

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



UNIVERSITE DE BATNA
Faculté de Technologie
Département Génie Industriel



Thèse

Présentée au Laboratoire d'Automatique et Productique

par

Ouahab KADRI

(Magistère en Informatique option : Informatique Industrielle)

Pour obtenir le grade de

Docteur en Sciences

(Spécialité : Génie Industriel)

Thème

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

Soutenue publiquement le 09 juin 2013

Devant le jury composé de :

Président	NASRI	Farid	Prof	Université de Batna
Examineur	BENBOUZID	Med. El Hachemi	Prof	Université de Brest
Examineur	BATOUCHE	Med. Chawki	Prof	Université de Constantine
Examineur	BENMOHAMMED	Mohamed	Prof	Université de Constantine
Rapporteur	MOUSS	Leila Hayet	Prof	Université de Batna
Co-Rapporteur	MOUSS	Med. Djamel	MCA	Université de Batna

Remerciements

En premier lieu, je tiens à remercier ma directrice de thèse, L. Hayet Mouss, Professeur à l'Université de Batna et Directrice du LAP (Laboratoire d'Automatique et Productique), pour la confiance qu'elle m'a accordée en acceptant d'encadrer ce travail, pour ses multiples conseils et son attention permanente sur l'évolution de mes travaux. Merci aussi pour toutes les relectures, suggestions et commentaires, qui m'ont permis d'améliorer la qualité de cette thèse. Enfin, j'ai été extrêmement sensible à ses qualités humaines d'écoute et d'encouragement qui ont joué un rôle important. J'adresse mes sincères remerciements à Med.Djamel Mouss, Maître de Conférences à l'Université de Batna et Chef du Département Génie Industriel, pour avoir accepté de Co-encadrer ce travail.

Je remercie Farid Nasri, Professeur à l'Université de Batna et Med. El Hachemi Benbouzid, Professeur à l'Université de Brest, qui ont accepté d'évaluer ce travail. Je remercie également Mohamed Benmohammed, Professeur à l'Université de Constantine et Med. Chawki Batouche, Professeur à l'Université de Constantine, d'avoir acceptés de participer à mon jury de thèse.

Je tiens à consacrer quelques lignes aux personnes sans qui cette aventure n'aurait vraisemblablement jamais commencée : Mohamed BenMohamed, qui avait accepté d'encadrer mon mémoire de Magister, Mohamed Slimane, Professeur à l'Université de Tours, pour leur soutien lorsque cette thèse n'était encore qu'un projet lointain.

J'adresse ma gratitude à Hakan Cevikalp Professeur à l'Université d'Osmangazi qui a eu la gentillesse de m'accueillir au sein du laboratoire *Machine Learning & Computer Vision*.

Mes remerciements vont également à mes anciens collègues, Loucif Redha, Adouane Samir, Sahli Toufik, Kouda Hichem, Hidjazi Djalal, Aouag Soufiane, Haouassi Hichem, Benhizia Rafik et tous les autres qui sont trop nombreux pour être tous cités.

Je tiens à remercier tous les membres de notre laboratoire, mais également Abdelhadi Adel et Laghrour Warda pour leurs encouragements quotidiens. Je consacre une mention particulière à Fateh pour son soutien à un moment où j'en avais grand besoin et à Dida pour sa confiance et ses encouragements.

.

Résumé

Dans le cadre de cette thèse, nous avons proposé plusieurs modules de diagnostic pour des systèmes complexes et dynamiques. Ces modules sont basés sur trois algorithmes de colonie de fourmis qui sont *AntTreeStoch*, *Lumer & Faieta* et *Binay ant colony*. Ces algorithmes ont été choisis pour leur simplicité et leur vaste domaine d'application. Toutefois ces algorithmes ne peuvent pas être utilisés sous leur forme basale pour le développement de modules de diagnostic puisqu'ils présentent plusieurs limites. Nous avons aussi proposé plusieurs adaptations afin que ces algorithmes soient aptes à être utilisés dans des modules de diagnostic. Nous avons proposé une version parallèle de l'algorithme *AntTreeStoch* basée sur un système multi-agents réactifs. Cette version permet de minimiser l'influence de tri initial sur le résultat final de classification. Nous avons introduit aussi un nouveau paramètre appelé S_{id} qui permet à plusieurs fourmis de se connecter à la même position et nous avons modifié les déplacements des fourmis en favorisant le chemin de la fourmi la plus similaire. Pour l'algorithme *Lumer & Faieta*, nous avons accéléré la vitesse de construction de classes en ajoutant un paramètre de vitesse différent pour chaque fourmi. Pour réduire le nombre de déplacements, nous avons proposé une nouvelle variable qui permet de sauvegarder les identifiants des objets déplacés par la même fourmi. Pour améliorer la qualité de classification, nous avons ajouté aussi à l'algorithme des indices pour signaler les classes malles construites. Pour l'algorithme *Binay ant colony*, nous avons proposé une variante appelée « Hybrid wrapper/filter-based ACO-SVM ». Cet algorithme permet la sélection de paramètres. Il combine les techniques de méthodes filtres et enveloppantes en profitant de la rapidité du rapport de Fisher et l'adaptation de paramètres sélectionnés au classifieur SVM. Il améliore la qualité de classification en fonction de la nature de données de la base d'apprentissage et le type de la fonction noyau utilisée. Il permet aussi de régler les hyper-paramètres de la fonction noyau. Nous avons testé ces algorithmes sur des bases de données issues de deux systèmes industriels qui sont le système de clinkérisation et le système de pasteurisation, ainsi sur quelques bases de données d'UCI (*University of California, Irvine*).

Mots-clés

Diagnostic, Classification, Sélection de paramètres, Algorithmes de colonie de fourmis, Systèmes Multi-Agents.

Abstract

We have proposed three diagnostic modules for complex and dynamic systems. These three modules are based on ant colony algorithms that are AntTreeStoch, Lumer & Faieta, and Binay ant colony. These algorithms were chosen for their simplicity. However, these algorithms can't be used in their basal form for the development of diagnostic modules as they have several limitations. We have proposed several modifications to these algorithms are suitable for use in diagnostic modules. We proposed a parallel version of the algorithm AntTreeStoch based on a multi-agent. This version allows minimizing the influence of initial sorting on the final classification. We have also introduced a new parameter called Sid allowing several ants to connect to the same position and we changed the movement of ants in the way of promoting the most similar ant. For the algorithm Lumer & Faieta, we accelerated the speed of construction of classes by adding a different speed setting for each ant. To reduce the number of trips, we proposed a new variable that saves the identifiers of the objects moved by the same ant. To improve the quality of classification, we also added to the algorithm indices to indicate the classes built trunks. For the ant colony algorithm Binay, we proposed a variant called "hybrid wrapper / filter-based ACO-SVM". This algorithm allows the selection of parameters. It combines filters and wrapper methods to take advantage of the speed of the Fisher report and adaptation of selected parameters in SVM. It improves the quality of classification depending on the nature of data from the training set and the type of kernel function used. It can also set the hyperparameters of the kernel function. We tested the algorithms proposed in this manuscript on the databases from two industrial systems: the clinkering system and pasteurization system, and some UCI databases.

Key words: Diagnosis, Classification, Feature Selection, Ant Colony Algorithms, Multi-Agent Systems.

ملخص

في هذه الأطروحة، اقترحنا عدة نماذج لتشخيص الأنظمة المعقدة والديناميكية. وتستند هذه النماذج الثلاث على خوارزميات مستعمرة النمل التي هي أنترستوك، ليمر وفايتا و مستعمرة النمل الثنائية. وقد تم اختيار هذه الخوارزميات لبساطتها ولأن نطاق استعمالها واسع. ومع ذلك، لا يمكن أن تستخدم هذه الخوارزميات في أشكالها القاعدية لتطوير أنظمة التشخيص حيث أن لديهما العديد من القيود. في هذه الأطروحة، اقترحنا عدة تعديلات على هذه الخوارزميات هي مناسبة للاستخدام في أنظمة التشخيص. اقترحنا صيغة موازية للخوارزمية أنترستوك. هذا الإصدار يسمح بتقليل تأثير الفرز الأولي على التصنيف نهائي. أدخلنا أيضا تعديل جديد يسمح للنمل بعدة اتصالات في نفس المكان، وقمنا بتغيير حركة النمل في سبيل اختيار النملة المماثلة. من أجل تسريع تكوين المصنفات المشكلة من طرف الخوارزمية ليمر وفايتا تم إضافة إعداد سرعة مختلفة لكل نملة. اقترحنا أيضا متغير جديد الذي يحتفظ بالكائنات المنقولة من طرف نفس النملة. لتحسين نوعية التصنيف، إضافة إلى اختيار المؤشر بواسطة خوارزمية مستعمرة النمل الثنائية، اقترحنا متغير يتيح اختيار الوسائط. فهو يجمع بين السرعة والتكيف مع الوسائط المحددة في س.ف.م. ويحسن نوعية التصنيف تبعا لطبيعة بيانات مجموعة التدريب ونوع النواة المستخدمة. تم اختبار الخوارزميات المقترحة في هذه الأطروحة على أساس قاعدة بيانات اثنتين من النظم الصناعية هي نظام صنع الإسمنت ونظام بستره الحليب، وعلى بعض قواعد البيانات أخرى.

الكلمات المفتاح : التشخيص، التصنيف، اختيار المؤشرات، خوارزميات مستعمرة النمل، الأنظمة المعقدة والديناميكية.

Table des matières

Introduction générale

1	Contexte	1
2	Organisation du manuscrit	3
3	Contributions de la thèse	4

Chapitre 1: Diagnostic par reconnaissance de formes

Résumé	5	
1	Introduction :	6
2	Définitions	7
3	Les différentes approches de diagnostic	8
4	Diagnostic par reconnaissance de formes	10
4.1	Le principe de base	10
4.2	Analyse des observations :	11
4.3	Réduction de la dimension de l'espace de représentation	12
4.4	L'espace de décision :	14
4.4.1	La classification hiérarchique :	14
4.4.2	La classification par partition :	14
4.5	La procédure de décision :	15
4.5.1	Méthodes paramétriques :	16
4.5.2	Méthodes non paramétriques :	17
4.5.3	Calcul direct des frontières :	19
4.6	Phase d'exploitation	21
4.6.1	Les critères d'évaluation	21
4.6.2	Les méthodes d'évaluation	22
5	Outils d'intelligence artificielle pour le diagnostic industriel	23
5.1	Réseaux de Neurones	23
5.2	La logique floue	25
6	Conclusion :	26

Chapitre 2 : Les colonies de fourmis

Résumé :	27
1 Introduction :	28
2 Les fourmis réelles	28
2.1 Principe de base	28
2.1.1 La résolution de problèmes complexes	29
2.1.2 Stigmergie	29
3 Les fourmis artificielles	29
4 Les systèmes multi-agents	30
4.1 Principes de bases	30
4.2 Les agents	30
4.3 Architecture des agents	31
4.4 Les architectures réactives :	32
5 L'intelligence collective des fourmis	32
5.1 Le partage de tâches	32
5.2 Auto-organisation	33
5.3 La communication	33
5.3.1 La communication sonore	33
5.3.2 La communication tactile	33
5.3.3 La communication visuelle	34
5.3.4 La communication chimique	34
6 Les différents domaines d'application	34
7 Les algorithmes de colonies de fourmis et les SMA	36
7.1 L'algorithme AntTreeStoch	37
7.1.1 Les grandes lignes	37
7.1.2 L'algorithme principal d'AntTreeStoch	38
7.1.3 Cas d'une fourmi sur le support	38
7.1.4 Cas d'une fourmi placé sur une autre	39
7.1.5 Critiques et limites de AntTreeStoch	40
7.1.6 Les Améliorations	41
7.2 L'algorithme de Lumer & Faieta	44
7.2.1 Les grandes lignes	44

7.2.2	La distance.....	44
7.2.3	La classification.....	45
7.2.4	Critiques et limites de l’algorithme basal.....	46
7.2.5	Amélioration de l'algorithme.....	46
7.3	L’algorithme binaire de colonie de fourmis	47
7.3.1	Les grandes lignes	47
7.3.2	Critiques et limites de l’algorithme basal.....	48
7.3.3	Amélioration de l’algorithme	48
8	CONCLUSION	52

Chapitre 3 : Les Séparateurs à Vastes Marges (SVM)

Résumé	53
1 Introduction	54
2 Historique	54
3 Les SVM	55
3.1 L’hyperplan optimal	55
4 SVM : formulation primale	56
5 SVM : formulation duale.....	57
6 Le cas d'un échantillon non linéairement séparable	59
7 Les fonctions noyaux	60
8 Les hyper-parametres	63
9 SVM multi-classes	64
9.1 SVM : Un Contre Tous	64
9.2 SVM : Un Contre Un	65
9.3 DAG-SVM	66
10 La régression par SVM	66
11 Outils logiciels.....	68
11.1 SVM Torch.....	68
11.2 LIBSVM.....	69
11.3 SVM Light.....	69
12 Conclusion.....	69

Chapitre 4: La sélection de paramètres pour le diagnostic industriel

Résumé :	71
1 Introduction	72
2 Les étapes de la sélection de paramètres	72
2.1 Critères d'évaluation	73
2.1.1 Les mesures d'erreur de classification	73
2.1.2 Les mesures d'information	73
2.1.3 Les mesures de consistance	73
2.1.4 Les mesures de dépendance	74
2.1.5 Les mesures de distance	74
2.2 Les méthodes filtres	74
2.3 Les méthodes enveloppantes	74
2.4 Les méthodes Intégrées :	75
3 La procédure de génération	75
3.1 La génération complète	75
3.2 La génération aléatoire	76
3.3 La génération séquentielle	78
3.3.1 La méthode de génération constructive	78
3.3.2 La méthode de génération destructive	78
4 Critère d'arrêt	79
5 La validation	79
6 Extraction des paramètres	80
6.1 Principe	80
6.2 L'analyse en Composantes Principales	81
7 Procédure de sélection proposée	81
8 Conclusion	83

Chapitre 5 : Application et validation des approches proposées

Résumé :	85
1 Introduction	86
2 Description du premier procédé	86
2.1 Présentation de SCIMAT	86
3 La fabrication du ciment	87
3.1 La Clinkérisation	87
3.1.1 Le four rotatif	88
3.2 Analyse de fonctionnement d'un four rotatif	88
3.3 Les paramètres de la partie Clinkérisation	90
3.3.1 Modes de fonctionnement étudiés	91
4 Description du deuxième procédé	92
4.1 Présentation de laiterie des Aurès	92
4.2 Production du lait	93
4.3 Pasteurisation	93
4.3.1 Principe	93
4.3.2 Le pasteurisateur	94
4.3.3 Les principaux paramètres du processus de pasteurisation	95
4.3.4 Modes de fonctionnements étudiés	96
5 L'algorithme AntTreeStoch	97
5.1 Utilisation de Netlogo	97
5.2 Les données de test	97
5.3 Les paramètres de AntTreeStoch	98
5.4 Les mesures d'évaluation	99
5.5 Les résultats	100
6 Application de l'algorithme Lumer & Faieta	104
6.1 Configuration de l'algorithme	104
6.2 Résultats des scénarii :	106
7 Application de l'algorithme HYBRID wrapper/filter-based ACO-SVM	108
7.1 Les données de test	108
7.2 Le facteur heuristique FH	109
7.3 La configuration du classifieur	111
7.3.1 Réglage des hyper-paramètres	112

7.3.2	Les résultats de classification.....	113
7.4	Les résultats.....	115
8	Conclusion.....	121

Conclusion générale

Références Bibliographique

Annexe A

Annexe B

Annexe C

1	Présentation générale.....	XV
2	Description du modèle :	XV

Liste des figures

Chapitre 1

Fig. 1.1 : Position de diagnostic par rapport à la supervision	6
Fig. 1.2 : Classification des méthodes de diagnostic	9
Fig. 1.3 La position de RdF par rapport aux méthodes de Diagnostic	10
Fig. 1.4: Le processus de la reconnaissance de formes	11
Fig. 1.5: Diagnostic par reconnaissance de formes	15
Fig. 1.6 : La méthode d'estimation par noyau	18
Fig. 1.7 : Structure du réseau de type RBF pour le diagnostic.....	24

Chapitre 2

Fig. 2.1: La colonie choisit le chemin le plus court	28
Fig. 2.2 : L'expérience du double pont	29
Fig. 2.3: Modèle d'agent réactif	32
Fig. 2.4 : L'arbre couvrant de poids minimal d'un graphe planaire.....	35
Fig. 2.5 : Un problème de tournées de véhicules avec un dépôt central	36
Fig. 2.6 : Une exécution séquentielle d'AntTreeStoch	40
Fig. 2.7 : Une exécution parallèle d'AntTreeStoch.....	41
Fig. 2.8 : Une exécution parallèle d'AntTreeStoch avec le paramètre S_{id}	42
Fig. 2.9 : Résultat possible de l'exécution de l'algorithme LF sur la grille	44
Fig. 2.10 : Résultats possibles de l'exécution de LF avec et sans des fourmis rapides	47
Fig. 2.11 : Le choix de la dimension de vecteur.....	48
Fig. 2.12 : L'ensemble de paramètres sélectionnés par l'algorithme Filter based ACO.....	49

Chapitre 3

Fig. 3.1: Classification des méthodes d'apprentissage à base de noyaux	54
Fig. 3.2: Séparation de deux ensembles de points par des séparateurs linéaires.....	56
Fig. 3.3: L'hyperplan optimal est perpendiculaire au segment de droite le plus court joignant un exemple d'apprentissage à l'hyperplan	56
Fig. 3.4 : Exemple de projection non-linéaire.....	59
Fig. 3.5 : Exemple de classification SVM multi-classes.....	64
Fig. 3.6 : Architecture d'un système de classification basée sur l'algorithme Un-Contre-Tous	65
Fig. 3.7 : une observation appartient à la zone d'ambiguïté	65
Fig. 3.8 : Une classification de trois classes par SVM-DAG.....	66
Fig. 3.9 : La régression par SVM.....	67

Chapitre 4

Fig. 4.1: Principe de la sélection de variables	72
Fig. 4.2: Les différentes étapes d'une procédure de sélection d'attributs	73
Fig. 4.3 : Principe d'extraction de variables	80
Fig. 4.4: Organigramme de l'algorithme Hybrid wrapper/filter-based ACO-SVM.....	82
Fig. 4.5: Les solutions finales proposées par l'algorithme Hybrid wrapper/filter-based ACO-SVM	83

Chapitre 5

Fig. 5.1: Représentation schématique du principe de fabrication du ciment	87
Fig. 5.3 : installation de cuisson.....	88
Fig. 5.6 : Le cheminement de lait dans le processus de fabrication.....	93
Fig. 5.7 : La base de données BD1 du système de clinkérisation	98

Fig. 5.8 : Le modèle multi-agents de l'algorithme AntTreeStoch	99
Fig. 5.9 : Exemple d'arbre obtenu par AntTreeStoch sur la base IRIS	102
Fig. 5.10 : Vue en 3D de la base IRIS	102
Fig. 5.11: Visualisation des quatre modes de dysfonctionnement du premier procédé en utilisant 3 paramètres	103
Fig. 5.12 : Visualisation 3D de l'arbre de classification de la base BD1 (500 nœuds) et de la base BD2 (300 nœuds)	104
Fig. 5.13 : Interface de configuration de LF.....	105
Fig. 5.14 : Modèle de simulation de l'algorithme Lumer & Faieta	106
Fig. 5.15 : Résultat de la simulation de l'algorithme Lumer & Faieta avec ($S \geq 5$).....	107
Fig. 5.16 : Simulation de l'algorithme LF avec et sans des fourmis rapides (BD1)	107
Fig. 5.17 : Simulation de l'algorithme LF avec et sans des fourmis rapides (BD2).....	108
Fig. 5.18 : Le facteur heuristique FH de la base BD2	110
Fig. 5.19 : Le facteur heuristique FH de la base Vehicle	110
Fig. 5.20 : Le facteur heuristique FH de la base RCK1	111
Fig. 5.21 : Le facteur heuristique FH de la base RCK2	111
Fig. 5.22 : Résultat de classification de la base de données Ex8a.....	113
Fig. 5.23 : Résultat de classification de la base de données Four_classe.....	114
Fig. 5.24 : Résultat de classification de la base de données Heart_scale	114
Fig. 5.25 : Résultat de classification de la base de données du système de pasteurisation de lait	115
Fig. 5.26 : Les valeurs de FV dans la dernière itération (Vehicle)	117
Fig. 5.27 : La meilleure solution obtenue dans chaque itération (BD2)	118
Fig. 5.28 : La valeur de FV obtenue par chaque agent dans la dernière itération (RCK1)....	118
Fig. 5.29 : Le meilleur résultat obtenu à la fin de chaque itération (RCK1).....	119
Fig. 5.30 : Le meilleur résultat obtenu à la fin de chaque itération (RCK2).....	119

