'analyse des données et l'aide à la décision. Les sections suivantes présentent le principe général de cette méthode ainsi qu'un exemple d'application dans le domaine du diagnostic.

3.6.1. Présentation générale

L'arbre de décision est une méthode d'apprentissage supervisé utilisée pour analyser les données et effectuer des tâches de classification [30]. Son principe consiste à partitionner progressivement les données selon différents critères afin de construire une structure arborescente facilitant la prise de décision.

La structure d'un arbre de décision débute par une racine représentant l'ensemble des données. Ensuite, les données sont divisées en plusieurs sous-ensembles à travers des nœuds

internes basés sur certaines caractéristiques. Les feuilles de l'arbre représentent finalement les classes ou les décisions obtenues [31].

Parmi les algorithmes les plus connus dans ce domaine, on trouve l'algorithme C4.5 proposé par Quinlan, largement utilisé pour la construction des arbres de décision dans les applications de classification et de diagnostic [30].

Les arbres de décision présentent plusieurs avantages, notamment leur simplicité, leur rapidité d'exécution et leur capacité à traiter différents types de données. Toutefois, leurs performances peuvent diminuer lorsque les données deviennent très complexes ou fortement bruitées [29].

3.6.2. Exemple d'application

Les arbres de décision sont largement utilisés dans les systèmes de diagnostic industriel pour la détection et la classification des défauts à partir des données collectées par les capteurs [29].

Dans une application de diagnostic, plusieurs paramètres du système, tels que la température, les vibrations, la pression ou le courant électrique, sont utilisés comme données d'entrée. L'arbre de décision analyse ensuite ces informations à travers une série de tests afin de déterminer l'état du système, qu'il soit normal ou défaillant [30].

Par exemple, dans un système industriel, une augmentation anormale de la température accompagnée de fortes vibrations peut conduire l'arbre de décision à identifier un défaut mécanique. Chaque chemin suivi dans l'arbre correspond à une suite de décisions permettant d'aboutir à un diagnostic final [31].

Grâce à leur rapidité d'exécution et à leur simplicité d'interprétation, les arbres de décision sont largement utilisés dans les systèmes de surveillance et de maintenance industrielle. Ils contribuent à améliorer la prise de décision et à réduire les temps d'arrêt des équipements [29].

3.6.3. Comparaison avec l'ACO

Après avoir présenté le principe et les caractéristiques des arbres de décision dans les chapitres précédents, il est possible de les comparer aux algorithmes de colonies de fourmis (ACO), puisque les deux approches sont utilisées dans les applications de diagnostic et de classification [6].

Les arbres de décision reposent sur une structure arborescente basée sur un ensemble de règles et de tests permettant d'obtenir rapidement une décision finale. Cette méthode se distingue par sa simplicité, sa rapidité d'exécution ainsi que par la facilité d'interprétation de ses résultats [30].

En revanche, les algorithmes de colonies de fourmis (ACO) reposent sur un mécanisme d'optimisation inspiré du comportement collectif des fourmis réelles, où les phéromones sont utilisées pour guider la recherche vers les meilleures solutions possibles [6]. Ces algorithmes possèdent une meilleure capacité d'exploration de l'espace de recherche et permettent de traiter des problèmes complexes comportant un grand nombre de paramètres et de données.

Bien que les arbres de décision nécessitent généralement un temps d'exécution plus faible que les algorithmes ACO, leurs performances peuvent diminuer lorsque les données deviennent très complexes ou fortement bruitées. À l'inverse, les algorithmes ACO demandent un temps de calcul plus important, mais ils offrent souvent des solutions plus optimisées et une meilleure précision dans les applications de diagnostic industriel [31].

Ainsi, le choix entre les arbres de décision et les algorithmes ACO dépend principalement de la nature du système étudié, de la complexité des données et du niveau de précision recherché dans le processus de diagnostic.

Le tableau suivant résume les principales différences entre les deux approches :

Tableau 3.1: Comparaison des caractéristiques de l'ACO et de l'arbre de décision

Critère	ACO	Arbre de décision
Principe	Optimisation collective par phéromones	Règles de décision hiérarchiques
Interprétabilité	Complexe	Simple et visuelle
Temps d'exécution	Plus élevé	Rapide
Données complexes et bruitées	Bien adapté	Performances dégradées
Capacité d'exploration	Élevée	Limitée
Systèmes dynamiques	Très adapté	Réapprentissage requis

Critère	ACO	Arbre de décision
Précision du diagnostic	Élevée	Variable

Le tableau précédent montre que les arbres de décision sont adaptés aux problèmes simples nécessitant une interprétation rapide, tandis que les algorithmes ACO sont plus efficaces pour les systèmes complexes nécessitant une optimisation avancée.

3.7. Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux des colonies de fourmis ainsi que les principes de fonctionnement des algorithmes de colonies de fourmis (ACO). Nous avons d'abord étudié le comportement collectif des fourmis réelles et le rôle des phéromones dans la recherche du chemin optimal.

Ensuite, nous avons présenté le passage de la fourmi réelle à la fourmi artificielle ainsi que le principe général de l'algorithme ACO et ses principaux paramètres.

Nous avons également présenté les variantes les plus connues de l'ACO, notamment AS, ACS et MMAS, qui ont permis d'améliorer les performances de cette méthode dans les problèmes d'optimisation. Par ailleurs, nous avons montré l'importance de l'ACO dans le domaine du diagnostic des systèmes et son lien avec la reconnaissance de formes.

Enfin, une comparaison avec les arbres de décision a été réalisée afin de mettre en évidence les avantages et les limites de chaque approche dans les applications de diagnostic industriel. Cette étude montre que les algorithmes ACO représentent une méthode efficace et adaptée aux systèmes complexes grâce à leur capacité d'exploration et d'optimisation.

Ainsi, les notions présentées dans ce chapitre constituent une base théorique importante pour la suite de ce travail et pour l'application des méthodes intelligentes dans le diagnostic des systèmes.

Chapitre4 Application et réalisation

4.1. Introduction

Dans ce chapitre, nous présentons la partie pratique de notre travail consacrée à l'implémentation de l'algorithme de colonies de fourmis (ACO) dans le domaine du diagnostic des systèmes. Après avoir étudié les concepts théoriques liés à cette méthode dans les chapitres précédents, nous nous intéressons ici à la réalisation d'une application permettant de simuler le comportement de l'algorithme et d'analyser ses performances.