ANNEXES

Fig. A.1 : Interface graphique du modèle.....	xv
Fig. A.2 : Présentation graphique de la solution	xviii

Liste des tableaux

Chapitre 1

Tableau 1.1 : Exemple d'un ensemble de données d'apprentissage.....	11
Tableau 1.2 : La table de contingence.....	22

Chapitre 3

Tableau 3.1 : Les noyaux SVM les plus utilisés dans le domaine de classification	63
--	----

Chapitre 4

Tableau 4.1 : Principe générale d'un algorithme génétique.....	76
--	----

Chapitre 5

Tableau 5.1 : Principaux paramètres de la partie Clinkérisation.....	90
Tableau 5.2: les différentes classes de fonctionnement	91
Tableau 5.3 : Les différentes opérations de pasteurisation.....	95
Tableau 5.4 : Les principaux paramètres du processus de pasteurisation.....	96
Tableau 5.5 : Les modes de fonctionnements du processus de pasteurisation.....	96
Tableau 5.6 : Les Paramètres d'AntTreeStoch.....	99
Tableau 5.7 : Résultats obtenus par une implémentation séquentielle et une autre parallèle de l'algorithme AntTreeStoch	100
Tableau 5.8 : L'arbre final obtenu par AntTreeStoch avec et sans le permettre S_{id}	101
Tableau 5.9 : les détails de bases de données utilisées dans les tests.....	109
Tableau 5.10 : Les paramètres de l'algorithme de sélection	109
Tableau 5.11 : La description de bases de données.....	112

Tableau 5.12 : Résultats obtenus pour les différentes valeurs de couple (C, σ)	112
Tableau 5.13 : Les entrées de GATOOOL.....	116
Tableau 5.14 : Les performances de classification en utilisant les différentes entrées	117
Tableau 5.15 : Description des paramètres sélectionnés.....	120

ANNEXES

Tableau A.1: La procédure de configuration	xvi
Tableau A.2 : La procédure d'initialisation	xvi
Tableau A.3 : La procédure d'affichage	xvi
Tableau A.4 : La fonction d'adaptation	xvii
Tableau A.5 : Création d'une nouvelle génération	xvii
Tableau A.6 : La procédure de croisement	xviii
Tableau A.7 : La procédure de mutation.....	xviii

INTRODUCTION GENERALE

Introduction générale

1 Contexte

Ce travail a été réalisé au sein de l'équipe 3S (Surveillance – Supervision- Sûreté) du LAP de l'université de Batna dans le cadre d'un projet de type CNEPRU intitulé « L'intelligence artificielle, les méthodes bio-inspirées, NTIC et les méthodes d'émergences : outils pour la supervision et la e-maintenance des systèmes de production » identifié sous le code B*03520100014.

Les objectifs de notre groupe sont la proposition des méthodes et le développement de logiciels pour la détection et localisation des défauts dans les systèmes de production en utilisant les techniques de reconnaissance de formes.

Le développement technologique des outils de production a conduit à l'apparition de systèmes industriels très complexes et dynamiques. Tout système industriel peut passer directement d'un mode de fonctionnement normal à un autre défaillant qui correspond à l'altération ou à la cessation de l'aptitude du système à accomplir une fonction de production. Pour éviter ce problème, on peut utiliser une stratégie de maintenance systématique en remplaçant des éléments du système selon un échéancier. Comme cette stratégie exige un arrêt total ou partiel du système de production, elle présente un inconvénient majeur qui est la réduction du temps de disponibilité des outils de production et conduit à son tour à une diminution de productivité. La solution idéale est d'utiliser une maintenance conditionnelle qu'on applique seulement dans le cas d'un dysfonctionnement. En utilisant ce type de maintenance, on peut améliorer le rendement du système et limiter le coût d'entretien. Néanmoins, avec ce type de maintenance, on doit toujours utiliser un système de diagnostic qui indique l'état de fonctionnement à tout instant.

On peut distinguer deux types de méthodes de diagnostic. Le premier type englobe les méthodes de diagnostic à base de modèles analytiques et qui se basent sur le calcul de l'écart entre les paramètres et les grandeurs de la machine et ceux du modèle pour détecter une éventuelle défaillance. Cette action est réalisée en utilisant des algorithmes d'observation et de comparaison. Ces méthodes ne nécessitent pas de faire des hypothèses à priori sur les différents modes de défaillance que peut subir le système et offrent la possibilité de prévoir

des dégradations des performances du système. Le deuxième type d'approche de diagnostic regroupe les méthodes sans modèles. Ces méthodes sont construites en utilisant l'historique de fonctionnement du système et l'expérience des experts. Le principe de base de ces méthodes est d'extraire des informations à partir des signaux mesurés. Ces informations sont utilisées pour la mise en œuvre du système de diagnostic. L'efficacité du système de diagnostic est liée à la pertinence des indicateurs de défauts retenus et au prétraitement des données utilisé. La reconnaissance de formes est une méthode très connue de ce deuxième type. Elle est la plus adaptée pour réaliser le diagnostic de systèmes complexes. Le principe de cette méthode est d'affecter toute nouvelle observation à une classe qui représente un mode de fonctionnement ou de dysfonctionnement du système. Il existe plusieurs méthodes de RdF. Nous avons choisi comme méthode de classification les algorithmes de colonies de fourmis parce qu'ils sont basés sur des principes simples. Nous avons aussi utilisé ce type d'algorithme pour résoudre le problème de la sélection de paramètres qui représente la phase la plus importante dans le processus de diagnostic des systèmes complexes et dynamiques. Ces systèmes possèdent un comportement non linéaire et non stable durant leurs exécutions. Leur modélisation est très difficile par l'approche mathématique puisqu'ils possèdent un nombre énorme de variables qui caractérisent leurs états de fonctionnement.

La problématique, à laquelle cette thèse se propose de répondre, est l'adaptation des algorithmes de colonies de fourmis aux spécificités de données issues de domaine industriel afin de développer des modèles de diagnostic qui soient performants et dont les résultats soient facilement interprétables par l'utilisateur.

Parmi les algorithmes de colonies de fourmis, nous nous intéressons à trois algorithmes qui sont AntTreeStoch, Lumer & Faieta et Binay ant colony. Ces algorithmes ne peuvent pas contribuer à résoudre le problème de diagnostic des systèmes dynamiques et complexes puisqu'elles présentent un ensemble de faiblesses.

- La faiblesse majeure de l'algorithme AntTreeStoch réside dans le tri initial de données qui exige un temps de calcul très important. Un autre inconvénient de cet algorithme est le choix aléatoire de chemin qui conduit à une mauvaise classification de données.
- L'algorithme Lumer & Faieta est très coûteux en terme de temps puisque selon son principe de base les objets peuvent être déplacé plusieurs fois même s'ils sont bien classés.

- Le principal inconvénient de l'algorithme Binay ant colony est sa grande lenteur puisqu'il exige plusieurs exécutions d'un algorithme génétique pour l'initialisation de phéromone sur le réseau et d'autres exécutions dans chaque itération.

Dans ce cadre, nous avons proposé d'améliorer ces trois algorithmes, d'une part en augmentant la vitesse de décision et d'autre part en perfectionnant la qualité de classification.

2 Organisation du manuscrit

Cette thèse, composée de cinq chapitres, débute par un premier chapitre relatif au diagnostic par reconnaissance de formes. Il présente les concepts fondamentaux de l'approche de diagnostic de systèmes complexes et dynamiques par la reconnaissance de formes. L'objectif de ce chapitre est de mettre en évidence l'utilisation de méthodes de classification pour le diagnostic de systèmes industriels complexes et dynamiques.

Le deuxième chapitre est composé de deux parties. La première partie constitue une étude bibliographique sur les algorithmes de colonie de fourmis. Dans la deuxième partie, nous proposons de nouvelles implémentations de quelques algorithmes que nous avons utilisés pour le développement d'un système de diagnostic industriel.

Dans le troisième chapitre, nous présenterons quelques notions de base de machines à vecteurs de support (SVM) qui sont Hyperplan, marge et support vecteur. Nous mettrons également en évidence l'utilisation des SVM dans les deux cas où les données sont (linéairement ou non linéairement) séparables. Nous terminons par une présentation de quelques algorithmes pour la classification SVM multi-classes.

Le quatrième chapitre discute le problème de la sélection de paramètre et sa position dans le processus de diagnostic. Il présente en détail les deux approches utilisées pour la réalisation de l'étape de réduction de la dimension du vecteur d'état qui sont la sélection de paramètres et l'extraction de paramètres. Il présente également une nouvelle approche de sélection de paramètres basée sur les machines à vecteurs de support et les algorithmes de colonie fourmis.

Nous terminons le document avec un chapitre dédié d'abord à l'étude de deux procédés industriels: le système de clinkérisation de la SCIMAT (Société des CIMents de Ain Touta) et le système de pasteurisation d'ULA (Unité de Laiterie Aurès de Batna). Dans une deuxième partie nous présentons les résultats de l'application des algorithmes proposés sur ces systèmes industriels et sur quelques bases de données.

3 Contributions de la thèse

Les contributions de la thèse se situent au niveau des points suivants :

- Un état de l'art sur le diagnostic industriel par reconnaissance de formes, les algorithmes de colonie de fourmis, la réduction de la dimension de vecteurs d'état par la sélection de paramètres et la classification de données par les machines à vecteurs de support.
- La proposition d'une version parallèle pour l'algorithme AntTreeStoch pour minimiser l'influence de tri initial sur la qualité de classification. Pour minimiser la taille de l'arbre de données classées. Nous avons proposé une variante de cet algorithme en ajoutant un nouveau paramètre appelé S_{id} qui permet à plusieurs fourmis de se connecter à la même position et nous avons modifié les déplacements des fourmis en favorisant le chemin de la fourmi la plus similaire.
- La proposition d'une variante de l'algorithme Lumer & Faieta en créant plusieurs sous-groupes de fourmis selon la vitesse de déplacement. Nous avons ajouté une petite mémoire à chaque fourmi pour minimiser le nombre de comparaisons. Nous avons ajouté aussi à l'algorithme un indice pour signaler un groupe d'objets non homogène.

L'originalité de ce travail réside dans la proposition d'un nouvel algorithme de sélection de paramètres appelé « Hybrid wrapper/filter-based ACO-SVM ». L'algorithme proposé combine les techniques de méthodes filtres et enveloppantes en profitant de la rapidité du rapport de Fisher et l'adaptation de paramètres sélectionnés au classifieur SVM (les machines à vecteurs de support).

Cet algorithme développé dans le cadre de nos travaux de recherche améliore la qualité de classification en fonction de la nature de données de la base d'apprentissage et le type de la fonction noyau utilisé.

Il permet également de régler les hyper-paramètres de la fonction noyau.

DIAGNOSTIC PAR RECONNAISSANCE DE FORMES

Résumé

Ce chapitre est consacré à la définition des concepts fondamentaux de l'approche de diagnostic de systèmes complexes et dynamiques par la reconnaissance de formes. Nous décrivons dans un premier temps la position du diagnostic par rapport à la fonction de surveillance. Nous traitons ensuite le diagnostic industriel comme un problème de reconnaissance de formes. Nous terminons ce chapitre par une présentation de deux outils d'intelligence artificielle très utilisés dans le domaine de diagnostic industriel.

1 Introduction :

Les systèmes complexes et dynamiques possèdent un comportement non linéaire et non stable durant leurs exécutions. Il est très difficile voire impossible de modéliser ce type de systèmes par l'approche mathématique. Cette difficulté liée au grand nombre de variables qui caractérisent l'état de fonctionnement du système. Notre travail concerne la surveillance de ce type de systèmes et plus précisément la phase de diagnostic. La fonction principale de la surveillance est de vérifier l'état de fonctionnement du système. Elle est composée de deux parties qui sont la détection et le diagnostic, (figure 1.1). La phase de détection permet de déterminer l'état du système comme étant normal ou anormal. La phase de diagnostic consiste à identifier les composants défectueux et de trouver les causes à partir d'un ensemble de symptômes observés.

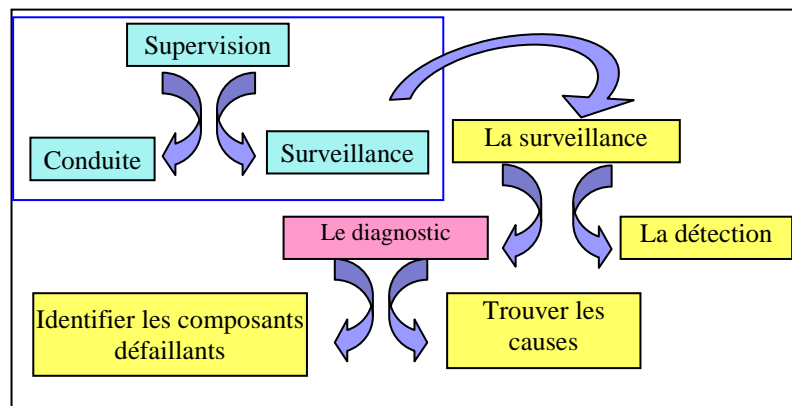


Fig. 1.1 : Position de diagnostic par rapport à la supervision

D'un point de vue général, la détermination des causes à partir des effets est un problème de classification. Puisque, il s'agit de comparer un vecteur de données qui représente l'état actuel du système avec des vecteurs références qui représentent l'historique de fonctionnement du système.

Plusieurs chercheurs ont montré l'intérêt de la reconnaissance de formes dans différents domaines où les systèmes sont de nature complexe tels que la reconnaissance d'image, la reconnaissance d'écriture, le diagnostic médical, le diagnostic industriel, etc. [Fasel et Luettin, 2003] [Plotz et Fink, 2009] [Mahdaoui et. Mouss, 2008] [Dazzi et al., 2001].

Il existe plusieurs avantages qui rendent l'utilisation de la reconnaissance de formes intéressante comme la rapidité de calcul pour la reconnaissance d'une nouvelle information, l'aptitude à traiter des données incertaines et imprécises. Et aussi, la RdF permet de détecter et de suivre l'évolution de fonctionnement de système [Ondel et al., 2006].

2 Définitions

Cette section vise à rappeler la terminologie utilisée dans la littérature et en particulier celle relative à nos travaux [Mouss, 2006].

- **La Supervision** est une tâche de commande et de surveillance de l'exécution d'une opération ou d'un travail effectué par d'autres personnes sans rentrer dans les détails de cette exécution.
- **La Surveillance** est une tâche contenue en temps réel dont le but est de caractériser le mode de fonctionnement du système physique, en enregistrant des informations, en reconnaissant et en indiquant les anomalies de comportement.
- **La Maintenance** est un ensemble des actions techniques et administratives correspondantes, y compris les opérations destinées à maintenir ou rétablir une entité dans un état spécifié ou dans des conditions données de sûreté de fonctionnement lui permettant d'accomplir une fonction requise.
- **Le Diagnostic** est une fonction visant à fournir des informations sur les anomalies au sein d'un système physique. On distingue traditionnellement plusieurs niveaux de diagnostic : détection, localisation et identification des anomalies.
- **La Détection** est le premier niveau de diagnostic consistant à déterminer de façon rapide et fiable l'existence d'une anomalie.
- **L'Identification** est un niveau de diagnostic consistant à caractériser précisément les anomalies qui se sont produites.
- **La Localisation** est un niveau de diagnostic, déclenché par une procédure de détection et consistant à déterminer les anomalies qui se sont produites.
- **La Défaillance** est l'altération ou la cessation de l'aptitude d'une entité à accomplir une fonction requise. La défaillance est un passage d'une entité d'un état de fonctionnement normal à un état anormal ou de panne.
- **Le Défaut** est l'écart entre une caractéristique réelle d'une entité et la caractéristique voulue. Cet écart dépassant des limites d'acceptabilité.
- **La Panne** est l'inaptitude d'une entité à accomplir une fonction requise ou à assurer le service approprié à la suite d'une défaillance.

- **La Reconnaissance de Formes (RdF)** est une science qui regroupe l'ensemble des algorithmes ou méthodes permettant la classification d'objets ou de formes en les comparant à des formes-types.
- **Un vecteur forme** est constitué d'un ensemble de paramètres qui peuvent être numériques ou symboliques.
- **Un prototype** est l'ensemble des valeurs précises des paramètres qui caractérisent une situation particulière.
- **Une classe** est l'ensemble possible de valeurs des attributs.
- **La classification d'un objet** est la décision d'affecter l'objet à une classe particulière.

3 Les différentes approches de diagnostic

Il existe deux types d'approches de diagnostic [Adrot, 2000]. Le premier réunit les méthodes de diagnostic à base de modèles analytiques. Pour détecter une éventuelle défaillance, ces méthodes utilisent un calcul de l'écart entre les paramètres et les grandeurs de la machine et ceux du modèle. Ces méthodes utilisent des algorithmes d'observation et de comparaison. Le deuxième type d'approche de diagnostic regroupe les méthodes sans modèles. Ces méthodes sont appliquées quand on ne peut pas construire un modèle fonctionnel du système à cause de l'absence totale de connaissance analytique et structurelle du système. Dans ce cas un modèle est construit en utilisant l'historique de fonctionnement du système et l'expérience des experts. Le principe de base de ces méthodes est d'extraire des informations à partir des signaux mesurés. Ces informations sont utilisées pour la mise en œuvre du système de diagnostic. L'efficacité du système de diagnostic est liée à la pertinence des indicateurs de défauts retenus et au prétraitement des données utilisées.

Une organisation des méthodes de diagnostic qui permet d'avoir une vision générale a été présentée dans [Narvaez, 2007]. La Figure 1.2. présente sa classification des méthodes de diagnostic.

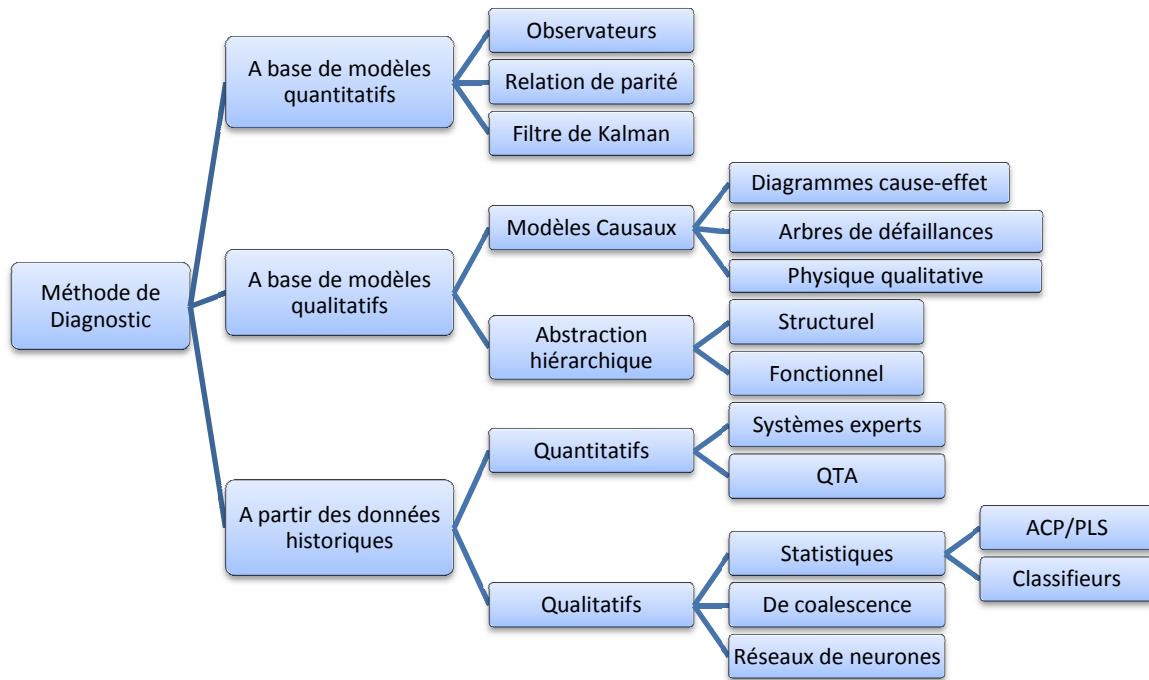


Fig. 1.2 : Classification des méthodes de diagnostic [Narvaez, 2007]

La figure 1.2. montre qu'il existe plusieurs méthodes pour construire un module de diagnostic d'un système industriel. La méthode utilisée dépend de la manière de représentation des données du système ainsi que le degré de connaissance de l'expert sur les différents modes de fonctionnement et de dysfonctionnement du système.

L'approche par modèle mathématique [Adrot, 2000] est utilisée pour le diagnostic si les données sont de nature numérique et on dispose d'un modèle mathématique du système. Le modèle mathématique compare les valeurs des observations avec des seuils pour détecter un état de dysfonctionnement. L'avantage majeur de ces méthodes est la capacité à détecter des pannes à travers une analyse de tendance des signaux ainsi la capacité de donner une localisation exacte du composant défaillant. Néanmoins, ces méthodes souffrent de plusieurs inconvénients, nous pouvons citer entre autres : la nécessité d'avoir une connaissance complète sur les différents modes de fonctionnement et de dysfonctionnement du système. Elles sont également très sensibles aux erreurs de modélisation.

Si les informations sont de nature symbolique l'approche par système expert est préférée [Mouss, 2006]. Ce type d'approche (quantitative) est adapté aux problèmes qui exigent le traitement d'une grande quantité de données hétérogènes et/ou contextuelles. On utilise une approche basée sur la reconnaissance de formes [Mahdaoui et Mouss, 2008] si on ne peut pas construire un modèle mathématique représentant les différents modes de fonctionnement et de

dysfonctionnement du système étudié ou si ce modèle est très complexe qui rend son utilisation est impossible.

Les méthodes basées sur la reconnaissance de formes permettent le développement d'un système de diagnostic même si le système étudié est complexe (figure 1.3).

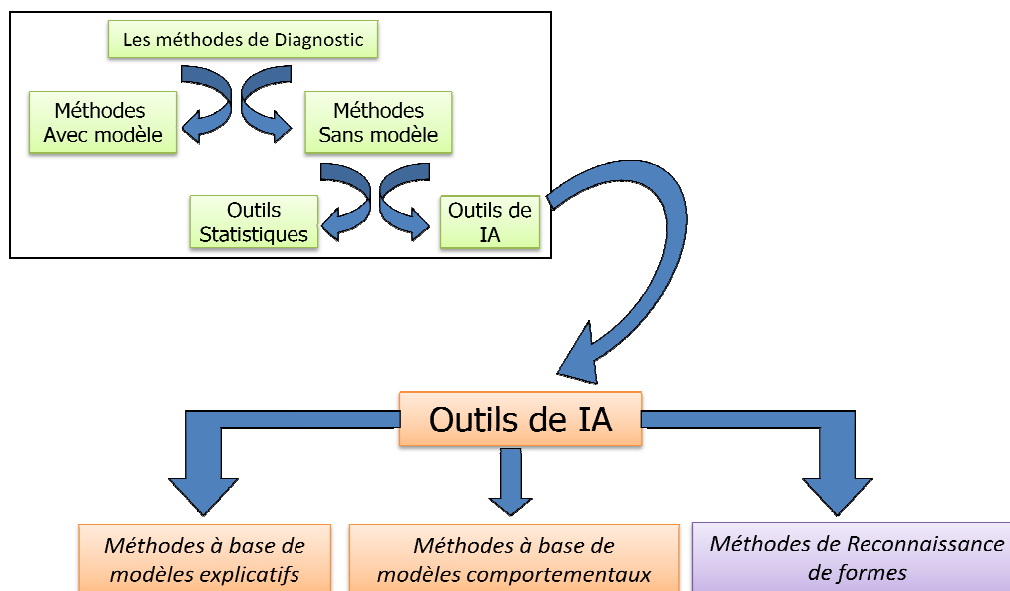


Fig. 1.3 La position de RdF par rapport aux méthodes de Diagnostic

Pour les systèmes étudiés dans le cadre de cette thèse, nous croyons qu'une approche basée sur la reconnaissance de formes est la plus adaptée puisque les deux procédés étudiés dans le cadre de cette thèse représentent des systèmes dynamiques et complexes [Kadri et al., 2012a].

4 Diagnostic par reconnaissance de formes

4.1 Le principe de base

Le diagnostic d'un système industriel cherche à identifier d'une façon permanente l'état de fonctionnement du système. On peut le définir comme un problème de reconnaissance de formes (RdF). Il a pour but la reconnaissance d'une forme parmi plusieurs possibilités à partir d'observations bruitées. On peut distinguer deux approches [Mokhtari, 2007] :

- La reconnaissance de formes numérique (statistique) qui exploite des modélisations de formes probabilistes ou floues ;
- La reconnaissance de forme structurale ou syntaxique qui exploite les relations entre les composants de la forme.

En RdF statistique, chaque état de fonctionnement (observation) est représentée par un vecteur forme x de d paramètres tel que $x = (x_1, \dots, x_d)$. Le vecteur x est représenté par un point dans un espace de représentation appelé \mathbb{R}^d . Les paramètres x_i du vecteur forme x traduisent l'état du système étudié. Ils sont issus des traitements réalisés sur les signaux fournis par les capteurs implantés (vibrations, vitesse, courants, etc.). Pour chaque nouvelle observation nous avons à décider parmi M classe qui correspondent à des zones dans l'espace de représentation, regroupant les formes semblables.

La figure 1.4 montre qu'un système de RdF est composé de 5 étapes principales : acquisition des données, génération de caractéristiques, extraction/sélection des paramètres, classification et évaluation du système.

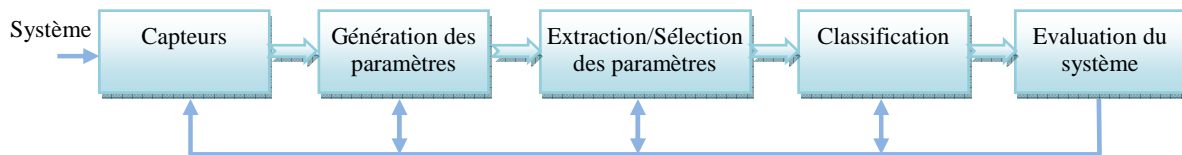


Fig. 1.4: Le processus de la reconnaissance de formes [Saint-Jean, 2001]

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. Cet ensemble peut s'écrire sous la forme matricielle (tableau 1.1).

Tableau 1.1 : Exemple d'un ensemble de données d'apprentissage

	Paramètre 1	...	Paramètre j	...	Paramètre d	Classes
Entité 1	12	...	18	...	20	Classes 1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Entité i	5	...	6	...	40	Classes k
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Entité n	1	...	1	...	60	Classes m

La phase d'analyse est très coûteuse en temps de calcul et nécessite la connaissance de plusieurs informations concernant le système étudié pour pouvoir sélectionner les paramètres adéquats et optimaux à un meilleur diagnostic.

4.3 Réduction de la dimension de l'espace de représentation

L'efficacité d'un système de diagnostic dépend de la pertinence des paramètres choisis. Il est donc indispensable de trouver les paramètres qui permettent de distinguer les différents modes de fonctionnement et de dysfonctionnement du système.

L'utilisation d'un nombre énorme de paramètres influe sur la vitesse d'identification. Il est aussi possible d'identifier les différents états de fonctionnement d'un système en utilisant un ensemble réduit de paramètres [Kadri et al., 2010a].

Une méthode de réduction doit être utilisée afin de minimiser la dimension du vecteur d'état et ne conserver que les paramètres jugés pertinents. Cette méthode permet de diminuer le temps d'identification d'une nouvelle observation et d'éviter une dégradation des performances du système de diagnostic causé par l'utilisation d'un nombre énorme de paramètres.

Il existe deux types de méthodes de réduction de dimension : le premier consiste à définir un nouvel ensemble de paramètres. Ils constituent des combinaisons linéaires ou non linéaires des anciens. Ces méthodes sont appelées les méthodes d'extraction [Duda, 2001]. Les plus utilisées sont l'Analyse en Composante Principale (ACP) [Sophian et al., 2003], l'Analyse Factorielle Discriminante (AFD) [Lebart et al., 2000], le réseau de quantification vectorielle et projection (VQP) [Mathieu et al., 2000] et la projection de Sammon [Jain et Mao, 1994].

Le deuxième type de méthodes de réduction de dimension consiste à sélectionner un sous-ensemble de paramètres depuis l'ensemble initial. Le nouvel ensemble doit offrir une minimisation de la variance intra-classe et une maximisation de la variance interclasse. Ces méthodes sont appelées méthodes de sélection. Toute méthode de sélection de paramètres consiste en quatre points essentiels [Al-Ani, 2005] :

- **Un ensemble de départ** qui représente l'ensemble de paramètres initialement utilisés par une procédure de recherche. Cet ensemble peut être vide, ou contient tous les paramètres ou bien un sous-ensemble aléatoire.
- **La procédure de recherche** est l'élément essentiel de toute méthode de sélection. Elle retourne comme résultat le sous-ensemble de paramètres qui répondent mieux au critère de qualité. Ce critère est utilisé par une fonction d'évaluation.
- **La fonction d'évaluation** détermine la qualité de classification obtenue en utilisant un sous-ensemble de paramètre.

- **Un critère d'arrêt** est utilisé pour terminer la procédure de recherche. Ce critère dépend soit de la fonction d'évaluation ou bien aux paramètres de configuration qui sont définis par l'utilisateur.

La sélection de m attributs parmi n représente un problème d'explosion combinatoire. Il existe plusieurs approches permettant d'éviter une recherche exhaustive. Parmi les premiers travaux dans ce domaine ont été ceux de Narendra et Fukunaga [Narendra et Fukunaga, 1977], ils ont utilisé l'algorithme branch and bound qui exige la monotonie du critère d'évaluation employé. Cela veut dire que l'ajout d'un nouveau paramètre au sous-ensemble actuel ne doit pas décroître la valeur du critère. Mais en réalité, cette condition n'est pas vérifiée par la plupart des critères utilisés dans le contexte de sélection de variables. Malgré les différentes variantes proposées à cet algorithme, ils ne sont pas applicables devant les problèmes où le nombre des paramètres est très élevé.

Il existe une autre catégorie d'algorithmes appelée Wrappers [Guyon et Elisseeff, 2003]. Les deux célèbres algorithmes de cette catégorie sont Sequential Forward Selection (SFS) [Blum et Langley, 1997] et Sequential Backward Selection (SBS) [Kittler, 1978]. Le point de départ de ces algorithmes est respectivement un ensemble vide ou complet de paramètres. Initialement, tous les paramètres sont évalués individuellement et on sélectionne celui qui conduit à la meilleure performance de classification. On répète cette itération jusqu'à ce que la qualité de classification n'augmente pas. La différence entre les deux algorithmes réside dans le fait que dans le premier, on ajoute un paramètre à chaque itération par contre on supprime un paramètre dans le deuxième. L'inconvénient majeur de cette catégorie est la forte dépendance entre le résultat final et l'ordre initial des paramètres. Yang et Honavar [Yang et Honavar, 1998] ont appliqué les algorithmes génétiques pour la sélection de paramètres. La population initiale est générée aléatoirement et représente un ensemble des solutions possibles. Le fitness est utilisé comme un critère d'évaluation. Le passage d'une génération à une autre est réalisé en appliquant la sélection, le croisement et la mutation. Le but de ces transformations est de converger vers une solution optimale. Les algorithmes génétiques ont donné des résultats plus intéressants que les méthodes précédentes. Nous proposons dans le deuxième chapitre une nouvelle méthode de sélection de paramètres basée sur une variante de l'algorithme de colonie de fourmis (ACO) [Weiqing et al., 2006]. Nous montrons par la suite que les résultats obtenus par cette approche sont comparables aux résultats obtenus en appliquant les algorithmes génétiques.

4.4 L'espace de décision :

Dans cette phase, on partitionne les données d'apprentissage en groupes en utilisant une méthode de classification. Dans une seconde étape, on établit une relation entre ces groupes et les différents états de fonctionnement du système. Donc l'objet de cette phase est de trouver un assemblage en classes de l'ensemble des données par la mise en place de procédures de classification. La classification est dite exclusive lorsque le partitionnement des objets conduit à associer chaque objet à une et une seule classe. Les méthodes de cette catégorie se décomposent en deux parties : les méthodes hiérarchiques (ascendantes ou descendantes) et les méthodes de classification par partition. Par contre, une classification non exclusive admet qu'un objet appartienne à plusieurs classes. Sur un autre axe, on peut aussi distinguer la classification supervisée où les objets disposent d'une information qui indique leurs dissimilarités. En classification non supervisée, on n'utilise qu'une matrice de dissimilarité. Aucune information sur la classe d'un objet n'est fournie à la méthode [Kadri et *al.*, 2010a].

4.4.1 La classification hiérarchique :

Ce type de classification se base sur le calcul de distances entre les informations pour construire des classes. A la fin de la procédure de classification, On obtient un arbre dont les informations de chaque nœud correspondent à un enregistrement de la base d'apprentissage. Pour effectuer une partition, on coupe cet arbre à un niveau donné (chaque niveau de coupure correspond à une partition). L'algorithme de classification ascendante procède par fusions successives des éléments, alors que la classification descendante procède de façon inverse. Elle considère la base d'apprentissage comme une classe globale, et la fractionne en deux sous classes. La division s'opère de façon à ce que la distance entre les deux sous classes soit la plus grande possible, de façon à créer deux classes bien séparées. En pratique, ces algorithmes posent des difficultés calculatoires pour les bases de données de grandes tailles. Cependant, nous proposons dans le prochain chapitre des améliorations pour un algorithme de ce type pour qu'il soit plus rapide et utilisable en ligne [Chen et *al.*, 2002].

4.4.2 La classification par partition :

Dans ce type de classification, une classe représente une zone de l'espace qui regroupe les observations semblables. La classification par partition est généralement réalisée grâce à des méthodes de coalescence [Azzag et *al.*, 2004]. Le résultat de l'exécution de ces méthodes est un ensemble de classes dont la similarité à l'intérieur d'une même classe est élevée mais faible entre les différentes classes. Dans une première étape ces méthodes calculent les centres

de gravité des classes et ensuite elles affectent chaque observation au centre de gravité le plus proche.

4.5 La procédure de décision :

A la fin de l'étape précédente, on obtient un ensemble d'apprentissage structuré en groupes séparés. Les observations sont affectées de telle sorte que chacune d'entre elles appartient à une des M classes. Dans plusieurs cas, l'affectation ne doit pas être systématique car des erreurs peuvent être commises. La procédure de décision doit proposer plusieurs solutions pour minimiser le risque de mauvaise identification. Ces solutions sont fournies par les options de rejets d'ambiguïté et de distance (figure 1.5) [Mokhtari, 2007]. Le rejet d'ambiguïté consiste à filtrer les observations ambiguës. Il affecte les observations qui se trouvent dans cette situation à une nouvelle classe fictive appelée classe de rejet d'ambiguïté. Tandis que le rejet de distance concerne les observations aberrantes qui ne correspondent à aucune des classes du système. Les observations se trouvant dans cette situation sont affectées à une nouvelle classe fictive appelée classe de rejet d'ambiguïté.

La possibilité de prendre en considération de nouvelles observations non rencontrées auparavant permet de minimiser le risque de mauvaise classification et représente une alternative nécessaire en diagnostic (figure 1.5).

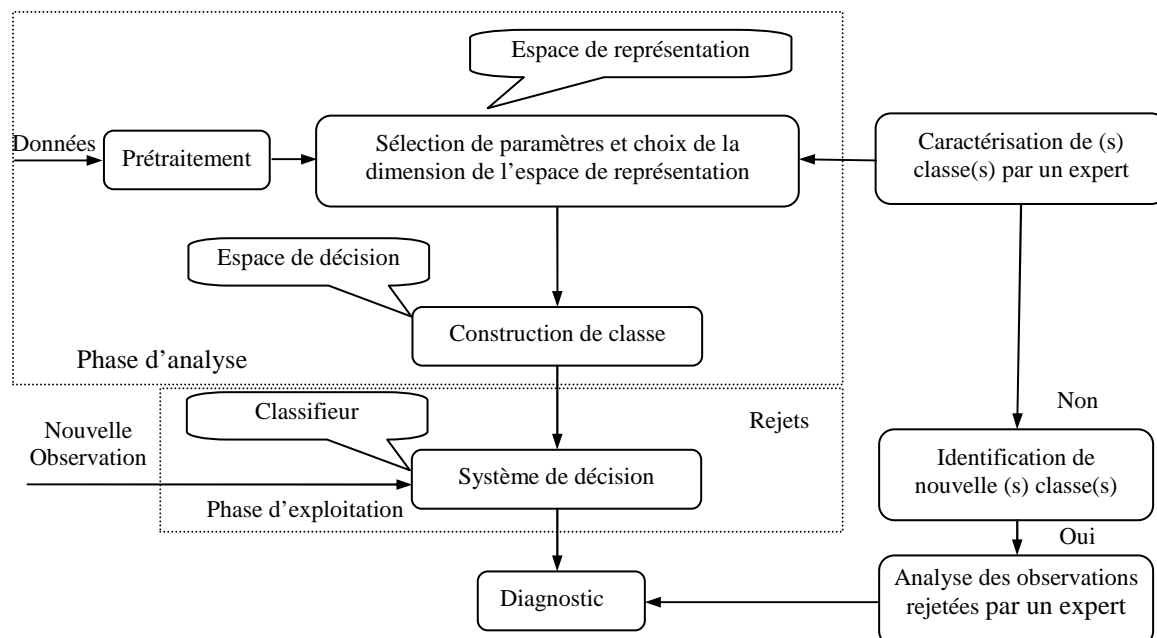


Fig. 1.5: Diagnostic par reconnaissance de formes [Mokhtari, 2007]

Aussi, on peut écrire les règles de décisions pour une nouvelle observation comme suit :

- Si une observation x est identifiée alors classer x dans la classe correspondante.

- Si x est rejetée en ambiguïté alors ajouter x à la classe rejet par ambiguïté (w_0).
- Si x est rejetée en distance alors ajouter x à la classe rejet de distance (w_1).

Il existe deux types d'approche pour l'implémentation de procédures de décision [Lebart et al., 2000]. Le premier consiste à utiliser des méthodes statistiques, paramétriques ou non paramétriques. Le deuxième type, analytique, consiste à privilégier le calcul des frontières de décision entre classes.

4.5.1 Méthodes paramétriques :

Dans les méthodes paramétriques, la loi de probabilité qui organise la distribution des observations dans les classes a une forme paramétrique connue et le problème devient alors celui de l'estimation d'un vecteur paramètre à partir d'observations issues de chaque densité $f(x/w_i)$ qui représente la loi de probabilité suivie par un vecteur x dans une classe w_i . La probabilité a priori $Pr(w_i)$ de chaque classe w_i est connue et vérifiant la condition d'orthogonalité :

$$\sum_{i=1}^m P(C_i) = 1 \quad (1.1)$$

Parmi ces méthodes, la méthode bayésienne est la plus courante [Lewis et Ringuette, 1994].