Cette application a été développée à l'aide du langage Python ainsi que de plusieurs bibliothèques dédiées au calcul scientifique, à la visualisation et à la conception d'interfaces graphiques. L'objectif principal est de mettre en œuvre l'algorithme ACO afin de détecter et localiser les défauts dans différents systèmes étudiés.

Dans ce chapitre, nous présentons d'abord l'environnement de travail et les outils utilisés pour le développement de l'application. Ensuite, nous décrivons les systèmes étudiés ainsi que les différentes étapes de l'implémentation de l'algorithme. Enfin, nous exposons les résultats expérimentaux obtenus avant de terminer par une discussion des performances et des limites de la méthode proposée.

4.2. Environnement de travail

Dans le cadre de ce travail, une application de diagnostic basée sur l'algorithme de colonies de fourmis (ACO) a été développée afin de simuler le comportement intelligent des fourmis artificielles dans la recherche et la détection des défauts. Cette application représente la partie pratique de notre étude et permet d'évaluer l'efficacité de l'algorithme dans différents systèmes étudiés.

Le développement de l'application a été réalisé en utilisant le langage de programmation Python, connu pour sa simplicité, sa flexibilité et sa large utilisation dans les domaines de l'intelligence artificielle, de l'optimisation et de l'analyse des données. Le choix de ce langage a également été motivé par la disponibilité d'un grand nombre de bibliothèques facilitant l'implémentation des algorithmes intelligents et la création d'interfaces graphiques interactives.

L'environnement de travail adopté permet d'assurer plusieurs fonctionnalités importantes, notamment l'exécution de l'algorithme ACO, le traitement des données, la simulation des systèmes étudiés ainsi que la visualisation des résultats obtenus. Plusieurs outils et bibliothèques ont été intégrés afin de garantir le bon fonctionnement de l'application et d'améliorer l'interaction avec l'utilisateur.

L'application développée offre également une représentation graphique des topologies étudiées ainsi qu'un suivi de l'évolution de l'algorithme à travers l'affichage des chemins empruntés et de la courbe de convergence. Cette approche permet de mieux analyser le comportement de l'algorithme et d'évaluer ses performances dans le domaine du diagnostic des systèmes.

Enfin, l'environnement de développement utilisé facilite les tests expérimentaux et permet d'étudier l'influence des différents paramètres de l'algorithme sur la qualité des résultats obtenus.

4.2.1. Langage et outils utilisés

- **Langage de programmation : Python**

Le développement de l'application a été réalisé à l'aide de Python.

Ce langage a été choisi grâce à sa simplicité, sa

flexibilité et la disponibilité de nombreuses bibliothèques

adaptées aux domaines de l'intelligence artificielle, de l'optimisation et de la visualisation des données.

Python facilite l'implémentation des algorithmes intelligents tels que l'algorithme de colonies de fourmis (ACO) ainsi que la création d'interfaces graphiques interactives. Il permet également de réaliser des simulations, d'analyser les résultats obtenus et de représenter graphiquement les différentes topologies étudiées dans l'application.

Grâce à ses fonctionnalités et à sa facilité d'utilisation, Python représente un environnement adapté au développement des applications de diagnostic intelligent.

- **Bibliothèques utilisées**

Afin d'assurer le bon fonctionnement de l'application développée, plusieurs bibliothèques Python ont été utilisées durant l'implémentation de l'algorithme de colonies de fourmis (ACO).



Figure 4.1: Logo de python.

Ces bibliothèques jouent un rôle important dans le traitement des données, la représentation graphique ainsi que la création de l'interface utilisateur.

Les principales bibliothèques utilisées sont :

- **NumPy**: utilisée pour effectuer les opérations mathématiques et manipuler les tableaux de données nécessaires à l'exécution de l'algorithme.
- **NetworkX** : utilisée pour représenter les graphes et les différentes topologies étudiées dans l'application.
- **Matplotlib** : utilisée pour tracer les courbes et visualiser l'évolution de l'algorithme ainsi que les résultats expérimentaux.
- **Tkinter** : utilisée pour développer l'interface graphique de l'application et assurer l'interaction avec l'utilisateur.

4.2.2. Architecture générale de l'application

L'application développée repose sur une architecture permettant d'assurer le fonctionnement de l'algorithme de colonies de fourmis (ACO) dans le processus de diagnostic des systèmes étudiés. Cette architecture regroupe plusieurs composants interconnectés assurant le traitement des données, l'exécution de l'algorithme ainsi que l'affichage des résultats obtenus.

Le fonctionnement de l'application débute par l'introduction des données relatives au système étudié à travers l'interface graphique. Ces données sont ensuite transmises au module principal chargé de l'exécution de l'algorithme ACO. Ce dernier analyse les différentes possibilités de parcours afin de rechercher les chemins optimaux et détecter les éventuels défauts présents dans le système.

Après l'exécution de l'algorithme, les résultats sont envoyés vers les modules de visualisation permettant d'afficher les graphes, les chemins sélectionnés par les fourmis artificielles ainsi que les différentes statistiques liées au processus de diagnostic.

L'architecture générale de l'application permet également d'assurer l'interaction entre les différents composants du système, notamment l'interface utilisateur, le module de calcul et le module d'affichage graphique.

Cette organisation facilite la compréhension du fonctionnement global de l'application et améliore la gestion des différentes opérations réalisées durant les simulations.