4.5.1.1 La méthode bayésienne:

Elle utilise la formule de Bayes pour calculer la probabilité à posteriori. Elle définit qu'un vecteur x appartient à une classe w_i , comme suit :

$$Pr(w_i|x) = \frac{f(x|w_i).Pr(w_i)}{\sum_{j=1}^m f(x|w_j).Pr(w_j)} \quad (1.2)$$

La règle de décision de Bayes, concernant la classe w_i d'un vecteur x , s'écrit comme suit :

$$x \rightarrow w_i \text{ si } Pr(w_i|x) = \max_{j=1,m} [Pr(w_j|x)] \quad (1.3)$$

La règle de décision de Bayes peut inclure le rejet en ambiguïté et cela en utilisant une constante C . Donc la formule précédente devient :

$$x \rightarrow w_i \text{ si } Pr(w_i|x) = \max_{j=1,m} [Pr(w_j|x)] \geq 1 - C$$

$$x \rightarrow w_0 \text{ si } \max_{j=1,m} [Pr(w_j|x)] < 1 - C \quad (1.4)$$

Sachant que le rejet en ambiguïté est possible pour :

$$0 \leq C \leq 1 - \frac{1}{m}$$

La règle de décision de Bayes peut inclure aussi le rejet de distance en utilisant la notion de densité de mélange notée $f(x)$. La fonction $f(x)$ doit être inférieure à un seuil de densité C_1 . Aussi 1.4 devient :

$$x \rightarrow w_1 \text{ si } f(x) = \sum_{j=1}^m \Pr(w_j) f(x|w_j) < C_1 \quad (1.5)$$

4.5.2 Méthodes non paramétriques :

Dans le cas où la densité $f(x/w_i)$ est inconnue on utilise les méthodes non paramétriques [Ondel, 2006]. pour l'estimer. Il existe plusieurs méthodes d'estimation non paramétriques. Dans cette section, nous allons présenter la méthode des noyaux de Parzen [Parzen, 1962] et la méthode des k plus proches voisins (k-ppv) [Fukunaga, 1990] [Ondel, 2006].

4.5.2.1 La méthode des noyaux de Parzen :

Parzen et Rozenblatt [Haykin, 1999] ont introduit la notion de fonction noyau ϕ pour pouvoir estimer la fonction de densité $f(x/w_i)$ en se basant sur un ensemble d'apprentissage (x_1, \dots, x_n) . La seule contrainte est que la fonction noyau doit être positive et normalisée.

L'estimation de la densité en un point x consiste à calculer la proportion d'observations (x_1, \dots, x_n) qui se trouvent à proximité de x . Pour cela, on construit un volume $V(x)$ qui est définie par une distance. Cette distance dépend de la répartition des points d'apprentissage, et ayant comme centre le point x . On remplace ensuite le volume $V(x)$ par une courbe en cloche centrée en x . Plus un point est proche du point de support x plus la courbe en cloche lui donnera une grande valeur. A l'inverse, les points trop éloignés de x se voient affecter une petite valeur. L'estimateur est formé par la somme des courbes en cloche. La figure 1.6. représente une estimation par la méthode du noyau d'un échantillon de 60 nombres aléatoires distribués selon la loi normale pour différentes valeurs de la fenêtre.

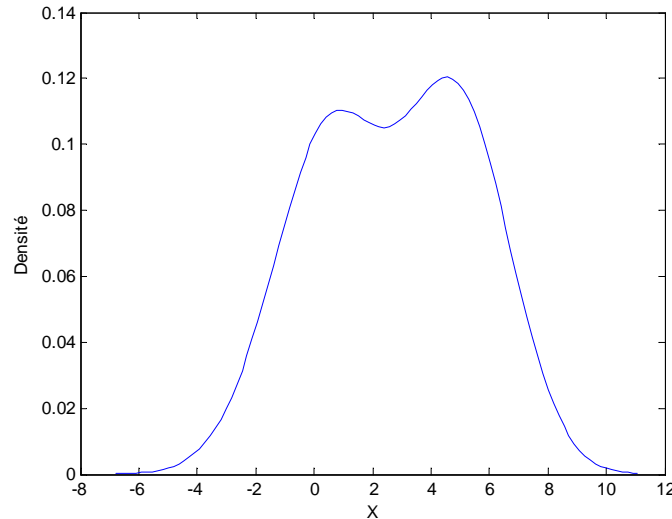


Fig. 1.6 : La méthode d'estimation par noyau

4.5.2.2 La méthode des k plus proches voisins

La méthode des k plus proches voisins (k -ppv) est la plus courante et la plus simple parmi les méthodes non paramétriques. Elle a été introduite par Fix et Hodges [Fukunaga, 1990]. K -ppv est une méthode de classification supervisée. Elle utilise un ensemble d'apprentissage (x_1, \dots, x_n) composé de n observations dont chacune est étiquetée à l'une des m classes connues. Pour identifier une nouvelle observation x à une classe, k -ppv calcule la distance entre x et les n observations de l'ensemble d'apprentissage et cherche ses k plus proches voisins. Donc cette nouvelle observation, x est affectée à la classe qui contient ses k plus proches voisins.

Pour calculer la distance entre deux vecteurs x et y , la formule générale est définie comme suit : $d(x, y) = \|x - y\|$

La distance la plus populaire est la distance euclidienne : $d(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$

Le choix de nombre de voisins k est un problème délicat. Prendre un grand nombre de voisins permet de réduire le bruit (les observations qui ne reflètent pas la réalité) puisque une observation aberrante sélectionnée comme plus proche voisin aura une influence tempérée par d'autres observations. En revanche, si le nombre de voisins est réduit cela conduit à une plus grande sensibilité aux spécificités de chaque classe puisque on prend moins en compte la moyenne des observations d'une classe. Un bon compromis pour éviter le problème d'ambiguïté d'indécision est de choisir $k = \sqrt{n}$.

Malgré que la méthode k -ppv ne nécessite pas un temps pour le calcul de paramètres, elle présente l'inconvénient de long temps pour trouver les k plus proches voisins. Pour diminuer

le temps de calcul, on peut utiliser un espace de dimension plus petit ou bien structurer les données sous forme un arbre qui accélère la recherche et évidemment avant de commencer l'exécution de procédure de recherche, il faut analyser l'ensemble d'apprentissage pour éliminer les éléments redondants.

4.5.3 Calcul direct des frontières :

L'objectif de la méthode de calcul direct des frontières est de définir des formules mathématiques qui permettent une meilleure séparation de classes. Le choix de la frontière la plus appropriée dépend de la complexité de la frontière de décision, autrement dit de la bonne ou mauvaise séparation des classes de l'ensemble d'apprentissage. L'avantage principal de cette approche est qu'elle utilise seulement des équations dans la phase de décision (et non pas l'ensemble d'apprentissage). Mais en réalité, on ne peut pas toujours séparer les classes linéairement à cause de formes de classes qui ne sont pas linéairement séparables. Alors des fonctions mathématiques plus complexes sont utilisées pour résoudre ce problème. Parfois, on doit ignorer des données de l'ensemble d'apprentissage pour pouvoir définir l'équation des frontières [Casimir, 2003].

où la fonction de discrimination linéaire a la forme suivante :

$$g(x) = f(w^t x + w_0) = f(\sum_{i=1}^d w_i x_i + w_0) \quad (1.6)$$

où w est vecteur de poids, w_0 est le biais, et f est une fonction qui convertit le résultat dans la sortie désirée.

On utilise un hyperplan pour définir les frontières de séparation linéaire entre deux classes w_i et w_j . Cet hyperplan a pour équation :

$$(w_i^t - w_j^t)x + (w_{i0} - w_{j0}) = 0 \quad (1.7)$$

que l'on peut écrire sous la forme suivante :

$$w_{ij}^t x + w_{ijk} = 0 \quad (1.8)$$

Pour définir l'équation d'un hyperplan qui sépare deux classes w_i et w_j , il faut trouver un vecteur de poids et de fixer un seuil. Il existe deux types de méthode pour calculer les paramètres du vecteur poids. Le premier type se base sur la probabilité conditionnelle. Un exemple de méthode de ce type est le discriminateur linéaire de Fisher [Duda, 2001]. Le deuxième type se base sur l'analyse par modèle discriminant [Duda, 2001]. On cite comme

exemple de méthode de ce type, la machine à vecteurs de support (SVM) [Youn, 2010]. Cette dernière sera présentée en détails dans le troisième chapitre.

4.5.3.1 Discriminateur linéaire de Fisher

Le discriminateur linéaire de Fisher [Youn, 2010].est basé sur la maximisation du critère de Fisher. Il permet d'évaluer la pertinence de chaque paramètre et calcule la qualité de classification effectuée en utilisant un seul paramètre. La formule de ce critère pour le cas de deux classes est comme suit :

$$FH(a) = \sum_{c=1}^M \sum_{\substack{r=1 \\ r \neq c}}^{M-1} \frac{m_c(a) - m_r(a)}{N_c \sigma_c^2(a) - N_r \sigma_r^2(a)} \quad (1.9)$$

Sachant que :

M représente le nombre de classes.

$m_c(a)$ représente le centre de gravité de la classe c en ne considérant que le paramètre a . Il est calculé comme suit :

$$m_c(a) = \frac{1}{N_c} \sum_{v=1}^{N_c} X_{cv}(a) \quad (1.10)$$

où X_{cv} est le vecteur v de la classe c . La valeur de N égale au nombre de vecteurs de la même classe.

$\sigma_c^2(a)$ est la variance de paramètre a des vecteurs de la classe c :

$$\sigma_c^2(a) = \frac{1}{N} \sum_{v=1}^N [X_{cv}(a) - m_c(a)]^2 \quad (1.11)$$

Le critère de Fisher prend en considération la variance intraclasse et la variance interclasses [Duda, 2001]. L'objectif de discriminateur linéaire de Fisher est de trouver un vecteur v pour lequel le critère suivant soit maximal :

$$j(X) = \frac{X^t \Sigma_B X}{X^t \Sigma_W X} \quad (1.12)$$

La valeur optimale pour ce critère est obtenue en annulant son gradient. Donc, la valeur de X est obtenue par :

$$X = \Sigma_W^{-1} (m_1 - m_2) \quad (1.13)$$

Sachant que la matrice de variance intraclasse est calculée comme suit :

$$\Sigma_W = \frac{1}{N} \sum_{c=1}^M \sum_{v=1}^{N_c} (X_{cv} - m_c)(X_{cv} - m_c)^t \quad (1.14)$$

Alors que la matrice de la variance interclasses représente la séparation entre les différentes classes est calculée comme suit :

$$\Sigma_B = \frac{1}{N} \sum_{c=1}^M (m_c - m)(m_c - m)^t \quad (1.15)$$

Avec :

- m : centre de gravité général
- M : nombre de classes
- m_c : centre de gravité de la classe numéro c
- X_{cv} : le $V^{\text{ième}}$ vecteur de la classe numéro c
- N_c : nombre de vecteurs de la classe numéro c
- N : nombre total de vecteurs

L'inconvénient majeur de ce type d'approche est que les équations de frontières dépendent de l'ensemble d'apprentissage [Casimir, 2003]. Cela signifie, si cet ensemble ne contient pas tous les cas possibles alors les équations ne permettent pas d'identifier une nouvelle observation. De plus, il faut faire la mise à jour les coefficients des équations pour quelle soit capable de reconnaître cette nouvelle observation.

4.6 Phase d'exploitation

4.6.1 Les critères d'évaluation

Pour évaluer les performances du système de diagnostic développé, il faut mesurer la qualité du classifieur employé en utilisant un critère d'évaluation. Un autre objectif d'utiliser un critère d'évaluation est d'ajuster les paramètres de classifieur [Kadri et al., 2012b]. Pour réaliser cette évaluation, nous utilisons un autre ensemble d'observations appelé ensemble de test. Nous connaissons à l'avance la classe de chaque observation de l'ensemble de test. Dans la littérature, il existe plusieurs critères d'évaluation. Ces critères reposent sur une table de contingence [Rezki et al., 2011] [Azzag et al., 2004]. Nous utilisons une table d'une classification binaire pour illustrer le calcul des différents critères qui vont suivre :

Tableau 1.2 : La table de contingence

	Observations appartenant à la classe	Observations n'appartenant pas à la classe
Observations assignées à la classe par le classifieur	a	b
Observations rejetés de la classe par le classifieur	c	d

On définit à partir des statistiques de cette table les critères suivants :

- Précision = $a / (a + b)$;
- Rappel = $a / (a + c)$;
- Exactitude = $(a + d) / (a + b + c + d)$;
- Erreur = $(b + c) / (a + b + c + d)$.

La *précision* représente le nombre d'assignations correctes sur le nombre total d'assignations, tandis que le *rappel* représente le nombre d'assignations correctes sur le nombre d'assignations qui auraient dû être faites. *L'exactitude* et *l'erreur* sont les mesures les plus utilisées en reconnaissance de formes. Elles se basent dans leurs calculs sur le nombre total d'observations. Ces deux dernières mesures ne sont pas adaptées aux problèmes où le classifieur n'arrive pas à classer les données et donc il envoie un grand nombre d'observations à la classe de rejet. Un tel classifieur retourne un taux d'erreur faible et une bonne exactitude. C'est pourquoi, il n'est pas intéressant d'utiliser ces deux mesures dans la phase d'apprentissage. Pour calculer ces mesures dans le cas de multi-classes, on cherche la mesure pour chaque classe, ensuite on calcule la moyenne pour trouver la mesure globale [Azzag et al., 2004].

4.6.2 Les méthodes d'évaluation

Ces critères d'évaluation sont exploités pour estimer l'erreur de classement en utilisant l'une des méthodes suivantes : la re-substitution, la méthode Holdout, la D-validation croisée, méthode du Leave-one-out et la méthode de ré-échantillonnage [Kalakech, 2011].

La méthode de re-substitution est une méthode facile et rapide à mettre en œuvre. Dans cette méthode, on utilise le même ensemble pour l'apprentissage et pour le test.

Par contre dans la méthode Holdout, ces deux ensembles ne sont pas identiques. Ils sont construits aléatoirement. La méthode de D-validation croisée consiste à utiliser D ensembles de même taille. On utilise $(D-1)$ ensembles pour l'apprentissage et on utilise le dernier ensemble pour faire les tests. La méthode du Leave-one-out est un cas particulier de la

méthode précédente pour lequel $D = N$. Sachant qu'ici, N représente le nombre total d'observations.

La méthode de ré-échantillonnage utilise pour l'apprentissage un ensemble de taille N égale au nombre total d'observations. L'ensemble d'apprentissage peut contenir le même élément plusieurs fois. Pour le test, on utilise toutes les observations [Kalakech, 2011].

5 Outils d'intelligence artificielle pour le diagnostic industriel

Aujourd'hui, les techniques classiques sont peu utilisées dans le domaine de diagnostic industriel. Les chercheurs sont orientés vers l'utilisation des techniques d'intelligence artificielle [Kadri et al., 2012b]. Ces techniques peuvent être appliquées sur différents types de système industriel (simple, complexe ou dynamique). Elles présentent une grande capacité d'apprentissage même avec un ensemble restreint de données. Parmi ces méthodes, on peut citer les réseaux neurones et la logique floue [Mahdaoui et Mouss, 2008].

5.1 Réseaux de Neurones

L'idée des réseaux de neurones vient originalement de la modélisation biologique du neurone [Mahdaoui et Mouss, 2008]. Ils utilisent des neurones formels basés sur une représentation mathématique. Ils montrent de grandes capacités à résoudre les problèmes de classification pour le diagnostic de systèmes complexes. Ils présentent une grande capacité d'apprentissage et de généralisation de leurs connaissances à des observations inconnues. En pratique, les réseaux de neurones peuvent être utilisés en tant que règle de décision dans un processus d'automatisation de l'opération du diagnostic.

Les réseaux de neurones sont structurés en plusieurs couches interconnectés (une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie). La couche d'entrée correspond aux observations. Donc, le nombre de neurones de cette couche est égale au nombre de paramètres du vecteur forme. Les couches cachées comprennent les prototypes des classes. Le nombre de neurones de la couche de sortie correspond au nombre de classes connues. Un résultat de cette couche représente le degré d'appartenance d'une observation à une classe.

La figure 1.7 illustre un exemple d'un réseau de neurones de type RBF (*Radial Basis Function*).

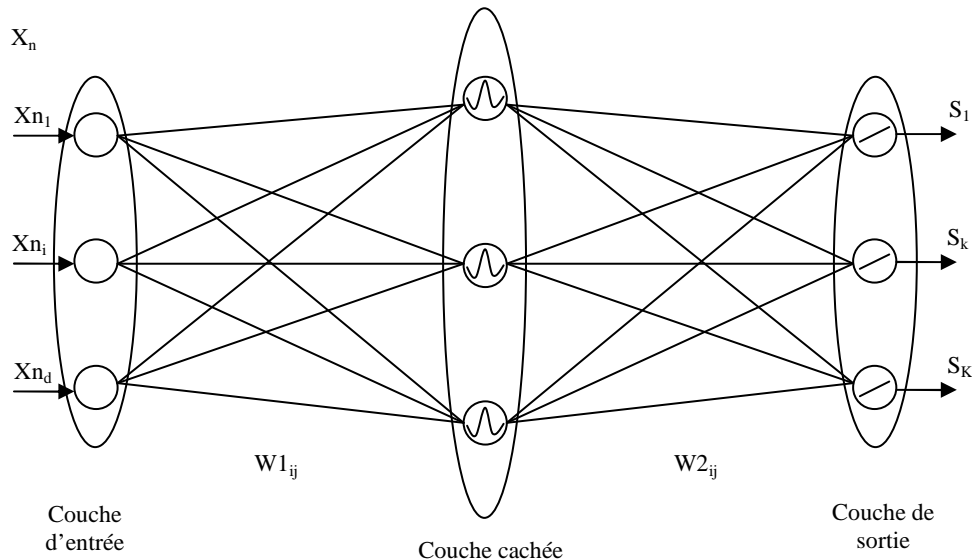


Fig. 1.7 : Structure du réseau de type RBF pour le diagnostic

L'équation générale d'un neurone est défini par :

$$y_i(t) = \sum_{j=1}^N w_{ij}x_j(t) \quad (1.15)$$

$$u_i(t) = f(y_i(t)) \quad (1.16)$$

Sachant que :

$y_i(t)$: est l'état interne du neurone i ;

$u_i(t)$: est l'état de sortie ou l'activation de sortie du neurone i ;

w_{ij} : sont les poids d'interconnexions entre les neurones.

Pour chaque nouvelle observation présentée à l'entrée d'un réseau de neurones, les degrés d'appartenance de cette observation sont évalués par rapport aux différents prototypes des classes. Dans la phase d'exploitation, le réseau est capable de classer des nouvelles observations. Les observations non classées sont rejetées par ambiguïté.

Dans plusieurs applications de diagnostic industriel [Mahdaoui et Mouss, 2008], l'apprentissage peut être long, difficile et gourmand en temps de calcul. Par ailleurs, il doit être réalisé sur toutes les données à la fois, avec le risque que le réseau ne conserve pas les expériences précédentes. Il est nécessaire de posséder une variété de données qui reflètent toutes les conditions de fonctionnement et de dysfonctionnement du système étudié. Les réseaux de neurones ne fonctionneront pas forcément correctement hors l'espace de données

d'apprentissage. Une fois entraîné, un réseau de neurone n'est pas flexible. Si de nouvelles données interviennent, il faut refaire un nouvel apprentissage.

5.2 La logique floue

La logique floue est l'une des techniques d'intelligence artificielle la plus connue. Elle a été proposée par Lotfi Zadeh en 1965. Elle est utilisée dans plusieurs domaines de recherche et d'application [Mahdaoui et Mouss, 2008], on peut citer : l'automatisme, la robotique, la gestion financière, la médecine, le diagnostic et bien d'autres.