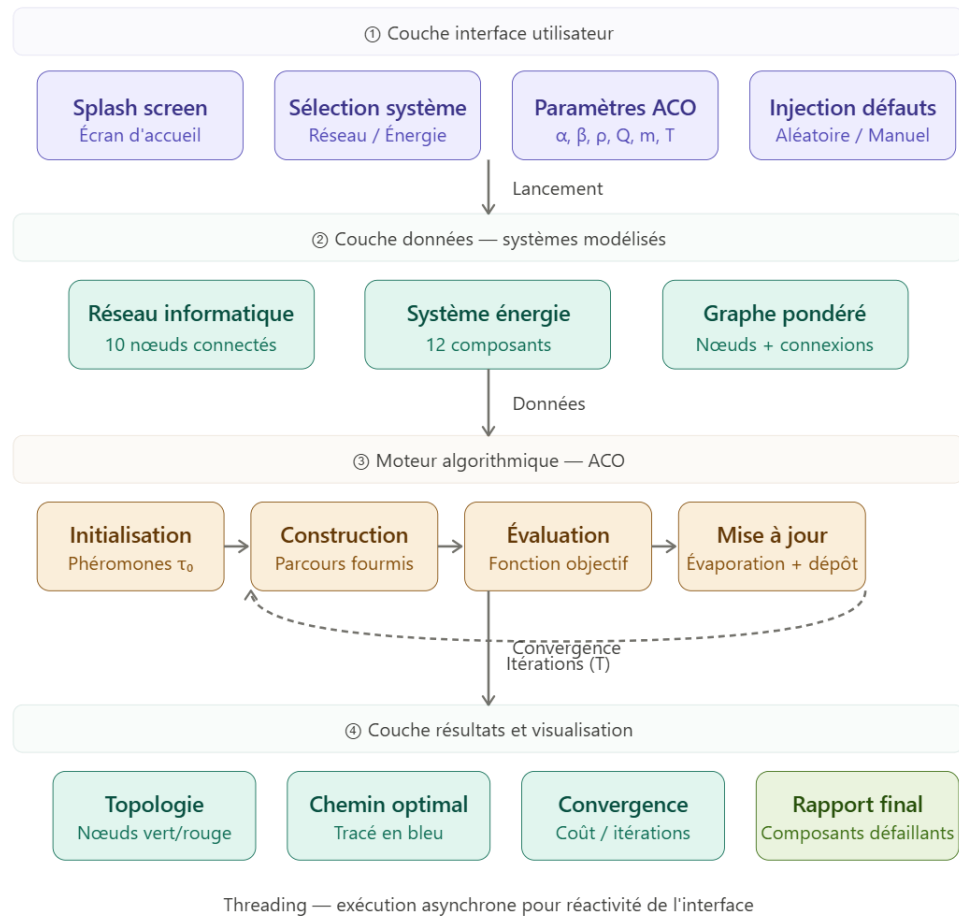


Figure 4.2: l'architecture générale de l'application.

- **L'écran d'accueil**

Cette écran représente la page principale de l'application développée en langage Python, qui apparaît lors du premier lancement du système. Elle contient le titre du projet « ACO *FaultDiagnosis* » ainsi qu'une indication de son objectif, à savoir le diagnostic des pannes dans les réseaux en utilisant l'algorithme de la colonie de fourmis (Ant ColonyOptimization – ACO).

L'interface inclut également un message de bienvenue destiné à l'utilisateur ainsi que les noms des étudiantes ayant participé à la réalisation du projet. Un design simple a été adopté, avec des couleurs sombres et des éléments graphiques représentant des fourmis, afin de mettre en évidence la relation entre l'application et l'algorithme de la colonie de fourmis utilisé pour le diagnostic.

Enfin, l'interface comporte un bouton « Entrée » permettant d'accéder à la page principale du système et de commencer à utiliser les différentes fonctionnalités de l'application.

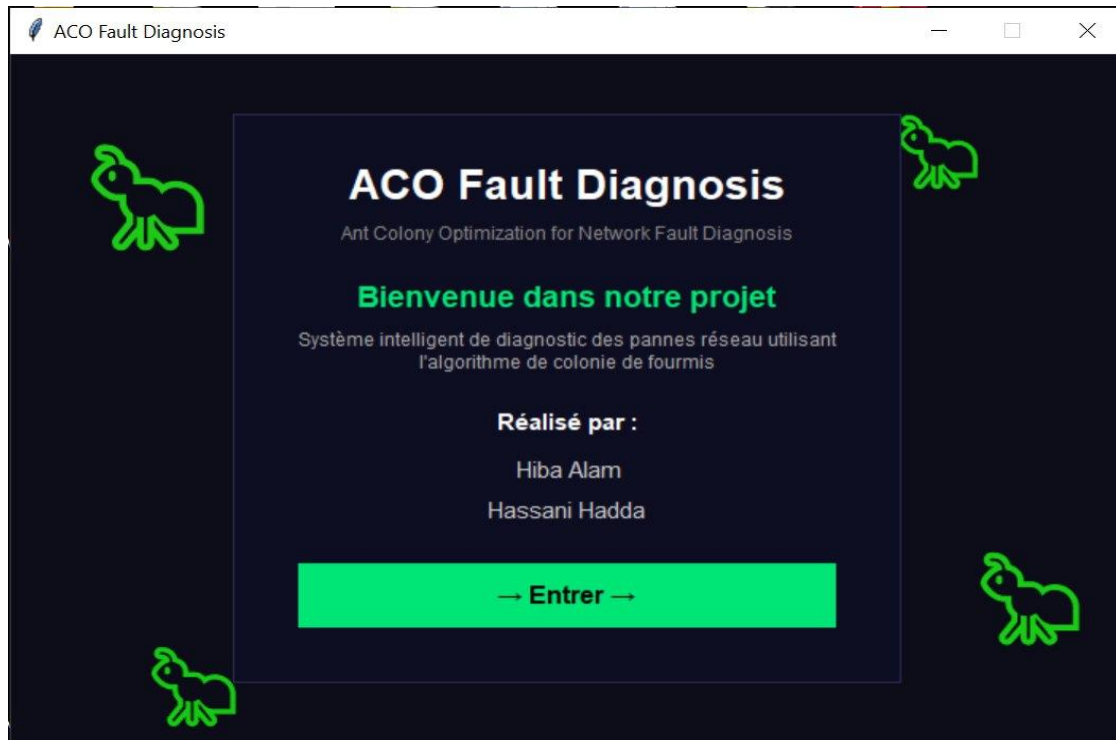


Figure 4.3: Écran d'accueil (Splash Screen) du projet avec verrouillage d'accès

- **Description de l'interface principale**

L'interface principale du système de diagnostic des pannes basé sur l'algorithme

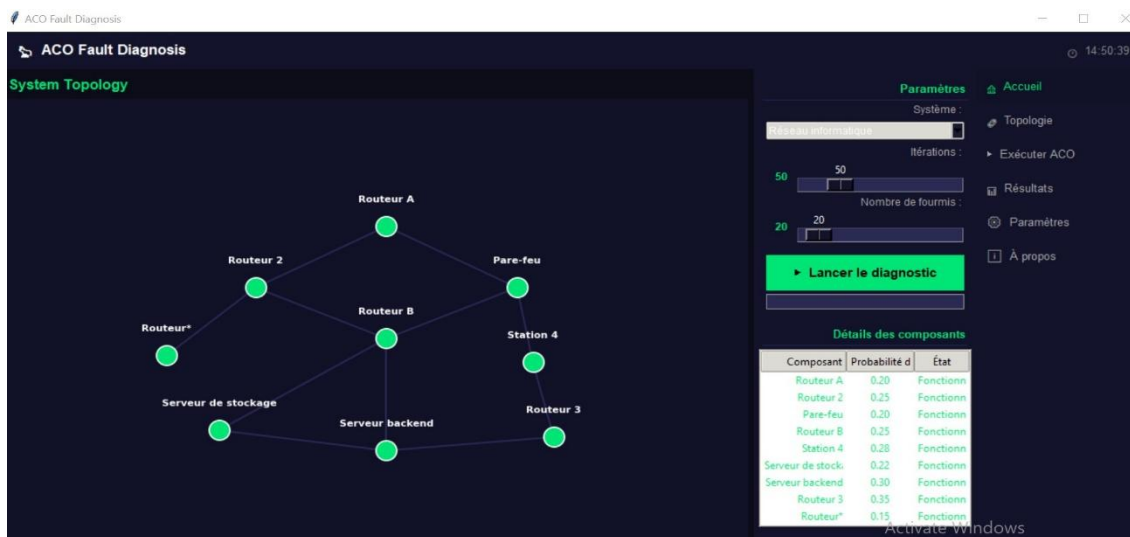


Figure 4.4: l'interface principale

d'optimisation par colonie de fourmis (ACO) est organisée en plusieurs sections :

Cette figure représente l'interface principale de l'application développée pour le diagnostic des pannes à l'aide de l'algorithme des colonies de fourmis (ACO).

L'interface contient :

Une représentation graphique de la topologie du système sous forme de nœuds et de connexions.

Un panneau de configuration pour régler les paramètres de l'algorithme, comme le nombre d'itérations et le nombre de fourmis ;

Un bouton permettant de lancer le processus de diagnostic ;

Un tableau affichant les résultats du diagnostic, notamment l'état des composants et la probabilité de panne ;

Des statistiques liées à l'exécution, telles que le temps d'exécution et le taux de réussite.

4.3. Présentation des systèmes étudiés

Afin d'évaluer les performances de l'algorithme de colonies de fourmis (ACO) dans le domaine du diagnostic, plusieurs systèmes ont été étudiés à travers l'application développée. Ces systèmes ont été choisis afin de représenter différents environnements dans lesquels les techniques de diagnostic intelligent peuvent être appliquées.

L'objectif principal de cette étude est d'analyser la capacité de l'algorithme ACO à détecter les défauts et à rechercher les chemins optimaux dans différentes structures. Chaque système étudié possède des caractéristiques spécifiques permettant d'observer le comportement des fourmis artificielles dans plusieurs situations.

Les systèmes utilisés dans cette application sont représentés par un réseau informatique, un système d'énergie industrielle. Pour chacun de ces systèmes, l'application permet de simuler le fonctionnement de l'algorithme, d'analyser les résultats obtenus et d'évaluer l'efficacité de la méthode proposée dans le processus de diagnostic.

Les sections suivantes présentent en détail les différents systèmes étudiés ainsi que leurs principales caractéristiques.

4.3.1. Réseau informatique

Le réseau informatique représente l'un des systèmes étudiés dans cette application afin de tester les performances de l'algorithme de colonies de fourmis (ACO) dans le domaine du diagnostic. Ce système est constitué d'un ensemble de nœuds reliés entre eux par différentes connexions permettant l'échange des données et des informations.

Dans cette étude, le réseau est représenté sous forme d'un graphe composé de plusieurs sommets et arêtes. Chaque nœud représente un équipement du réseau, tandis que les connexions représentent les chemins possibles empruntés pour la transmission des données.

L'objectif principal de la simulation est de permettre à l'algorithme ACO de rechercher le chemin optimal entre les différents nœuds tout en détectant les éventuelles anomalies ou défaillances présentes dans le réseau. Les fourmis artificielles explorent les différents chemins disponibles et utilisent les phéromones afin de privilégier progressivement les trajets les plus efficaces.

Cette approche permet d'améliorer la gestion du réseau et de faciliter la détection des défauts pouvant affecter les performances du système.

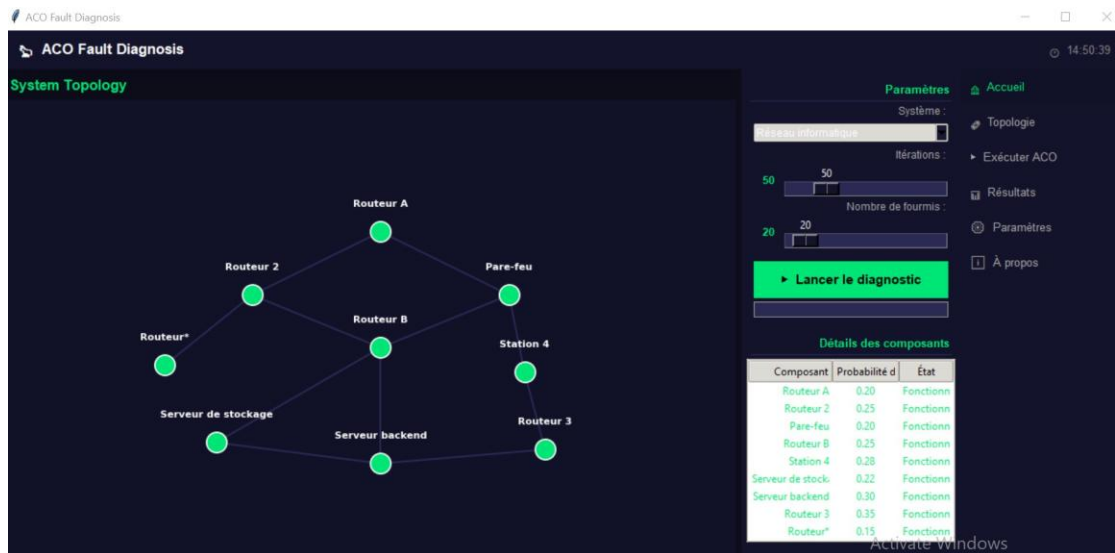


Figure 4.5: Topologie du réseau informatique.