Dans la logique classique, une variable, qui désigne l'appartenance ou non d'un élément à un ensemble, ne peut pas prendre que la valeur 0 ou 1. Cette logique ne traite pas les cas où on manipule des données vagues, imprécises, contradictoires ou dans le cas où les frontières entre les différentes classes sont mal définies. Par contre, dans la logique floue, les variables sont de type inexact et prennent des valeurs réelles entre 0 et 1.

Dans un système de diagnostic, on modélise les connaissances d'un expert sur le système étudié sous forme d'un ensemble de règles [Mouss, 2006]. Donc la construction d'un système de diagnostic flou se compose de deux phases : l'acquisition des données et la conception des règles. Les règles expriment la relation entre les symptômes et les causes. Elles permettent de localiser l'élément défaillant. En général, une règle possède la forme suivante :

SI condition ALORS conclusion

La condition comporte tous les symptômes observés par l'expert et la conclusion représente l'élément défaillant indiqué par l'expert. Alors, la classification des données par cette technique est réalisée par l'identification des propriétés communes qui caractérisent l'ensemble d'éléments de même classe. Le fonctionnement d'un système flou se décompose en Fuzzification, Inférence et Défuzzification [Mahdaoui et Mouss, 2008].

La phase de Fuzzification consiste à déterminer le degré d'appartenance de chaque observation à un état de fonctionnement ou dysfonctionnement. Cette phase est réalisée en utilisant des fonctions d'appartenances qui consistent de trois à sept états flous.

Les degrés d'appartenance permettent d'appliquer les règles floues. Donc, La logique floue propose des formalismes rigoureux permettant d'inférer le degré d'appartenance des variables de sortie.

La phase de Défuzzification permet de trouver à partir des données floues des données précises. Cette phase est réalisée soit par la sélection de grand degré d'appartenance, soit par

le calcul de la moyenne pondérée ou bien par la détermination de centre de gravité des valeurs obtenues.

La logique floue permet de donner une vision globale et plus complète sur l'état courant du système complexe en indiquant le degré d'appartenance. Elle aide par exemple l'expert à prendre des décisions pour prévenir un basculement vers un état de défaillance dans le cas où le système fonctionne normalement, mais présente un degré d'appartenance élevé à un état de défaillance. L'inconvénient majeur d'utiliser des règles floues est la génération d'un grand nombre d'erreurs. Donc pour mettre en place un système de diagnostic flou, il faut refaire la phase de test plusieurs fois pour réduire le taux d'erreurs [Meunier et *al.*, 1996].

6 Conclusion :

L'objectif de ce chapitre est de définir quelques concepts que nous allons utiliser dans la suite de cette thèse. Dans un premier temps, nous avons présenté l'architecture générale d'un système de diagnostic basé sur la RdF. Ensuite, quelques techniques utilisées dans le domaine de diagnostic industriel ont été présentées et discutées. Il s'agissait de : la méthode bayésienne, la méthode des noyaux de Perzen, la méthode des kppv, la méthode de calcul direct des frontières, discriminateur linéaire de Fisher, RdN et la LF. Le choix d'une méthode de reconnaissance des formes se fait après une évaluation de classifieur et un traitement de données et de nature du système.

Finalement, on peut dire que les outils de l'IA sont plus adaptés aux systèmes complexes et dynamiques puisque dans ce type de système, les informations recueillies sont à la fois volumineuses, imprécises et incertaines.

Dans le prochain chapitre, nous allons compléter l'étude des techniques intelligentes en présentant les algorithmes de colonie de fourmis inspirés du comportement des fourmis réelles. Une colonie de fourmis est capable de résoudre des problèmes complexes malgré que chaque fourmi de la colonie possède des capacités cognitives limitées. Ce comportement est appelé intelligence collective.

Nous allons expliquer par la suite l'utilisation de ces algorithmes dans chaque phase de diagnostic en tant que méthode d'optimisation ou en tant que méthode de classification.

LES COLONIES DE FOURMIS

Résumé :

Les algorithmes de colonies de fourmis ont été initialement introduits par Dorigo en 1992 dans sa thèse de doctorat. L'objectif de ce chapitre est d'introduire les trois algorithmes de colonies de fourmis que nous avons utilisés dans le cadre de cette thèse. Dans la première partie de ce chapitre, nous allons présenter une étude bibliographique sur les algorithmes de colonie de fourmis. Dans la deuxième partie, nous proposons des nouvelles implémentations de quelques algorithmes que nous avons utilisés pour le développement d'un système de diagnostic industriel.

1 Introduction :

Au cours de ces dernières années, plusieurs méta-heuristiques ont montré leur efficacité pour la résolution des problèmes complexes, l'ensemble auquel appartient le problème de diagnostic industriel. Les insectes sociaux comme les fourmis sont réputés pour leur capacité à réaliser des tâches collectives relativement complexes. On cite par exemple les constructions de nids, le tri du couvain ou l'auto-répartition des tâches dans la population [Dorigo, 1992]. Ces réalisations sont généralement considérées comme le résultat d'un travail collectif puisque les individus sont incapables d'appréhender les structures en question et pourtant la communauté parvient à les parachever. Cette capacité est nommée intelligence collective.

2 Les fourmis réelles

2.1 Principe de base

Dans un premier temps, les fourmis marchent aléatoirement de nid à la source de nourriture et vice-versa. Chaque fourmi dépose sur son chemin une substance chimique appelée phéromone qui la guide au retour.

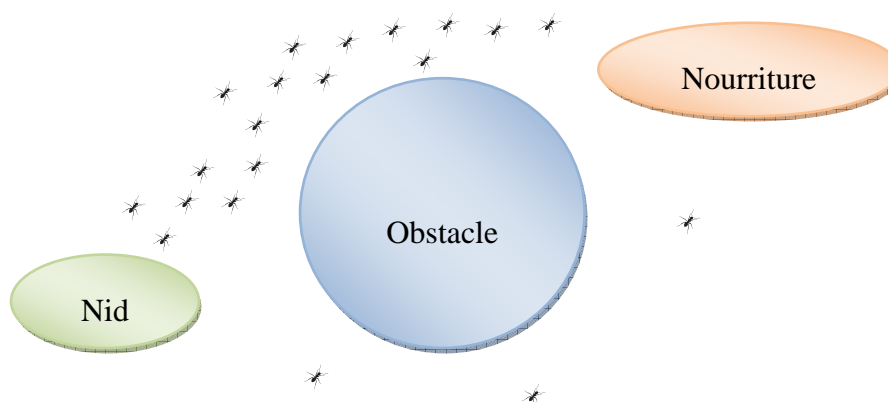


Fig. 2.1: La colonie choisit le chemin le plus court

Dans ce dernier cas, la nature de phéromone est différente puisqu'elle contient une information sur la qualité de la source de nourriture. Par la suite, la phéromone incite les autres fourmis de choisir ce chemin. Les fourmis ont aptitude de suivre le chemin qui contient la plus grande quantité de phéromones. Autrement dit, lorsqu'une fourmi se trouve devant plusieurs chemins, elle peut trouver le chemin le plus court grâce à la densité de phéromone. Ce comportement permet à la colonie de trouver le plus court chemin entre le nid et la source de nourriture (figure 2.1) [Colorni et al., 1991].

2.1.1 La résolution de problèmes complexes

La coopération entre les membres de la fourmilière permet la réalisation d'activités complexes, c'est ce que nous allons expliquer à l'aide de l'exemple du double pont binaire (figure 2.2) [Deneubourg et al., 1990]:

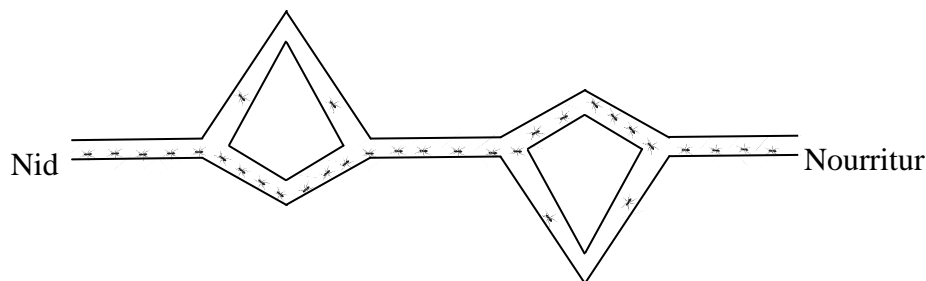


Fig. 2.2 : L'expérience du double pont

Initialement, les ponts sont vides et ne contiennent pas de phéromone. Chaque chemin possède la même probabilité d'être choisi. Les fourmis vont choisir aléatoirement un chemin parmi les chemins possibles. Le chemin le plus court sera marqué rapidement par plus de phéromone puisque les fourmis qui prennent ce chemin déposent plus de phéromone pour une même période que les autres fourmis qui choisissent un autre chemin.

2.1.2 Stigmergie

La stigmergie est une méthode de communication indirecte dans un environnement émergent auto-organisé, où les individus communiquent entre eux en modifiant leur environnement. Cette communication indirecte engendre un comportement complexe de l'ensemble à partir de comportements individuels simples.

3 Les fourmis artificielles

En se basant sur le comportement social des fourmis présenté dans la section précédente, une variété d'algorithmes d'optimisation, appelée OCF ("Optimisation par Colonie de Fourmis") ont été développés. Ces algorithmes sont relativement simples et efficaces. Chaque fourmi comporte un ensemble de fonctions qui définissent son comportement. Il est très similaire à celui de la fourmi réelle lorsqu'elle recherche de la nourriture (figure 2.2). La fourmi artificielle se déplace dans son environnement qui est un espace combinatoire. Cet espace est composé d'un ensemble d'objets obtenus par une modélisation du système étudié. Les déplacements de chaque fourmi représentent une solution au problème posé. Les solutions

trouvées sont évaluées en utilisant une fonction qui détermine la meilleure solution [Ouadfel et Batouche, 2007].

4 Les systèmes multi-agents

4.1 Principes de bases

Le thème des systèmes multi agents (SMA) est un champ de recherche très actif. Actuellement, cette discipline est à la connexion de plusieurs domaines en particulier : l'intelligence artificielle, les systèmes informatiques distribués et de génie logiciel. Elle s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents.

4.2 Les agents

Selon Ferber, un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents [Ferber, 1995]. A partir de cette définition, on définit un agent logiciel comme une entité informatique qui possède les caractéristiques suivantes [Deneubourg et al., 1990]:

1. Son environnement est un système informatique ouvert qui peut contenir un ensemble d'applications, de réseaux et de systèmes hétérogènes;
2. Possède des moyens qui lui permet de communiquer avec d'autres agents ;
3. Est mue par un ensemble d'objectifs propres ;
4. Possède des ressources propres ;
5. Ne dispose que d'une représentation partielle des autres agents ;
6. Possède des compétences (services) qu'elle peut offrir aux autres agents ;

A partir des ces caractéristiques, on peut dire que chaque fourmi virtuelle représente un agent capable d'agir et de communiquer. La fourmi virtuelle modifie les valeurs de phéromone associées aux différents éléments et elle utilise la phéromone comme medium de communication indirecte. Donc, le résultat de son comportement est le changement de son environnement et par conséquent, le choix des autres fourmis sera changé à l'itération suivante.

On peut aussi dire, que chaque fourmi virtuelle représente un agent doué d'autonomie. Le but d'une fourmi virtuelle est de trouver une solution au problème traité. Elle applique un ensemble de règles et utilise la phéromone et des valeurs heuristiques pour réaliser son objectif. Comme les agents, les fourmis virtuelles ne possèdent qu'une perception partielle de l'environnement et n'ont pas conscience de la totalité des éléments qui influencent la colonie. On peut classer les fourmis virtuelles comme de agents réactifs car elles ne peuvent pas acquérir de nouvelles connaissances et se contentent d'agir/réagir suivant les informations qu'elles trouvent dans leur environnement.

Pour terminer cette petite présentation d'agent, il est nécessaire de définir ce que l'on appelle un système multi-agents (SMA). Un SMA est un système qui comprend un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. D'après Ferber [Ferber, 1995], un SMA est composé généralement des éléments suivants :

- Un espace combinatoire E qui représente son environnement.
- Un ensemble d'objets O . dont chaque objet possède une position dans l'environnement E . Ces objets représentent des ressources pour les agents. Donc, les agents peuvent manipuler ces objets pour réaliser leurs objectifs.
- Un ensemble A d'agents, qui représentent les entités actives du système.
- Un ensemble R de relations qui peuvent exister entre les agents.
- Un ensemble d'opérateurs Op . permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on pourrait appeler « les lois de l'univers ».

4.3 Architecture des agents

Les architectures d'agents abordent les questions de concevoir et de créer des systèmes informatisés qui satisfont les propriétés de l'agent telles que : l'autonomie, la réactivité, la proactivité et les capacités sociales.

Wooldridge et Jennings [Wooldridge et Jennings, 1995] indiquent que les architectures d'agent peuvent être vues comme des modèles de technologie de la programmation des agents. Ils identifient les classes suivantes des architectures de l'agent.

- Les architectures cognitives (ou délibératives).
- Les architectures réactives (ou comportementales).

- Les architectures hybrides.

4.4 Les architectures réactives :

Les agents réactifs sont souvent qualifiés de ne pas être intelligents par eux-mêmes. Ils sont des composantes très simples qui perçoivent l'environnement et sont capables d'agir sur celui-ci. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances, pas de mécanismes d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action (figure 2.3) [Ferber, 1995].

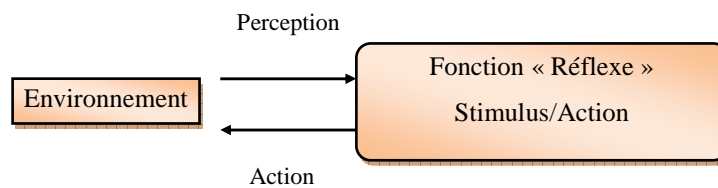


Fig. 2.3: Modèle d'agent réactif

Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent. Les autres architectures ne seront pas abordées parce qu'elles sortent du cadre de cette thèse. Nous invitons le lecteur intéressé à consulter le livre de Ferber [Ferber, 1995].

5 L'intelligence collective des fourmis

Les fourmis sont présentes dans la quasi-totalité des écosystèmes terrestres. Il existe plusieurs races différentes de fourmis qui présentent une grande diversité de comportements et de morphologies. Pour des raisons pratiques, les recherches du domaine d'intelligence collective de fourmis s'intéressent uniquement à un nombre limité de races.

Le comportement collectif des fourmis constitue une source d'inspiration pour les chercheurs. La coopération, la communication, la compétition et l'apprentissage de fourmis ont permis la conception de plusieurs algorithmes pour la résolution de divers problèmes [Oquadfel et Batouche, 2007]. Dans ce qui suit, nous allons présenter les principales caractéristiques des fourmis que l'on pourra modéliser et l'exploiter dans des systèmes informatiques.

5.1 Le partage de tâches

L'une des caractéristiques les plus étonnantes est l'aptitude de la colonie de fourmis à se partager le travail. Les fourmis assurent d'une façon parallèle un ensemble de tâches importantes pour la survie et le développement de la colonie. On peut citer entre autres : la

recherche de nourriture, la défense du nid, l'entretien et la construction du nid et l'entretien des larves et leur approvisionnement en nourriture [Monmarché, 1999].

5.2 Auto-organisation

Un autre comportement observé chez les fourmis est l'auto-organisation. Le résultat de ce comportement est la construction de structures vivantes ayant différentes fonctionnalités. Ces structures sont composées de fourmis qui sont accrochées les unes aux autres. L'auto-assemblage représente a représenté une source riche d'inspiration pour plusieurs chercheurs surtout dans le domaine des algorithmes de classification non supervisée (la section 6.1.) [Azzag et al., 2004].

5.3 La communication

La fourmi possède plusieurs moyens de communication. On peut citer entre autres [Ferber, 1995] :

5.3.1 La communication sonore

Elle est obtenue par les frottements d'un mince grattoir transversal situé sur sa taille, contre un plateau de fines crêtes parallèles, placée sur l'abdomen. Cette stridulation peut remplir différentes fonctions, selon l'espèce et les circonstances :

- Signal de détresse : Ce signal se propage dans le sol. Il est émis par l'ouvrière lorsqu'il pleut ou quand l'ouvrière est en danger. La fourmi utilise ses pattes pour la réception de ce type de message.
- Signal de qualité de l'alimentation : Ce signal est émis par l'ouvrière pour attirer les autres fourmis. L'intensité de la vibration émise indique la valeur nutritive de la nourriture.
- Comme signal de renforcement : Ce signal est émis par l'ouvrière lorsqu'elle trouve un aliment de grande taille pour obtenir l'aide des autres fourmis.

5.3.2 La communication tactile

Chez les fourmis, certains messages simples sont transmis par des contacts physiques tels que les attouchements, les tapotements ou les effleurements. La majorité de communications sont réalisées par l'intermédiaire des antennes qui sont sensibles au toucher.

5.3.3 La communication visuelle

Ce type de communication est de moins en moins utilisé chez les fourmis. Il était bien plus fréquent au stade primaire. A cause de leur évolution considérable, les fourmis utilisent d'autres moyens de communication plus efficaces. La communication visuelle est encore utilisée dans des situations particulières. Par exemple lorsqu'une fourmi est confrontée à un objet trop gros, elle tourne autour de l'objet plusieurs fois pour que les autres fourmis la voient et viennent l'aider.

5.3.4 La communication chimique

Les phéromones sont formées d'hydrocarbures cuticulaires. Les molécules des phéromones sont organiques. Ils sont constitués de carbone, hydrogène, oxygène et azote. Les phéromones ne sont pas les mêmes suivant la colonie et l'espèce. Elles représentent un véritable moyen d'identification et contiennent à la fois les informations sur l'espèce, sur le nid, sur la caste ou encore sur le stade de développement des individus. Elles permettent d'obtenir un véritable équilibre social au sein de la fourmilière grâce à des messages de communication échangés entre les membres de la colonie. Les phéromones représentent un moyen d'alerte très performant qui permet à la fourmi d'avertir ses congénères d'un danger et de préparer une attaque rapidement. Par exemple, dans le cas où un élément étranger entre dans la zone de la colonie, chaque fourmi rencontre cet élément produit une phéromone d'alarme qui sera diffusée à travers la colonie avec une vitesse propre à chaque phéromone et la nature de la phéromone définit le degré et le temps d'agressivité de fourmis.

6 Les différents domaines d'application

Le premier algorithme de colonie de fourmis a été appliqué pour la résolution de problème de recherche du plus court chemin. Les bons résultats obtenus ont favorisé les chercheurs à développer d'autres algorithmes pour la résolution de problèmes complexes comme le voyageur de commerce, l'ordonnancement séquentiel, détection de graphe hamiltonien, l'affectation quadratique, le problème de routage dans un réseau non commuté, l'optimisation numérique, etc [Monmarché, 1999].

Dans cette section, nous présentons quelques travaux concernant la résolution de problèmes d'optimisation combinatoire en utilisant les colonies de fourmis. On peut modéliser tous ces problèmes par des graphes. La construction de graphe représente le premier pas dans le processus de la programmation en utilisant les colonies de fourmis. Les graphes permettent de

modéliser divers problèmes du monde réel : réseau routier $G = (\text{villes}, \text{routes})$, site web $G = (\text{pages}, \text{liens})$, réseau internet $G = (\text{serveurs et utilisateurs}, \text{interconnexions})$, assignation des touches sur un clavier $G = (\text{touches}, \text{emplacements})$.

Le premier exemple concerne (DCMSTP) le problème de l'arbre couvrant de poids minimal (Degree Constrained Minimum Spanning Tree Problem). L'objectif de ce problème est de trouver un arbre couvrant contraint par degré avec le coût minimal (figure 2.3). DCMSTP est connu comme un problème NP-complet. Ainsi une solution de temps polynomial pour lui peut être impossible. DCMSTP a un grand nombre d'applications pratiques dans plusieurs domaines comme la conception d'ordinateur, télécommunication et des réseaux de transportation. Jusqu'à maintenant, il y avait beaucoup de méthodes proposées pour résoudre ce problème. Parmi elles, la relaxation lagrangienne est l'une des plus importantes méthodes utilisées dans réalité.

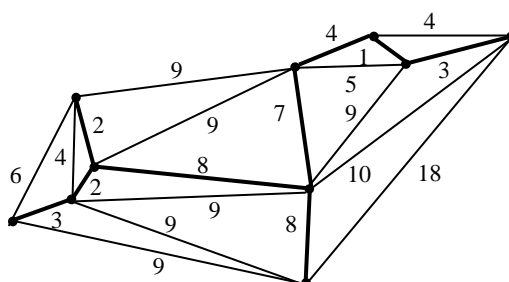


Fig. 2.4 : L'arbre couvrant de poids minimal d'un graphe planaire

Colas a appliqué les algorithmes de colonies de fourmis pour la résolution de ce problème. Elle a proposé une heuristique à base de fourmis artificielles pour ajuster les poids de graphe afin de rapprocher les requêtes de sessions différentes de navigation sur le web. A partir de ce graphe, un arbre de recouvrement minimal est calculé. La racine de cet arbre est la requête en moyenne la plus proche des autres requêtes du log [Colas, 2008].

Le deuxième exemple concerne le problème de tournées de véhicules (vehicle routing problem). Il consiste à trouver les tournées d'un ensemble de véhicules afin de livrer une liste de clients, ou de réaliser des tournées d'interventions (maintenance, réparation, contrôles) ou de visites (visites médicales, commerciales, etc.). L'objectif est de minimiser le coût de livraison des biens. Ce problème est une extension classique du problème du voyageur de commerce, et fait partie de la classe des problèmes NP-complet (la figure 2.4). Il peut se modéliser à l'aide d'un graphe de $n+1$ nœuds, où le nœud 0 correspond au dépôt et les nœuds 1 à n correspondent à des clients. S'il existe une interaction entre les véhicules, la résolution du problème précédent peut devenir très difficile. Il existe plusieurs variantes classiques du

problème de tournées de véhicules (VRP) : VRP avec contraintes de capacité où les véhicules ont une capacité d'emport limitée (quantité, taille, poids, etc.), VRP avec contraintes liées aux ressources et aux clients (disponibilité, localisation, compétences requises, etc.), VRP avec fenêtre de temps où pour chaque client on impose une fenêtre de temps dans laquelle la livraison doit être effectuée et VRP avec collecte et livraison où un certain nombre de marchandises doivent être déplacées de sites de collecte vers des sites de livraisons [Gambardella, 1999].

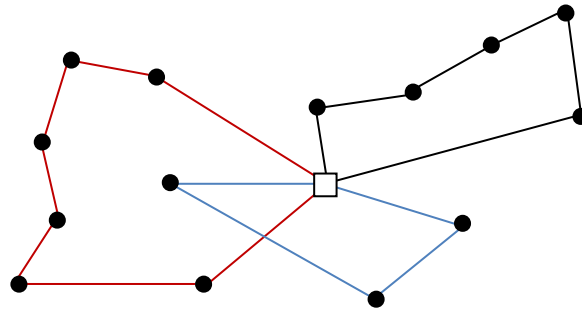


Fig. 2.5 : Un problème de tournées de véhicules avec un dépôt central

Des méthodes exactes ont été proposées pour la résolution de ce type de problèmes mais elles ne permettent de résoudre que les problèmes du VRP lorsqu'il existe moins de 60 clients. Plusieurs métaheuristiques ont été appliquées pour la résolution des problèmes de tournée de véhicules. Montamenni a appliqué pour la première fois les algorithmes de colonies de fourmis pour la résolution du VRP. Ce dernier traite une composante dynamique dont elle s'exprime par les apparitions de nouveaux clients, de nouvelles demandes, ou de pannes des véhicules de transport [Gambardella, 1999].

7 Les algorithmes de colonies de fourmis et les SMA

Dans cette section, nous proposons de nouvelles implémentations de quelques algorithmes inspirés du monde réel. Le premier modèle concerne l'algorithme AntTreeStoch basé sur l'auto-organisation observée chez les fourmis réelles.

Le deuxième algorithme est Lumer & Faieta est basé sur les principes de tri des éléments du couvain par les fourmis.

Enfin, le dernier modèle se base sur l'algorithme binaire de colonie de fourmis qui permet de trouver le plus court chemin entre la nourriture et le nid.

Nous montrons par la suite, les avantages apportés par ce nouveau paradigme de programmation qui est la programmation par les systèmes multi-agents (SMA) réactifs dans le domaine d'ACO (Ant Colony Optimization).

7.1 L'algorithme AntTreeStoch

7.1.1 Les grandes lignes

Holldöbler et Wilson en 1990 [Holldöbler et Wilson, 1990] ont montré que les fourmis disposent d'un système de reconnaissance basé sur l'odeur de chaque fourmi. Ce système permet aux fourmis qui appartiennent au même groupe de s'assembler et de rejeter les intrus. En se basant sur ces études, Labroche [Labroche et al., 2002] a développé une nouvelle méthode de classification non-supervisée appelée AntClust. Le principe de l'algorithme AntClust est basé sur un nombre limité des contacts directs entre les fourmis. Grâce à ces contacts, chaque fourmi tente de trouver un congénère qui a presque la même odeur. L'émergence de ces contacts est la répartition des fourmis réelles sur plusieurs nids. Dans le cas des fourmis artificielles, chaque nid représente une classe de données. Grâce toujours à une fonction de similarité basée sur l'odeur, les fourmis isolées sont affectées à la classe la plus similaire. AntClust a été appliqué au problème de classification des sessions de navigation des internautes pour la mesure d'audience sur le web.

En se basant sur le même principe, Azzag a proposé un nouvel algorithme [Azzag et al., 2004]. Elle a exploité un autre comportement observé chez les fourmis réelles qui est la construction des structures vivantes. La combinaison de ces deux principes a permis de proposer un algorithme de classification hiérarchique et non supervisée appelé AntTreeStoch. Elle l'a appliqué au problème de la classification hiérarchique de documents afin de générer automatiquement des sites portails pour le web.

Le principe de cet algorithme est de placer toutes les fourmis sur un support principal qui représente la racine de l'arbre. Les fourmis portent des vecteurs de données qui sont les informations à classer. Chaque fourmi peut agir de trois manières différentes :

- s'accrocher au support ;
- se déplacer sur une autre fourmi ;
- s'accrocher à une autre fourmi pour prolonger l'arbre.

Ce comportement est déterminé notamment par la similarité entre les données et la structure locale de l'arbre. Il permet de construire l'arbre d'une manière distribuée. Le résultat final des

déplacements et des accrochements des fourmis sera un arbre des données classées qu'on peut interpréter de différentes manières selon le domaine étudié.

7.1.2 L'algorithme principal d'AntTreeStoch

Pour classer les données, on utilise la fonction $\text{Sim}(i, j)$ qui calcule la similarité entre deux nœuds i et j . Chaque nœud représente une donnée à classer. La valeur de la similarité est comprise entre 0 et 1. Pour les données numériques, on calcule la similarité en utilisant la distance euclidienne et pour les données symboliques, on utilise la distance de Hamming [Kadri et al., 2010a].

Algorithme 2.1 : La construction d'un arbre par des fourmis artificielles
Initialisation des seuils de similarité et de dissimilarité des fourmis Tant qu'il existe une fourmi f_i non connectée faire Si sa position est le support alors Cas support Sinon Cas fourmi Fin Si Fin Tant que

7.1.3 Cas d'une fourmi sur le support

Pour chaque fourmi f_i qui se trouve sur la racine, si f_i est assez similaire à une fourmi f_+ qui est connectée à la racine, f_i se place sur celle-ci. Dans le cas inverse et si f_i est assez dissimilaire à f_+ et qu'il existe un lien entrant libre sur la racine, f_i se connecte à la racine.

Algorithme 2.2 : La simulation du comportement d'une fourmi f_i en déplacement sur le support f_0 .
Si aucune fourmi n'est connectée au support f_0 alors Connecter f_i à f_0 Sinon Si $\text{Sim}(f_i, f_+) \geq \text{SSim}(f_i)$ alors Déplacer f_i vers f_+ Sinon Si $\text{Sim}(f_i, f_+) < \text{SDissim}(f_i)$ alors Si place libre sur le support alors

```

    Connecter  $f_i$  au support  $f_0$ 
  Sinon
    Diminuer  $SSim(f_i)$  et déplacer  $f_i$  vers  $f_+$ 
  Finsi
  Sinon
    Diminuer  $SSim(f_i)$  et augmenter  $SDissim(f_i)$ 
  Fin Si
  Fin Si
  Fin Si

```

7.1.4 Cas d'une fourmi placé sur une autre

Lorsqu'une fourmi f_i est positionnée sur une autre fourmi alors on la notera f_{pos} . S'il y a une place sur f_{pos} et qui sont assez similaires et f_i suffisamment dissimilaire aux voisines de f_{pos} , alors f_i se connecte à f_{pos} . S'il n'y a pas une place sur f_{pos} alors f_i se positionne aléatoirement sur une des voisines de f_{pos} , dans ce cas on ne met pas à jour les seuils, ces derniers ont permis de bien localiser f_i dans l'arbre et en les modifiant on risquerait de mal classer f_i .

Algorithme 2.3 : La simulation du comportement d'une fourmi f_i en déplacement sur une autre fourmi f_{pos} connectée à la structure

```

  Si  $Sim(f_i, f_{pos}) \geq SSim(f_i)$  alors
    Si  $Sim(f_i, f_+) < SDissim(f_i)$  alors
      Si nombre de liens entrants  $< L_{max}$  sur  $f_{pos}$  alors
        Connecter  $f_i$  à  $f_{pos}$ 
      Sinon
        Déplacer  $f_i$  aléatoirement vers  $f_k$ 
      Fin Si
    Sinon
      Diminuer  $SDissim(f_i)$ , augmenter  $SSim(f_i)$  et déplacer  $f_i$  aléatoirement vers  $f_k$ 
    Fin Si
  Sinon
    Déplacer  $f_i$  aléatoirement vers  $f_k$ 
  Fin Si

```

Maintenant, si f_i est suffisamment similaire à f_{pos} mais pas suffisamment dissimilaire aux voisines de f_{pos} , on positionne f_i aléatoirement sur une des voisines de f_{pos} et on met à jour les

seuils (de la même façon que dans le cas support) pour que f_i soit plus tolérante à la prochaine itération qui la concerne. f_i se déplace dans ce cas dans l'arbre et pourra trouver un meilleur endroit pour se fixer.

S'il n'y a pas de place sur la racine, f_i est repositionnée sur f_+ en diminuant son seuil de similarité pour rendre f_i plus tolérante. Enfin, si f_i n'est ni suffisamment similaire, ni suffisamment dissimilaire, on met à jour ses seuils en diminuant le seuil de similarité et en augmentant le seuil de dissimilarité pour la rendre plus tolérante et augmenter ses chances de se connecter à la prochaine itération la concernant.

Enfin si f_i n'est pas suffisamment similaire à f_{pos} , on change de chemin en positionnant f_i aléatoirement sur une des voisines de f_{pos} . L'algorithme se termine lorsque toutes les fourmis sont connectées.

7.1.5 Critiques et limites de AntTreeStoch

Nous avons résumé les inconvénients de l'algorithme AntTreeStoch dans les points suivants [Kadri et al., 2010a] :

La position de toutes les fourmis est initialisée à 0 qui désigne la racine de l'arbre. Les décisions de fourmis dépendent de deux valeurs (seuil de similarité et de dissimilarité). Ces seuils sont initialisés respectivement à 1 et à 0. La première fourmi est connectée directement au support. Ensuite pour chaque fourmi f_i , on distingue deux cas: se placer sur une autre fourmi ou se connecter à la racine.

L'exécution séquentielle de l'algorithme influe sur la qualité de classification puisque le choix de la branche que va prendre une fourmi quelconque dépend des fourmis déjà classées (figure 2.6). Donc le tri initial représente un facteur majeur à cause de l'absence totale de parallélisme dans les déplacements des fourmis.

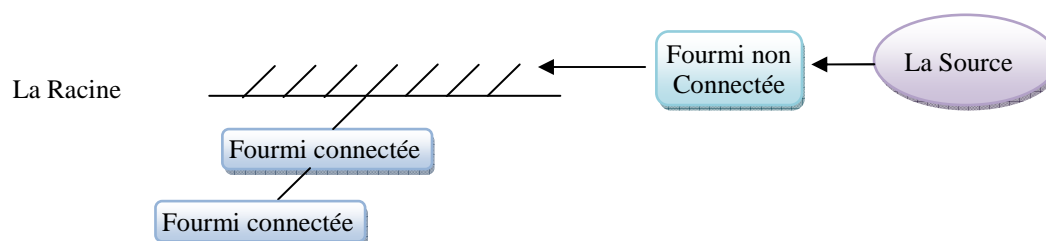


Fig. 2.6 : Une exécution séquentielle d'AntTreeStoch

Un autre point faible de AntTreeStoch est la non signification de déplacement aléatoire de fourmis dans le cas où la similarité entre la fourmi f_i et f_+ inférieur de seuil de disimilarité de f_i . Le déplacement aléatoire de f_i va l'éloigner de sa zone de recherche réelle.

La taille énorme de l'arbre engendré par l'algorithme rend son interprétation difficile voir impossible si le nombre de données d'apprentissage est énorme. Aussi la croissance de la taille de l'arbre rend la phase de classification très longue, ce qui contredit l'objectif de l'utilisation des métaheuristiques.

7.1.6 Les Améliorations

Suite aux travaux de [Azzag et al., 2004], nous proposons de porter des améliorations sur l'algorithme AntTreeStoch. Ces dernières concernent le prétraitement des données utilisées lors de la phase d'apprentissage, l'ajustement de paramètres de l'algorithme suivant la nature des données de notre système et l'optimisation de l'arbre durant la classification de nouvelles données.

Deux phases d'exécution de l'algorithme sont connues. La première est celle de la préclassification qui consiste à construire un arbre de base à partir d'un échantillonnage de données. L'ensemble de données représente toutes les situations possibles du système. Le tri des données d'entrée influe forcément sur le résultat final. Dans cette phase, le parallélisme de l'algorithme est possible mais il faut prendre en considération les cas d'accès simultanés (priorité d'accès). La deuxième phase est la classification des données réelles. Le parallélisme total n'est pas permis mais on peut entamer la classification d'une nouvelle donnée dès que la précédente a choisi une branche (une sorte de pipeline). Puisque chaque donnée est un agent qui agit d'une façon autonome [Kadri et al., 2010a].

Devant les inconvénients d'AntTreeStoch, nous avons effectué un ensemble de modifications sur l'algorithme pour l'adapter au type d'application où le volume de données est immense.

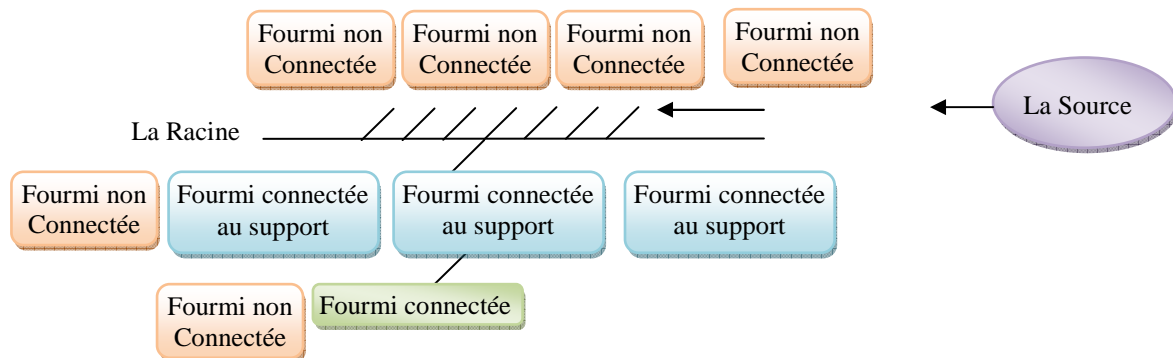


Fig. 2.7 : Une exécution parallèle d'AntTreeStoch

Faire déplacer les fourmis d'une façon parallèle va minimiser l'influence de tri initial sur la qualité de classification. Alors chaque fourmi exécute une seule action à la fois au lieu de tous les déplacements et les accrochages (figure 2.7). Ici le parallélisme offre plus de liberté aux fourmis et par conséquent les fourmis ne sont pas obligées de créer des nouveaux niveaux inférieurs.

Toujours pour minimiser la taille de l'arbre, nous avons ajouté un nouveau paramètre appelé S_{id} qui permet aux plusieurs fourmis de se connecter à la même position (figure 2.8). Le paramètre S_{id} prend une valeur proche de zéro pour ne pas influencer sur la qualité de classification.

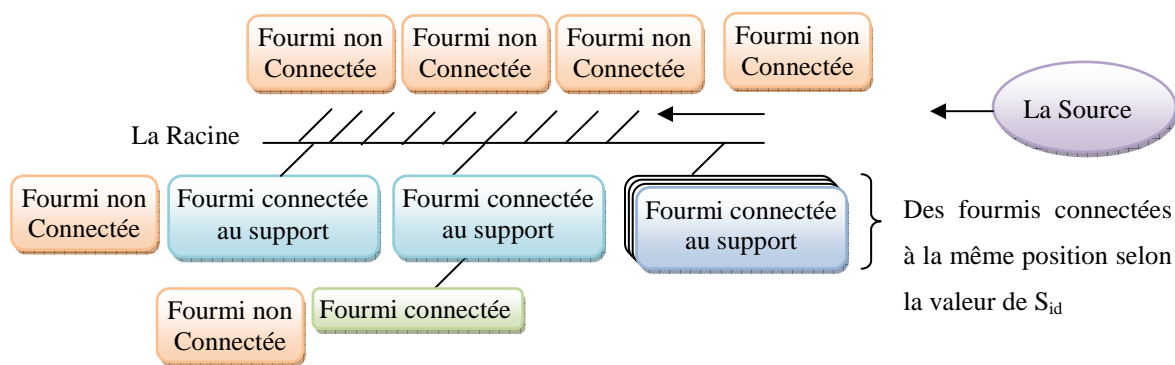


Fig. 2.8 : Une exécution parallèle d'AntTreeStoch avec le paramètre S_{id}

Pour rendre la position de chaque nœud plus significative, nous avons modifié les déplacements des fourmis en favorisant le chemin de la fourmi la plus similaire.