4.3.2. Système d'énergie industrielle

Le système d'énergie industrielle constitue un autre environnement étudié dans cette application afin d'évaluer l'efficacité de l'algorithme de colonies de fourmis (ACO) dans le domaine du diagnostic. Ce système représente un ensemble d'équipements énergétiques interconnectés assurant la distribution et le transfert de l'énergie entre plusieurs composants.

Dans cette étude, le système est modélisé sous forme d'un graphe composé de plusieurs nœuds et connexions représentant les différentes unités énergétiques et les liens existants entre elles. Cette représentation permet de simuler le fonctionnement du système ainsi que les échanges réalisés entre les différents composants.

L'objectif principal de cette simulation est de permettre à l'algorithme ACO d'identifier les chemins optimaux de circulation et de détecter les éventuels défauts pouvant perturber le bon fonctionnement du système énergétique. Les fourmis artificielles explorent les différentes connexions disponibles tout en utilisant les phéromones afin de renforcer progressivement les solutions les plus efficaces.

Cette approche permet d'améliorer l'analyse du système énergétique et de faciliter la localisation des anomalies susceptibles d'influencer les performances globales du réseau industriel.

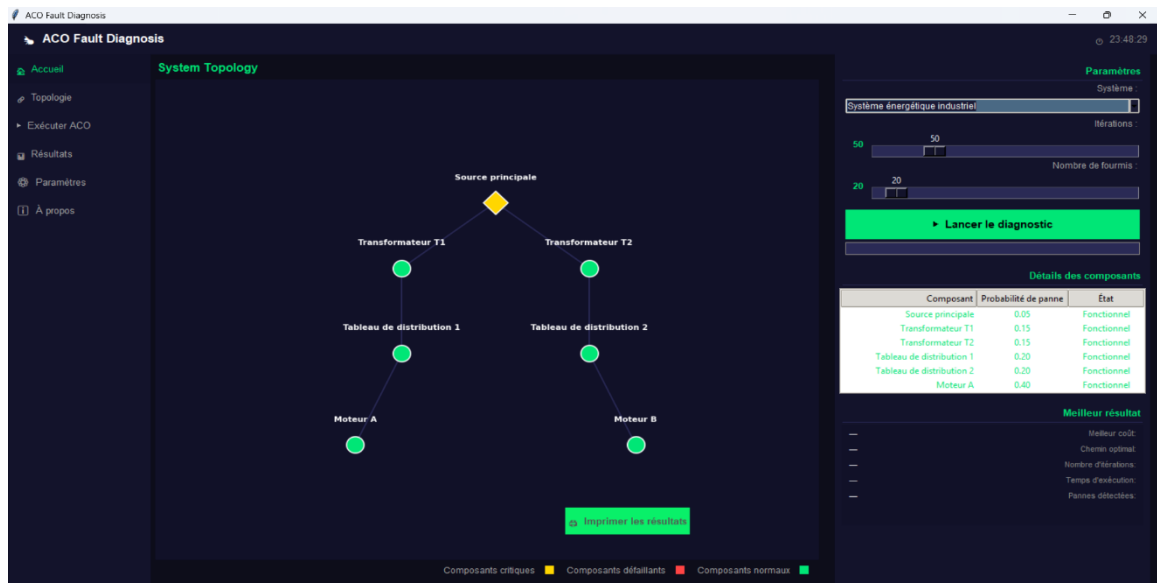


Figure 4.6: Topologie du système d'énergie industrielle.

4.4. Implémentation de l'algorithme ACO

L'implémentation de l'algorithme de colonies de fourmis (ACO) constitue l'étape principale de la réalisation de cette application. Cette phase consiste à traduire les principes théoriques de l'algorithme en un programme capable de simuler le comportement des fourmis artificielles dans le processus de recherche et de diagnostic.

Le développement de l'algorithme repose sur plusieurs étapes importantes, notamment la représentation des systèmes étudiés sous forme de graphes, l'initialisation des paramètres de l'algorithme ainsi que la gestion du déplacement des fourmis artificielles entre les différents nœuds.

Durant l'exécution, chaque fourmi explore les différents chemins disponibles selon les informations heuristiques et les traces de phéromones déposées sur les connexions. Après

chaque itération, les phéromones sont mises à jour afin de renforcer les chemins les plus efficaces et de guider progressivement la recherche vers les meilleures solutions.

Cette implémentation permet de simuler le fonctionnement de l'algorithme ACO dans différents environnements étudiés et d'évaluer son efficacité dans la détection des défauts et l'optimisation des chemins.

4.4.1. Structure générale de l'algorithme

La structure du code de l'application a été organisée de manière à faciliter l'implémentation de l'algorithme de colonies de fourmis (ACO) ainsi que la gestion des différentes fonctionnalités de l'application. Le programme est composé de plusieurs modules assurant chacun une tâche spécifique dans le processus de simulation et de diagnostic.

Le module principal est responsable de l'exécution de l'algorithme ACO et de la gestion des différentes étapes de calcul. Il permet notamment d'initialiser les paramètres de l'algorithme, de générer les fourmis artificielles et de contrôler leurs déplacements à travers les différents graphes étudiés.

D'autres modules sont utilisés pour la représentation des systèmes sous forme de graphes, l'affichage graphique des résultats ainsi que la gestion de l'interface utilisateur. Cette organisation permet de rendre le code plus clair, plus structuré et plus facile à maintenir.

La structure adoptée facilite également l'ajout de nouvelles fonctionnalités et l'amélioration future de l'application, notamment dans le cadre de l'optimisation des performances ou de l'intégration d'autres méthodes de diagnostic intelligentes.