Nous avons utilisé un paradigme de développement à base d'agents pour implémenter l'algorithme 2.4. Ce paradigme permet de programmer des entités qui s'exécutent en parallèle. Le code de l'algorithme 2.4. représente le corps de chaque agent du système développé. Dans chaque itération de la boucle principale de l'algorithme 2.4, tous les agents non-connectés font un seul déplacement avec un ordre d'exécution aléatoire. Quelques exemples d'exécution de ce système seront présentés dans le dernier chapitre de cette thèse

Algorithme 2.4 : L'algorithme AntTreeStoch avec les améliorations proposées
Initialisation des seuils de similarité et de dissimilarité des fourmis
Tant qu'il existe une fourmi f_i non connectée faire
Si sa position est le support alors
Si aucune fourmi n'est connectée au support f_0 alors
Connecter f_i à f_0
Sinon

Si $\text{Sim}(f_i, f_+) \geq \text{SSim}(f_i)$ alors

 Déplacer f_i vers f_+

Sinon

 Si $\text{Sim}(f_i, f_+) < \text{SDissim}(f_i)$ alors

 Si $\text{Sim}(f_i, f_+) < S_{id}$ alors

 fixer f_i sur f_{pos}

 Sinon

 Si place libre sur le support alors

 Connecter f_i au support f_0

 Sinon

 Diminuer $\text{SSim}(f_i)$ et déplacer f_i vers f_+

 FinSi

 FinSi

 Sinon

 Diminuer $\text{SSim}(f_i)$ et augmenter $\text{SDissim}(f_i)$

 FinSi

 FinSi

 FinSi

Sinon

 Si $\text{Sim}(f_i, f_{pos}) \geq \text{SSim}(f_i)$ alors

 Si $\text{Sim}(f_i, f_+) < \text{SDissim}(f_i)$ alors

 Si $\text{Sim}(f_i, f_+) < S_{id}$ alors

 fixer f_i sur f_{pos}

 sinon

 Si nombre de liens entrants $< L_{max}$ sur f_{pos} alors

 Connecter f_i à f_{pos}

 FinSi

 Sinon

 Chercher f_k la fourmi la plus similaire à f_i

 Déplacer f_i vers f_k

 Fin Si

 Sinon

 Diminuer $\text{SDissim}(f_i)$, augmenter $\text{SSim}(f_i)$

```

    Chercher fk
    Déplacer fi vers fk
  Fin Si
Sinon
    Chercher fk
    Déplacer fi vers fk
  Fin Si
FinSi
FinTQ
  
```

7.2 L'algorithme de Lumer & Faieta

7.2.1 Les grandes lignes

L'algorithme proposé par Deneubourg [Deneubourg et al., 1990] a été repris et étudié par Lumer & Faieta pour la classification des données numériques. Les extensions introduites concernent en particulier les points suivants [Lumer et Faieta, 1994] :

- Les données sont représentées par des vecteurs de données numériques.
- La similarité entre deux données est mesurée comme une distance euclidienne entre leur vecteur de caractéristiques.
- La fourmi est capable de percevoir une région R_s de $s \times s$ cases autour de sa position courante sur la grille (figure 2.9).

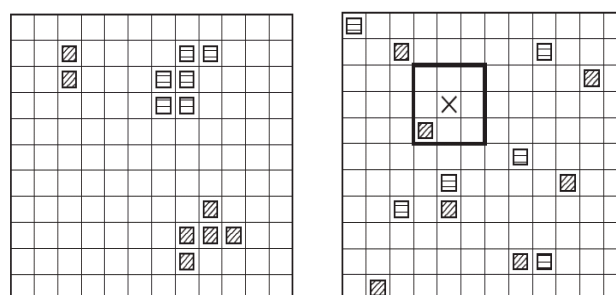


Fig. 2.9 : Résultat possible de l'exécution de l'algorithme LF sur la grille

7.2.2 La distance

Etant donné qu'un ensemble $O = \{o_1, \dots, o_n\}$ de N objets correspondant chacun à un point d'un espace métrique à M dimensions dont les coordonnées sont notées par le vecteur $x_i = \{x_{i1}, \dots, x_{iM}\}$. Pour tout objet o_i , on peut définir une mesure de distance.

Rappelons qu'une mesure de distance $d(\cdot)$ doit vérifier les propriétés suivantes :

- $d(x, y) = d(y, x)$
- $d(x, y) \geq 0$
- $d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) \leq d(x, z) + d(z, y)$ où x, y et z sont des vecteurs de données.

Etant donné qu'une distance est assimilable à une mesure de disimilarité. L'approche la plus simple pour trouver la disimilarité entre deux ensembles de données est de ranger chaque ensemble dans un vecteur et puis d'employer une r -distance pour définir la mesure de similarité. L'indice de disimilarité le plus souvent utilisé est la métrique de Minkowski :

$$d_r(o_i; o_j) = d_r(x_i; x_j) \sum_{k=1}^M W_k \left(|x_{ik} - x_{jk}|^r \right)^{\frac{1}{r}} \quad (2.1)$$

W_k est un facteur de pondération que nous considérerons égal à 1 par la suite. Suivant la valeur de r (sachant que r est strictement supérieur à zéro) on obtient des mesures différentes. Pour $r=2$, on mesure la distance euclidienne ; pour $r=1$, c'est la distance de Manhattan.

Ces mesures sont souvent utilisées pour des données numériques. Dans le cas de données symboliques (qualitatives nominales), d'autres distances doivent être utilisées. La plus connue est la distance de Hamming, à l'origine définie pour mesurer la distance entre des chaînes binaires, et qui correspond au nombre de bits différents dans les deux chaînes pour une même position.

7.2.3 La classification

Les probabilités de déplacement P_p et de dépôts P_d des objets sont donnés par :

$$P_p(i) = \left(\frac{k_1}{k_1 + f(o_i)} \right)^2 \quad (2.2)$$

$$P_d(i) = \begin{cases} 2 \cdot f(o_i) & \text{si } f(o_i) \leq k_2 \\ 1 & \text{sinon} \end{cases} \quad (2.3)$$

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in R_s(r(o_i))} 1 - \frac{d(o_i, o_j)}{\alpha} & \text{si } f \geq 0 \end{cases} \quad (2.4)$$

$r(o_i)$ est la position de l'objet o_i sur la grille. $f(o_i)$ est une mesure de similarité moyenne de l'objet o_i avec les objets o_j de son entourage. Le facteur α contrôle la consistance de la

fonction de similarité entre les objets. Si α est trop élevé les objets différents seront mis dans la même classe. Dans le cas contraire, les objets similaires ne seront pas regroupés ensemble. L'algorithme 2.5 présente les étapes de l'algorithme Lumer & Faieta en utilisant A fourmis. Les valeurs de paramètres de l'algorithme sont : $k_1=0.1$, $k_2=0.15$, $s=3$, $\alpha=0.5$.

Algorithme 2.5 : Algorithme LF
Placer aléatoirement les N objets o_1, \dots, o_N sur la grille G pour T = 1 à T _{max} faire pour tout $a_j \in \{ a_1, \dots, a_A \}$ faire si la fourmi a_j ne transporte pas d'objet et $r(o_i) = r(a_j)$ alors Calculer $f(o_i)$ et $P_p(o_i)$ La fourmi a_j ramasse l'objet o_i suivant la probabilité $P_p(o_i)$ sinon si la fourmi a_j transporte l'objet o_i et la case $r(a_j)$ est vide alors Calculer $f(o_i)$ et $P_d(o_i)$ La fourmi a_j dépose l'objet o_i sur la case $r(a_j)$ avec une probabilité $P_d(o_i)$ finsi finsi Déplacer la fourmi a_j sur une case voisine non occupée par une autre fourmi finpour retourner l'emplacement des objets sur la grille

7.2.4 Critiques et limites de l'algorithme basal

- L'inconvénient majeur de cet algorithme est sa grande lenteur. L'origine de cette faiblesse est la vitesse fixe de déplacement de fourmis.
- Dans LF, les objets peuvent être déplacés plusieurs fois même s'ils sont bien classés qui représente un autre point faible de cet algorithme.
- Les résultats obtenus ont montré que l'algorithme génère un nombre de classes qui est souvent très éloigné du nombre réel de classes.

7.2.5 Amélioration de l'algorithme

Afin de remédier aux inconvénients de l'algorithme LF, nous avons introduit trois extensions au comportement de base des fourmis artificielles [Kadri et *al.*, 2011b] :

1. Les fourmis se déplacent sur la grille avec une vitesse propre à chacune d'elles comprise entre 1 et v . les fourmis les plus rapides sont moins sensibles aux dissimilarités entre deux objets, ce qui permet de diminuer le nombre de classes générées.

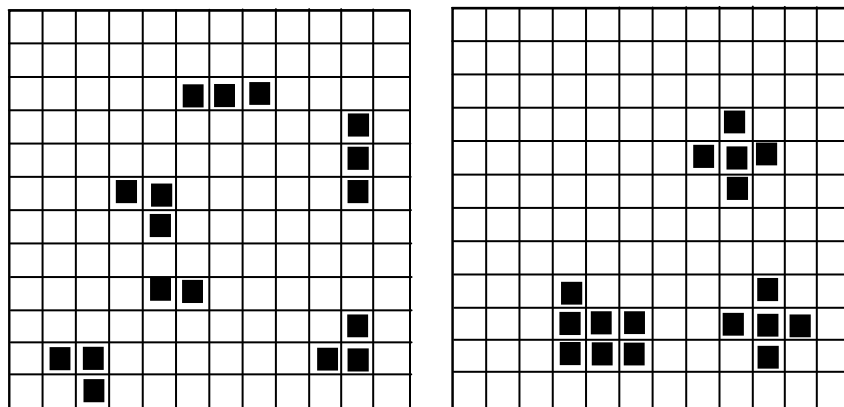


Fig. 2.10 : Résultats possibles de l'exécution de LF avec et sans des fourmis rapides

2. Chaque fourmi possède une mémoire à court terme lui permettant de se souvenir des positions des M derniers objets classés. Si une fourmi transporte un objet, elle cherche dans sa mémoire l'objet déjà classé qui est le plus proche de celui qu'elle transporte. Si elle le trouve, elle se déplacera (avec une certaine probabilité) vers ce nouvel emplacement pour y déposer son objet.
3. Si au bout d'un certain nombre d'itérations la fourmi stagne (ne fait plus aucun déplacement) elle peut détruire un groupe en ramassant l'objet le plus éloigné du groupe. Cette dernière action permet de réduire le nombre de classes (Figure 2.10).

7.3 L'algorithme binaire de colonie de fourmis

7.3.1 Les grandes lignes

L'hybridation d'algorithmes à base de fourmis avec les méthodes génétique a donné un type d'algorithmes plus puissant et plus rapide. Ce type combine l'aspect coopératif d'algorithmes à base de fourmis avec l'aspect évolutif de méthodes génétiques. L'algorithme binaire de colonie de fourmis appartient à ce type d'algorithme. Il consiste de quatre opérations de base : croisement, sélection, mutation et la mise à jour de phéromone. Le principe général de L'algorithme binaire de colonie de fourmis peut être résumé de la façon suivante [Weiqing et al., 2006]:

Algorithme 2.6 : L'algorithme binaire de colonie de fourmis [Weiqing et al., 2006]

Génération aléatoire de la population initiale

Exécution de l'algorithme génétique N fois pour obtenir une solution qui va être utilisée pour initialiser la phéromone sur le réseau.

Répéter

 Lancer les fourmis de recherche

 Exécuter l'algorithme génétique en utilisant la solution générée par l'algorithme de colonie de fourmis

 Mis à jour de phéromone sur le réseau par la solution optimale

Jusqu'à satisfaire le critère d'arrêt

7.3.2 Critiques et limites de l'algorithme basal

- Le principal inconvénient de l'algorithme Binay ant colony est sa grande lenteur puisqu'il exige plusieurs exécutions d'un algorithme génétique pour l'initialisation de phéromone sur le réseau et d'autres exécutions dans chaque itération.
- Un autre point faible de cet algorithme est la loi de déplacement qui est très complexe. Le calcul de cette loi de probabilité nécessite un temps considérable puisqu'elle dépend de tous les nœuds de réseau.

7.3.3 Amélioration de l'algorithme

Pour ce troisième algorithme, nous allons traiter un autre problème qui est la sélection de paramètres. Dans ce cas, le critère d'arrêt représente la dimension du vecteur obtenu par l'algorithme où le critère de qualité n'évolue pas si on ajoute un autre paramètre (figure 2.11).

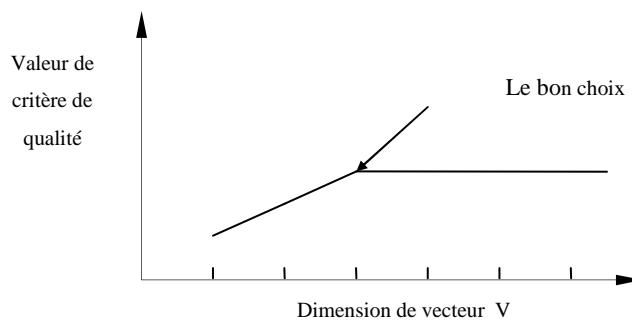


Fig. 2.11 : Le choix de la dimension de vecteur

Pour la représentation de l'information, nous avons utilisé un codage binaire. La dimension de chaque individu égale au nombre total de paramètres. La forme générale d'un vecteur est: $v = (x_1, x_2 \dots x_n)$. Sachant que si le paramètre est présent, x_i prend la valeur 1 sinon il prend la

valeur 0. De cette façon, uniquement les paramètres qui correspondent leurs positions à la valeur 1 seront considérés pour calculer la fonction d'adaptation (figure 2.12).

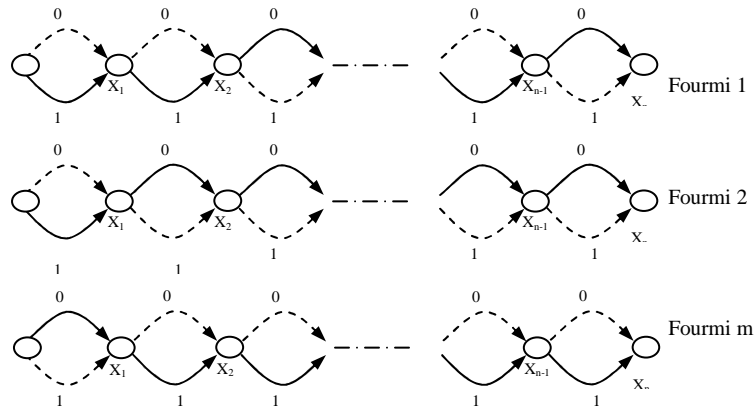


Fig. 2.12 : L'ensemble de paramètres sélectionnés par l'algorithme Filter based ACO [Kadri et al., 2010b]

Nous avons proposé un ensemble de modification sur l'algorithme binaire pour l'adapter au problème de sélection de paramètres. Initialement, nous créons m agents qui représentent la population initiale. Chaque agent est caractérisé par un vecteur V qui ne contient que des zéros et des uns. Après l'exécution de la procédure de recherche, chaque vecteur V est associé à une fonction d'adaptation $F(V)$. La valeur retournée par la fonction F indique la qualité de la solution V . la fonction F se base sur l'utilisation de deux critères de qualités qui sont la variance intra-classe et la variance interclasse.

$$F(V) = \text{trace} \left(\sum_{w=1}^M \sum_{B=1}^{N_C} \frac{1}{w} \right) \quad (2.5)$$

Sachant que la matrice de variance intra-classe est calculée comme suit :

$$\sum_W = \frac{1}{N} \sum_{C=1}^M \sum_{V=1}^{N_C} (X_{CV} - m_C) \cdot (X_{CV} - m_C)^t \quad (2.6)$$

Alors que la matrice de la variance interclasse représente la séparation entre les différentes classes.

$$\sum_B = \frac{1}{N} \sum_{C=1}^M (m_C - m) \cdot (m_C - m)^t \quad (2.7)$$

Avec :

- m : centre de gravité général
- M : nombre de classes

- m_C : centre de gravité de la classe numéro C
- X_{cv} : le $v^{\text{ième}}$ vecteur de la classe numéro C
- N_C : nombre de vecteurs de la classe numéro C
- N : nombre total de vecteurs

Les valeurs de phéromone sur les arcs sont initialisées d'une façon aléatoire. Ensuite, chaque fourmi de la colonie trouve une solution.

L'arc choisi par une fourmi dépend d'une probabilité PS que nous avons proposé. Cette probabilité est définie proportionnellement à un facteur phéromonal FP et un facteur heuristique FH [Kadri et al., 2010c].

$$PS_{i1} = \frac{FP_{i1} + \frac{FP_{i0}}{\text{Max}(FH)} FH_i}{FP_{i1} + FP_{i0}} \quad (2.8)$$

Le facteur phéromonal dépend de la stratégie phéromonale choisie est décrit plus tard. Le facteur heuristique FH est basé sur le rapport de Fisher qui nous permet de classer les paramètres par ordre de pertinence. Ce rapport est exprimé par la formule suivante :

$$FH(\alpha) = \sum_{c=1}^M \sum_{\substack{r=1 \\ r \neq c}}^{M-1} \frac{m_c(\alpha) - m_r(\alpha)}{N_c \sigma_c^2(\alpha) - N_r \sigma_r^2(\alpha)} \quad (2.9)$$

Sachant que :

M représente le nombre de classes.

$m_c(\alpha)$ représente le centre de gravité de la classe numéro c en ne considérant que le paramètre α . Il est calculé comme suit :

$$m_c(\alpha) = \frac{1}{N_c} \sum_{v=1}^{N_c} X_{cv}(\alpha) \quad (2.10)$$

Avec X_{cv} est le vecteur numéro v de la classe numéro c . La valeur de N égale au nombre de vecteurs de la classe en question.

$\sigma_r^2(\alpha)$ est la variance de la composante a des vecteurs de la classe numéro c .

$$\sigma_r^2(\alpha) = \frac{1}{N_c} \sum_{v=1}^{N_c} [X_{cv}(\alpha) - m_c(\alpha)]^2 \quad (2.11)$$

Le facteur phéromonal FP représente la quantité de phéromone disposée par la colonie sur un arc. La mise à jour de FP est effectuée après que toutes les fourmis ont trouvé des solutions. Nous modifions les quantités de phéromone FP sur les arcs uniquement pour les chemins de fourmis qui ont réalisé un rapport important de la dispersion entre classes et de la dispersion intra-classe, selon la formule :

$$\Delta FP = \frac{1}{1 + F(V) - F(V')} \quad (2.12)$$

Sachant que : $F(V)$ représente la meilleure solution construite depuis le début de l'exécution et $F(V')$ représente la meilleure solution construite pendant la dernière itération.

Donc la formule complète de mise à jour de phéromone est comme suit :

$$FP \leftarrow (1 - \rho)FP + \Delta FP \quad (2.13)$$

Sachant que $\rho \in]0,1[$ et représente le taux d'évaporation. Il est utilisé pour ne pas converger rapidement vers des optima locaux.

Pour simplifier la méthode, nous n'avons pas réutilisé l'algorithme génétique dans la boucle de recherche (algorithme 2.7). L'algorithme s'arrête si on converge vers une solution optimale. Mais nous pouvons aussi fixer le nombre total d'itérations à effectuer [Kadri et al., 2011a].

Algorithme 2.7 : L'algorithme Filter based ACO proposé

initialiser la phéromone sur le réseau ;

Calculer $FH(a)$ en utilisant l'équation 2.9 ;

Répéter

Lancer les fourmis de recherche en appliquant l'équation 8 ;

Evaluer les solutions trouvées en utilisant l'équation 2.5 ;

Mise à jour de phéromone sur le réseau par la solution optimale et l'équation 2.13 ;

Jusqu'à satisfaire le critère d'arrêt

Notre algorithme est relativement rapide puisque sa complexité est d'ordre $I * M$, sachant que I représente le nombre d'itération et M représente le nombre de fourmis.

8 Conclusion

Dans ce chapitre, nous avons présenté quelques caractéristiques essentielles des fourmis artificielles. Après une étude critique de trois algorithmes, nous avons trouvé que les formes basales de ces algorithmes ne peuvent pas contribuer à résoudre le problème de diagnostic des systèmes dynamiques et complexes puisqu'elles présentent un ensemble d'inconvénients.

Nous verrons par la suite comment appliquer les algorithmes proposés pour résoudre le problème de diagnostic industriel. Dans la deuxième partie de ce chapitre, trois variantes d'algorithmes de colonies de fourmis ont été proposées. Les trois algorithmes peuvent traiter des données issues d'un système industriel dans les différentes phases du processus de diagnostic.

Deux des trois algorithmes proposés sont des classifieurs non supervisés. Ils permettent de suivre l'évolution du système en détectant l'apparition des nouvelles classes de dysfonctionnement. Mais l'inconvénient majeur de ces deux algorithmes est le risque de diviser les données de la même classe sur plusieurs sous-classes.

Le troisième algorithme est un outil très efficace pour l'optimisation. Nous allons l'utiliser par la suite pour la sélection des paramètres d'un système industriel et aussi pour l'optimisation des paramètres d'un classifieur. Pour ce faire, nous avons choisi d'utiliser un classifieur nommé Séparateurs à Vaste Marge (SVM).

Le chapitre suivant répond au problème de qualité de classification. Il présente une description détaillée d'un classifieur très performant le SVM. Nous avons choisi ce type de classifieur à cause de sa simplicité et de sa capacité d'extension à une classification non linéaire qui est le cas d'un système industriel.