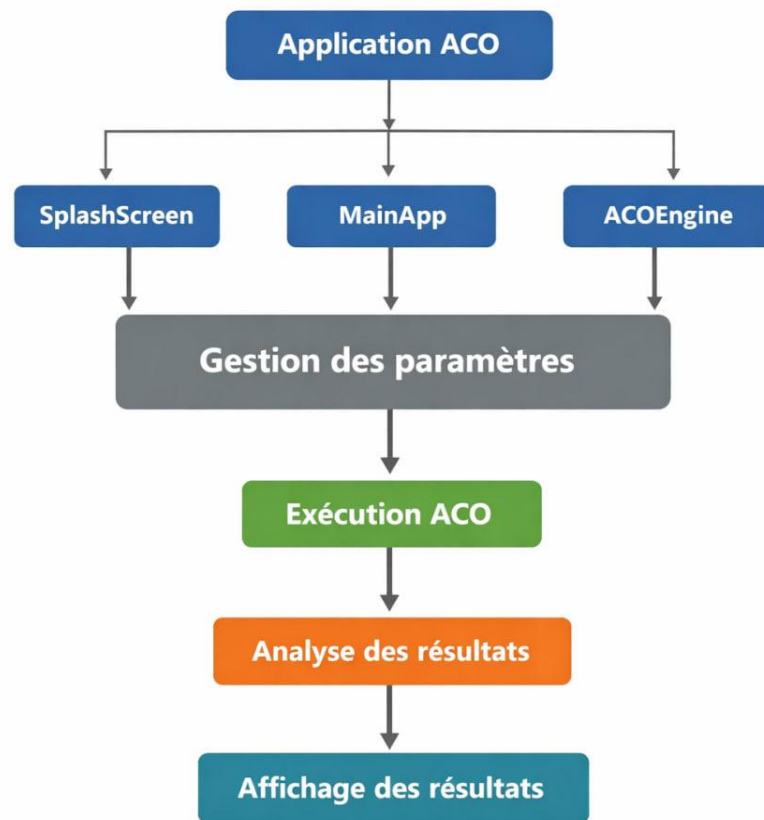


Figure 4.7: la structure générale de l'algorithme

4.4.2. Paramètres utilisés

Le fonctionnement de l'algorithme de colonies de fourmis (ACO) dépend de plusieurs paramètres influençant le comportement des fourmis artificielles durant le processus de recherche et de diagnostic. Le choix de ces paramètres joue un rôle important dans la qualité des résultats obtenus ainsi que dans la rapidité de convergence de l'algorithme.

Parmi les principaux paramètres utilisés dans cette application, le paramètre Alpha contrôle l'influence des phéromones déposées sur les différents chemins. Plus sa valeur est élevée, plus les fourmis privilégient les chemins fortement marqués par les phéromones.

Le paramètre Beta représente l'importance des informations heuristiques dans le choix des chemins. Il permet aux fourmis artificielles de prendre en considération les caractéristiques locales du graphe afin d'orienter efficacement la recherche vers les solutions les plus appropriées.

Le taux d'évaporation des phéromones constitue également un paramètre important dans l'algorithme. Ce mécanisme permet d'éviter une accumulation excessive des phéromones et favorise l'exploration de nouveaux chemins durant les différentes itérations.

Enfin, le nombre de fourmis ainsi que le nombre d'itérations influencent directement les performances de l'algorithme. Une augmentation de ces paramètres permet généralement d'obtenir de meilleures solutions, mais nécessite un temps d'exécution plus important.

Tableau 4.1: Principaux paramètres influençant le fonctionnement de l'ACO

Paramètre	Rôle
Alpha (α)	Importance des phéromones
Beta (β)	Importance de l'information heuristique
Taux d'évaporation (ρ)	Réduction progressive des phéromones
Nombre de fourmis	Nombre d'agents explorateurs
Nombre d'itérations	Nombre de cycles de l'algorithme
Phéromone initiale	Valeur de départ des traces de phéromones

4.4.3. Processus de détection des défauts

Le processus de détection des défauts dans cette application repose sur le comportement collectif des fourmis artificielles lors de l'exploration des différents chemins disponibles dans les systèmes étudiés. L'objectif principal est d'identifier les zones susceptibles de contenir des anomalies ou des perturbations pouvant affecter le bon fonctionnement du système.

Au début de l'exécution, les fourmis artificielles sont distribuées sur les différents nœuds du graphe représentant le système étudié. Chaque fourmi se déplace entre les nœuds en fonction des informations heuristiques et de la quantité de phéromones présente sur les connexions.

Durant les différentes itérations, les chemins considérés comme plus efficaces reçoivent une quantité plus importante de phéromones, ce qui augmente progressivement leur probabilité

d'être sélectionnés par les autres fourmis. En revanche, les chemins moins performants deviennent moins attractifs à cause du phénomène d'évaporation des phéromones.

Lorsqu'un comportement anormal ou un chemin présentant des caractéristiques inhabituelles est détecté, l'algorithme peut identifier la présence d'un défaut dans le système. Cette approche permet de localiser les zones critiques et d'améliorer le processus de diagnostic dans les différents environnements étudiés.

Le mécanisme adopté permet également d'assurer une meilleure exploration des solutions possibles et d'optimiser progressivement la qualité des résultats obtenus durant les simulations.

4.5. Résultats et discussion

Après l'implémentation de l'algorithme de colonies de fourmis (ACO), plusieurs tests et simulations ont été réalisés afin d'évaluer les performances de l'application dans les différents systèmes étudiés. Les résultats obtenus permettent d'analyser le comportement des fourmis artificielles durant le processus de recherche et de diagnostic.

L'application développée a permis de visualiser les chemins empruntés par les fourmis ainsi que l'évolution progressive des solutions au cours des différentes itérations. Grâce au mécanisme des phéromones, l'algorithme parvient progressivement à privilégier les chemins les plus efficaces et à identifier les zones pouvant contenir des défauts.

Les simulations réalisées sur les différents systèmes étudiés montrent que l'algorithme ACO possède une bonne capacité d'exploration et permet d'obtenir des résultats satisfaisants dans le domaine du diagnostic. Les résultats affichés par l'application facilitent également l'analyse des performances de l'algorithme ainsi que l'interprétation des solutions obtenues.

Les sections suivantes présentent les différentes visualisations obtenues, l'analyse des performances de l'algorithme ainsi que les limites observées durant les expérimentations.

4.5.1. Visualisation des résultats

L'application développée permet de visualiser les résultats obtenus après l'exécution de l'algorithme de colonies de fourmis (ACO). Cette visualisation facilite l'analyse du comportement des fourmis artificielles ainsi que l'interprétation des solutions générées durant le processus de diagnostic.

Les résultats sont représentés sous forme de graphes illustrant les différents chemins empruntés par les fourmis à travers les systèmes étudiés. Les chemins considérés comme les

plus efficaces apparaissent progressivement grâce au renforcement des traces de phéromones déposées durant les différentes itérations.

L'application permet également d'afficher certaines informations importantes telles que le chemin optimal trouvé, le nombre d'itérations réalisées ainsi que l'évolution des performances de l'algorithme au cours de l'exécution.

En complément, une courbe de convergence est utilisée afin de représenter graphiquement l'évolution de la solution optimale durant les simulations. Cette représentation facilite l'observation du comportement de l'algorithme et permet d'évaluer sa capacité à converger vers une solution efficace.

4.5.2. Analyse des performances

L'analyse des performances de l'algorithme de colonies de fourmis (ACO) permet d'évaluer son efficacité dans le processus de diagnostic des différents systèmes étudiés. Les résultats obtenus montrent que l'algorithme possède une bonne capacité d'exploration des chemins et permet d'identifier progressivement les solutions les plus efficaces.

Au cours des simulations, les fourmis artificielles ont réussi à rechercher les chemins optimaux grâce au mécanisme des phéromones et aux informations heuristiques utilisées durant l'exécution. Les résultats affichés montrent également une amélioration progressive de la qualité des solutions au fil des itérations.

L'algorithme présente aussi une capacité d'adaptation aux différentes structures étudiées, notamment les réseaux informatiques, les systèmes d'énergie industrielle ainsi que les lignes de production. Cette flexibilité permet d'utiliser l'approche ACO dans plusieurs domaines liés au diagnostic des systèmes.

Cependant, les performances de l'algorithme dépendent fortement des paramètres utilisés, tels que le nombre de fourmis, le taux d'évaporation des phéromones et le nombre d'itérations. Une mauvaise configuration de ces paramètres peut influencer la rapidité de convergence ainsi que la qualité des résultats obtenus.

4.5.3. Limites et perspectives

Malgré les résultats satisfaisants obtenus lors des simulations, certaines limites ont été observées durant l'utilisation de l'algorithme de colonies de fourmis (ACO). Les performances de l'algorithme dépendent fortement du choix des paramètres utilisés, ce qui peut influencer la rapidité de convergence ainsi que la qualité des solutions obtenues.

L'augmentation du nombre de nœuds et de connexions dans les systèmes étudiés peut également entraîner une augmentation du temps d'exécution et de la complexité des calculs. De plus, certaines situations complexes peuvent nécessiter un nombre important d'itérations afin d'obtenir des résultats plus précis.

Dans les travaux futurs, il serait possible d'améliorer l'application en intégrant d'autres techniques d'intelligence artificielle afin d'augmenter les performances du processus de diagnostic. Il serait également intéressant d'optimiser les paramètres de l'algorithme de manière automatique afin d'obtenir de meilleurs résultats dans différents environnements.

Enfin, l'application pourrait être étendue à d'autres domaines industriels nécessitant des méthodes intelligentes de détection et d'optimisation.

4.6. Conclusion

Dans ce chapitre, nous avons présenté la partie pratique de notre travail consacrée à l'implémentation de l'algorithme de colonies de fourmis (ACO) dans le domaine du diagnostic des systèmes. Nous avons décrit l'environnement de développement utilisé ainsi que les différents systèmes étudiés à travers l'application réalisée.

L'implémentation de l'algorithme ACO a permis de simuler le comportement des fourmis artificielles dans la recherche des chemins optimaux et la détection des défauts. Les résultats obtenus montrent que cette approche possède une bonne capacité d'exploration et permet d'obtenir des solutions satisfaisantes dans différents environnements étudiés.

Les différentes simulations réalisées ont également permis d'analyser les performances de l'algorithme ainsi que l'influence de certains paramètres sur la qualité des résultats obtenus. Malgré certaines limites observées, l'approche proposée reste une solution intéressante pour les applications de diagnostic intelligent.

Enfin, ce travail peut être amélioré dans le futur à travers l'intégration d'autres techniques d'intelligence artificielle et l'optimisation des paramètres de l'algorithme afin d'obtenir des performances plus élevées.

Conclusion

Ce travail avait pour objectif d'appliquer l'algorithme de colonie de fourmis (ACO) au diagnostic de systèmes dynamiques et complexes, en s'appuyant sur les principes de la reconnaissance de formes comme cadre méthodologique. À travers les quatre chapitres développés, nous avons progressivement construit les bases théoriques nécessaires avant de les concrétiser dans une application pratique.

Le premier chapitre a permis d'établir une vision claire de la notion de système dans toute sa diversité, en mettant en évidence les spécificités des systèmes dynamiques et complexes qui constituent le cœur de ce travail. La modélisation a été présentée comme un outil indispensable pour représenter ces systèmes et préparer le terrain à toute démarche de diagnostic.

Le deuxième chapitre a exposé le diagnostic par reconnaissance de formes comme une approche structurée et efficace pour identifier les états de défaillance d'un système à partir de données observées. Les différentes méthodes de classification, de la méthode bayésienne aux arbres de décision, ont été présentées et comparées, offrant ainsi une vue d'ensemble des outils disponibles dans ce domaine.

Le troisième chapitre a été consacré aux algorithmes de colonie de fourmis, en retraçant leur origine biologique, leurs mécanismes de fonctionnement et leurs principales variantes. La comparaison avec les arbres de décision a mis en évidence les avantages de l'ACO pour les systèmes complexes, notamment sa capacité d'exploration globale et son adaptabilité aux environnements dynamiques.

Enfin, le quatrième chapitre a présenté l'application développée en Python, qui intègre l'algorithme ACO pour le diagnostic de deux systèmes : un réseau informatique et un système d'énergie industrielle. Les résultats obtenus montrent que l'algorithme converge progressivement vers les solutions les plus efficaces, en identifiant les composants défaillants grâce au mécanisme des phéromones.

Ce travail ouvre plusieurs perspectives intéressantes. Il serait envisageable d'étendre l'application à d'autres types de systèmes industriels, d'intégrer des mécanismes d'optimisation automatique des paramètres de l'ACO, ou encore de combiner cette approche avec d'autres techniques d'intelligence artificielle telles que les réseaux de neurones, afin d'améliorer davantage la précision et la robustesse du diagnostic.

En définitive, ce travail confirme que les algorithmes d'intelligence collective, et l'ACO en particulier, représentent une voie prometteuse pour le diagnostic intelligent des systèmes

complexes, à une époque où la fiabilité et la réactivité des systèmes sont devenues des exigences incontournables.

Références

- [1] J.-L. Le Moigne, *La théorie du système général. Théorie de la modélisation*. Paris: Presses Universitaires de France, 1977.
Disponible sur: <https://scholar.google.com/citations?user=2dcvIhwAAAAJ&hl=ar&oi=sra>
- [2] L. von Bertalanffy, *General System Theory: Foundations, Development, Applications*. New York: George Braziller, 1968.
- [3] L. Ljung, *System Identification: Theory for the User*, 2nd ed. UpperSaddle River, NJ: Prentice Hall, 1999.
- [4] F. E. Cellier, *Continuous System Modeling*. New York: Springer, 1991.
- [5] D. Karnopp, D. Margolis, and R. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, 5th ed. Hoboken, NJ: Wiley, 2012.
- [6] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
doi: 10.7551/mitpress/1290.001.0001
.
- [7] D. Mouss, *Diagnostic et conduite des systèmes de production par approche à base de connaissances*, Thèse de Doctorat, Université de Batna, Algérie, 2006.
- [8] O. Kadri, *L'application des algorithmes de colonies de fourmis pour le diagnostic des systèmes dynamiques et complexes*, Thèse de Doctorat, Université de Batna 2, Algérie. [En ligne]
. Disponible : <https://dspace.univ-batna2.dz/handle/123456789/994>
- [9] L. Wu et al., "Fault diagnosis based on artificial intelligence techniques," *Applied Sciences*, vol. 11, no. 9, 2021.
- [10] A. Mokhtari, *Diagnostic des systèmes hybrides : développement d'une méthode associant la détection*
Thèse de Doctorat, INSA de Toulouse, France, 2007.
- [11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.

- [12] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. UpperSaddle River, NJ: Pearson Addison-Wesley, 2006.
- [13] K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [14] M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [15] Dubuisson and M. Masson, "A statistical decision rule within incomplete knowledge about classes," *Pattern Recognition*, vol. 24, no. 11, pp. 1007–1019, 1991.
- [16] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 4, pp. 1093–1131, 1962.
- [17] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 175–226, 1936.
- [18] M. Kalakech, *Sélection semi-supervisée d'attributs : Application à la classification de textures couleur*, Thèse de Doctorat, Université de Lille 1, France, 2011.
Disponible sur: <https://pepite-depot.univ-lille.fr/LIBRE/EDSPI/2011/50376-2011-Kalakech.pdf>
- [19] A. Rezki, A. Kadri, and F. Mokhati, "Diagnostic des systèmes complexes par une approche multi-agents," *Pattern Recognition Letters*, vol. 32, no. 6, pp. 707–714, 2011.
- [20] H. Azzag, P. Collet, and P. Krafft, "Apprentissage incrémental pour la classification de données complexes," *Pattern Recognition Letters*, vol. 32, no. 6, pp. 715–724, 2011.
- [21] A. Kadri et al., "Diagnostic industriel basé sur les techniques d'intelligence artificielle," 2012.
- [22] A. Mahdaoui and M. Mouss, "Réseaux de neurones et logique floue," *Pattern Recognition Letters*, vol. 32, no. 6, pp. 725–734, 2011.
- [23] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97–116, 1999. doi: 10.1162/106454699568700
- [24] O. Ouadfel and M. Batouche, "Ant Colony System for Dynamic Routing Problem," *International Journal of Computer Science and Network Security*, vol. 7, no. 5, pp. 139–145, 2007.
Disponible sur:

- https://www.researchgate.net/profile/Fatima-Zahra-Mhada/publication/326181211_Ant_Colony_System_for_Dynamic_Vehicle_Routing_Problem_with_Overtime_Colony-System-for-Dynamic-Vehicle-Routing-Problem-with-Overtime.pdf
- [25] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 6, pp. 29–41, 1996. doi: 10.1109/3477.484436
 - [26] T. Stützle and H. H. Hoos, "MAX–MIN Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.
 - [27] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Academic Press, 2009.
 - [28] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Berlin: Springer, 2009.
 - [29] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. John Wiley & Sons, 2011.
 - [30] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.
 - [31] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
 - [32] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2003.
 - [33] C. Saint-Jean, *Classification paramétrique robuste partiellement supervisée en reconnaissance des formes*, Thèse de La Rochelle, France, 2001. Disponible sur: https://theses.hal.science/tel-00145895/file/these_csj.pdf

ملخص

يُعدّ تشخيص الأنظمة مجالاً مهماً في البيئات الصناعية الحديثة للكشف عن الأعطال وتحسين موثوقية الأنظمة المعقدة. في هذه المذكرة تمت دراسة استخدام خوارزمية مستعمرات النمل (ACO) في تشخيص الأنظمة. تم تطوير تطبيق بلغة Python لمحاكاة سلوك النمل الاصطناعي والبحث عن المسارات المثلى للمساعدة في كشف الأعطال. أظهرت النتائج أن خوارزمية ACO تُعدّ طريقة فعالة في التحسين والاستكشاف رغم بعض القيود المتعلقة بالمعاملات ووقت التنفيذ.

الكلمات المفتاحية: تشخيص الأنظمة؛ ACO ؛ الذكاء الاصطناعي؛ بايثون؛ التعرف على الأشكال.

Résumé

Le diagnostic des systèmes joue un rôle important dans les environnements industriels modernes pour détecter les défauts et améliorer la fiabilité des systèmes complexes. Dans ce mémoire, nous avons étudié l'utilisation de l'algorithme des colonies de fourmis (ACO) appliqué au diagnostic des systèmes.

Une application en Python a été développée afin de simuler le comportement des fourmis artificielles et rechercher des chemins optimaux permettant d'aider à la détection des défauts. Les résultats obtenus montrent que l'ACO représente une méthode efficace d'optimisation et d'exploration malgré certaines limites liées aux paramètres et au temps d'exécution.

Mots-clés : Diagnostic des systèmes , ACO ,intelligence artificielle , Python, Reconnaissance de forme.

Abstract

System diagnosis is an important field in modern industrial environments for fault detection and improving the reliability of complex systems. In this thesis, we studied the use of the Ant Colony Optimization (ACO) algorithm for system diagnosis. A Python application was developed to simulate artificial ants and search for optimal paths that support fault detection. The obtained results show that ACO is an effective optimization and exploration method despite some limitations related to parameters and execution time.

Keywords: System diagnosis, ACO, artificial intelligence, Python, Pattern recognition