LES SEPARATEURS A VASTES MARGES

Résumé

L'objectif de ce chapitre est la présentation des séparateurs à vaste marge, un classifieur très performant, que nous allons utiliser avec l'algorithme binaire de colonie de fourmis. Dans la première partie de ce chapitre, nous présenterons quelques notions de base qui sont Hyperplan, marge et support vecteur. Nous mettrons également en évidence l'utilisation des SVM dans les deux cas où les données sont (linéairement ou non linéairement) séparables. Nous finissons par une présentation de quelques algorithmes pour la classification SVM multi-classes.

1 Introduction

Comme nous l'avons constaté dans le chapitre précédent, les deux algorithmes (AntTreeStock et LF) n'offrent pas une fonction d'apprentissage. Dans le présent chapitre, nous introduisons les séparateurs à vaste marge que nous allons utiliser pour développer notre module diagnostic (figure 3.1).

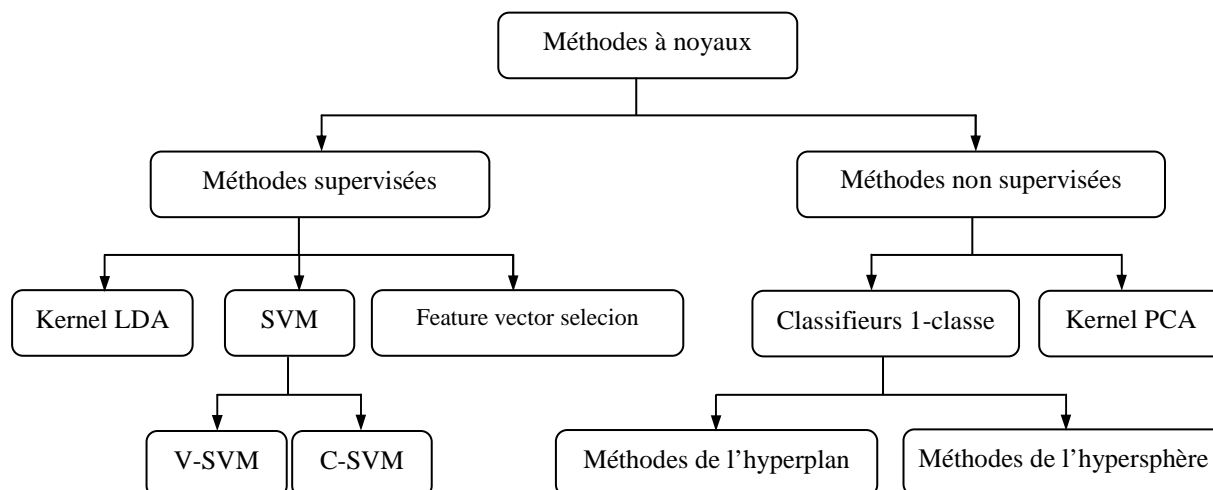


Fig. 3.1: Classification des méthodes d'apprentissage à base de noyaux [NedjemEddine, 2004]

2 Historique

Les séparateurs à vaste marge ou les machines à vecteurs de support, plus connues sous l'acronyme anglais SVM (Support Vector Machine) sont proposées par Vapnik en 1995 [Vapnik, 1995]. Elles sont de plus en plus utilisées dans le domaine de classification des données. Leur objectif principal est de trouver un séparateur optimal qui maximise la marge entre deux classes. Les classifieurs SVM utilisent des fonctions appelées noyaux (kernel) qui permettent une séparation optimale des données [NedjemEddine, 2004].

Depuis sa création par Vapnik, SVM a suscité l'intérêt de plusieurs chercheurs de différents domaines grâce à leur capacité à résoudre des problèmes non linéaires par apprentissage. On cite par exemple les travaux de Pai qui a appliqué une approche hybride basée sur les algorithmes génétiques et le SVM pour faire le pronostic d'un système industriel [Burges, 1998]. Chen et ses collègues [Chen et al., 2001] ont traité la reconnaissance de visages dans une séquence vidéo. Anguita et ses collègues [Anguita et al., 2005] ont utilisé le SVM pour la sélection des paramètres. Dans la plupart des cas, la performance du SVM égale ou dépasse celle de réseaux de neurones. D'autres auteurs ont aussi étudié l'effet de changement de

noyaux d'hyper-paramètres sur la qualité et la vitesse de ce type de classifieurs [Anguita et Bozza, 2005].

3 Les SVM

3.1 L'hyperplan optimal

Etant donné un ensemble de données d'apprentissage $S = \{(x_1, u_1), \dots, (x_i, u_i), \dots, (x_m, u_m)\}$, avec $x_i \in \mathbb{R}^d$ et $u_i \in \{-1, +1\}$. On suppose dans un premier temps qu'il existe un hyperplan linéaire permettant de discriminer les données positives (étiquetées +1) des données négatives (étiquetées -1). L'identification d'un tel hyperplan dans X revient à trouver une fonction hypothèse $h(x) = w^t x + w_0$ telle que :

$$w^t x + w_0 = \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow u_i = \begin{cases} -1 \\ +1 \end{cases} \quad (3.1)$$

Cet hyperplan séparateur est valide sur l'ensemble d'apprentissage si :

$$\forall 1 \leq i \leq m, u_i h(x_i) > 0.$$

Soit encore si : $\forall 1 \leq i \leq m, (w x_i + w_0) > 0$.

La fonction $h(x)$ correspond à l'équation d'un hyperplan dans X de vecteur normal w . la distance d'un point x à l'hyperplan d'équation $h(x) = w x + w_0$ est égale à $h(x) / \|w\|$, où $\|w\|$ est la norme euclidienne du vecteur w . En général, il existe une infinité de séparateurs linéaires entre les données d'apprentissage. On peut alors chercher parmi ces hyperplans, un hyperplan séparateur optimal celui qui se trouve au milieu des deux ensembles de données positives et de données négatives (figure 3.2).

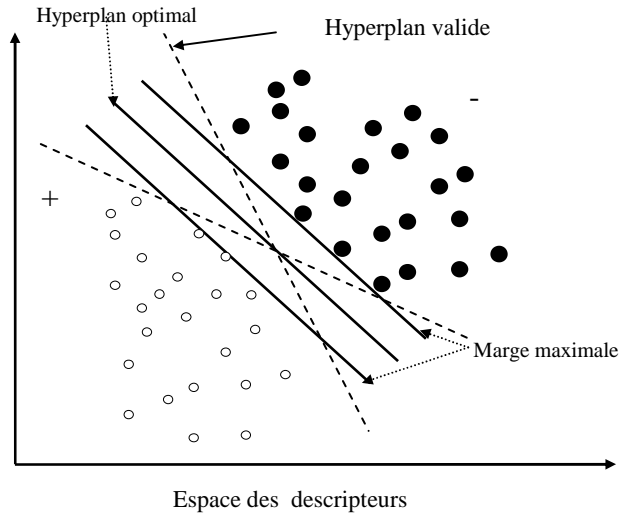


Fig. 3.2: Séparation de deux ensembles de points par des séparateurs linéaires.

L'hyperplan séparateur optimal est défini par les vecteurs de poids w et w_0 vérifiant l'équation suivante :

$$\operatorname{argmax}_{w, w_0} \min\{\|x - x_i\| : x \in \mathbb{R}^d, (wx + w_0) = 0, i = 1, \dots, m\}$$

C'est-à-dire l'hyperplan séparateur qui maximise la distance minimale aux données d'apprentissage (figure 3.3). Dans ce cas, la marge vaut : $2/\|w\|$.

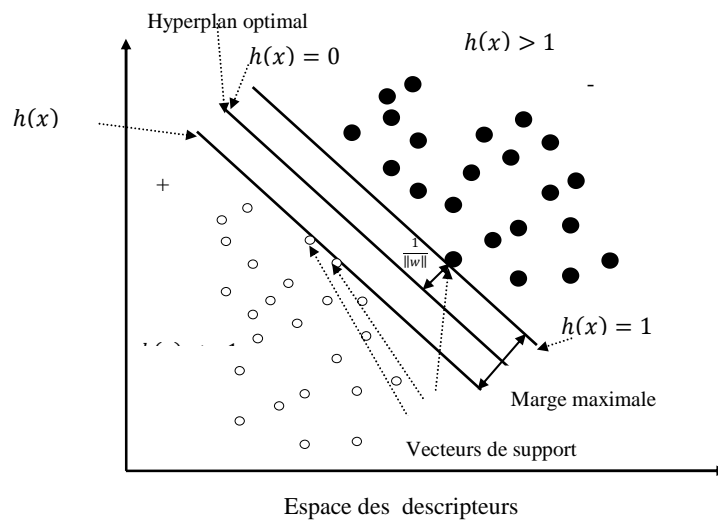


Fig. 3.3: L'hyperplan optimal est perpendiculaire au segment de droite le plus court joignant un exemple d'apprentissage à l'hyperplan

4 SVM : Formulation Primale

La recherche de l'hyperplan optimal revient donc à minimiser $\|w\|$ soit à résoudre le problème d'optimisation suivant qui porte sur les paramètres w et w_0 :

$$\left\{ \begin{array}{l} \text{minimiser } \frac{1}{2} \|w\|^2 \\ \text{sous les contraintes } u_i(wx + w_0) \geq 1, \quad i = 1, \dots, m \end{array} \right.$$

Ce problème est désigné comme la version primale du problème des SVM. Il nécessite le réglage de $d + 1$ paramètres, sachant que d représente la dimension de l'espace des entrées X . On peut résoudre ce problème par des algorithmes de programmation quadratique pour un nombre réduit de paramètres. Mais il devient impossible si le nombre de paramètres dépasse quelques centaines. Pour ce dernier cas, on peut transformer la version primale de ce problème dans une formulation duale que l'on peut résoudre en pratique.

5 SVM : Formulation Duale

Selon la théorie de l'optimisation, un problème d'optimisation a une forme duale si et seulement si la fonction objective et les contraintes sont strictement convexes. Alors, la solution de la forme duale du problème est la même du problème original.

On peut définir une formulation lagrangienne correspondante au problème dual. La représentation lagrangienne comporte des informations sur les contraintes et sur la fonction objective. Plus précisément, Il s'agit de faire rentrer les contraintes dans la fonction objective et de pondérer chacune d'entre elles par une variable duale. Les variables duales $a_i \geq 0$ sont appelés multiplicateurs de Lagrange.

$$L(w, w_0, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m a_i (u_i((x_i \cdot w) + w_0) - 1) \quad (3.2)$$

En appliquant le théorème de Kuhn-Tucker, on peut calculer w en fixant seulement n paramètres qui correspond au nombre de données de test. Le théorème de Kuhn-Tucker, prouve que la solution de la formulation duale correspond à un *point-selle* du lagrangien. L'idée est de maximiser L par rapport aux variables duales a_i et de le minimiser par rapport aux variables primaires w et w_0 . Le *point-selle* doit donc satisfaire les conditions nécessaires de stationnarité.

Ceci s'écrit :

$$\frac{\partial}{\partial w} L(w, w_0, a) = 0, \quad \frac{\partial}{\partial w_0} L(w, w_0, a) = 0 \quad (3.3)$$

et conduit à :

$$\sum_{i=1}^m a_i u_i = 0 \quad (3.4)$$

et à :

$$w = \sum_{i=1}^m a_i u_i x_i \quad (3.5)$$

Les multiplicateurs de Lagrange sont non nuls pour les points qui se trouvent sur les hyperplans frontières $(x_i \cdot w) + w_0 = \pm 1$ jouent un rôle. Ces points représentent les vecteurs de support et ils sont utilisés pour calculer l'hyperplan optimal.

En substituant (3.4) et (3.5) dans (3.3), nous obtenons la forme duale du problème d'optimisation :

$$\begin{cases} \text{Max}_a \left\{ \sum_{i=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j (x_i \cdot x_j) \right\} \\ a_i \geq 0, i = 1, \dots, m \\ \sum_{i=1}^m a_i u_i = 0 \end{cases} \quad (3.6)$$

Ce dernier problème peut être résolu en utilisant des méthodes standards de programmation quadratique. L'hyperplan solution correspondant peut alors être écrit :

$$h(x) = (w^* \cdot x) + w_0^* = \sum_{i,j=1}^m a_i^* u_i \cdot (x \cdot x_i) + w_0^* \quad (3.7)$$

où les a_i^* sont la solution de (3.6) et Le paramètre de biais w_0^* permet de définir des surfaces de séparation ne passant pas par l'origine. Il est calculé en utilisant un exemple critique (x_c, u_c) dans l'équation :

$$a_i \left[u_i \cdot \left((x_i \cdot w^*) + w_0 \right) - 1 \right] = 0 \quad (3.8)$$

D'après cette dernière équation, on remarque que le calcul de la solution de l'hyperplan optimal ne nécessite que l'évaluation des produits scalaires $\langle x, x_i \rangle$ entre des vecteurs de la base de test X et donc la solution ne dépend plus de la dimension d de l'espace d'entrée, mais d'un nombre réduit de vecteurs de données qui représentent des vecteurs critiques.

La section précédente décrit le principe de calcul d'un séparateur SVM dans le cas où les données sont linéairement séparables. Mais dans la pratique, ce n'est pas toujours le cas et il est donc indispensable de trouver une solution pour ce nouveau problème.

6 Le cas d'un échantillon non linéairement séparable

Dans le cas des données non linéairement séparable, Le problème d'optimisation énoncé dans l'équation 3.6 n'admit pas une solution puisque les conditions de Tucker (KKT) ne sont pas satisfaites. Cortes et ses collègue ont ajouté à la méthode précédente un paramètre de pénalisation appelé C [Li et al., 2002]. Ce paramètre est utilisé pour rendre la distance entre les vecteurs bien classés et l'hyperplan optimal maximale et la distance entre les vecteurs mal classés et l'hyperplan optimal minimale. Pour ce fait, on modifie les contraintes en les relâchant grâce à des variables ressort $\xi_i \geq 0$:

$$u_i((w \cdot x_i) + w_0) \geq 1 - \xi_i \quad (3.9)$$

L'objectif est de minimiser la fonction suivante :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (3.9)$$

Pour une constante $C > 0$.

Le résultat est comme dans le cas des données linéairement séparables, une forme duale mais avec une légère différence dans les contraintes.

$$\left\{ \begin{array}{l} \text{Max}_a \left\{ \sum_{i=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j (x_i \cdot x_j) \right\} \\ C \geq a_i \geq 0, \forall i \\ \sum_{i=1}^m a_i u_i = 0 \end{array} \right. \quad (3.10)$$

On voit que le paramètre de pénalisation C est utilisé pour maximiser la marge possible entre les classes et minimiser le nombre d'erreurs admissibles. Donc, Il doit être choisi selon la nature de données et l'avis de l'expert.

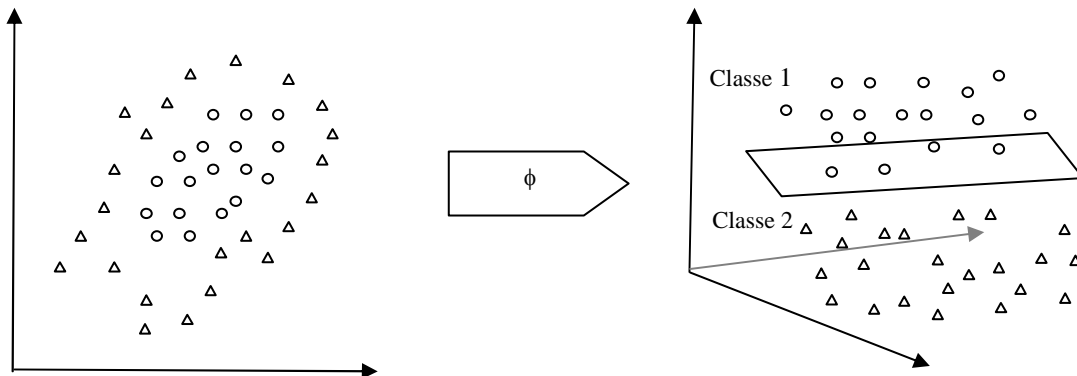


Fig. 3.4 : Exemple de projection non-linéaire.

7 Les fonctions noyaux

Les sections précédentes décrivent l'essentiel des SVM dans le cas du problème de classification des données linéairement séparables. Les SVM ne souffrent pas du problème d'optima local puisqu'il existe seulement un optimum global facile à trouver. Boser et ses collègues [Boser et al., 1992] ont démontré qu'un problème de séparation non linéaire des classes peut se transformer en problème de séparation linéaire des classes. Ils ont utilisé un noyau de Mercer [Vapnik, 1995] qui est basé sur l'idée que plus la dimension de l'espace de présentation de données est grande, plus la probabilité de pouvoir trouver un séparateur linéaire entre les classes est élevée. Si on applique une projection des données de l'espace d'entrée sur un espace de redescription qui possède une dimension plus grande, il sera facile d'appliquer les SVM sur ce nouveau ensemble de données. Notons ϕ une transformation non linéaire de l'espace d'entrée X en un espace de redescription $\phi(X)$:

$$x = (x_1, \dots, x_d)^T \rightarrow \phi(x) = (\phi_1(x), \dots, \phi_d(x), \dots) \quad (3.11)$$

Le problème d'optimisation se transcrit dans ce cas par :

$$\begin{cases} \text{Max}_a \left\{ \sum_{i=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j \langle \phi(x_i), \phi(x_j) \rangle \right\} \\ C \geq a_i \geq 0, \quad i = 1, \dots, m \\ \sum_{i=1}^m a_i u_i = 0 \end{cases} \quad (3.12)$$

Où $\langle \cdot, \cdot \rangle$ indique le produit scalaire.

L'équation de l'hyperplan séparateur dans l'espace de redescription devient :

$$h(x) = \sum_{i=1}^m a_i^* u_i \langle \phi(x), \phi(x_j) \rangle + w_0^* \quad (3.13)$$

où les coefficients a_i^* et w_0^* sont obtenus par résolution de (3.12).

Le produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$ devient difficile voire impossible à calculer quand la dimension de $\phi(X)$ est relativement grande. On remarque que dans la résolution de problème précédent seul le résultat de produit scalaire est nécessaire. Il existe une fonction $K(x,y)$, appelée fonction noyau qui permet de calculer le produit scalaire $\langle \phi(x), \phi(y) \rangle$ dans un espace de grande dimension sans passer par le calcul de valeurs des nouvelles coordonnées dans l'espace de redescription. En introduisant la fonction noyau, le problème d'optimisation (3.12) devient alors:

$$\begin{cases} \text{Max}_a \left\{ \sum_{i=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j K(x_i, x_j) \right\} \\ C \geq a_i \geq 0, \quad i = 1, \dots, m \\ \sum_{i=1}^m a_i u_i = 0 \end{cases} \quad (3.14)$$

dont la solution de la version noyau est d'équation de l'hyperplan séparateur suivante :

$$h(x) = \sum_{i=1}^m a_i^* u_i K(x, x_i) + w_0^* \quad (3.15)$$

où les coefficients a_i^* et w_0^* sont obtenus comme précédemment par résolution de (3.14).

La fonction noyau polynomiale [Vapnik, 1995] (3.16) peut calculer le résultat du produit scalaire dans un espace de n dimensions :

$$K(x, y) = (x \cdot y)^n \quad (3.16)$$

La figure 3.3 représente un exemple de changement de description de vecteurs bidimensionnels par un noyau Polynomial où $n = 2$ et $x, y \in R^2$ dans R^3 , alors on obtient :

$$(x, y)^2 = (x_1^2, x_2^2, \sqrt{2}x_1x_2)(y_1^2, y_2^2, \sqrt{2}y_1y_2)^T = \langle \phi(x), \phi(y) \rangle$$

Avec

$$\phi : X \equiv R^2 \rightarrow R^3 \equiv T$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \phi(x) = \begin{pmatrix} \phi(x)_1 \\ \phi(x)_2 \\ \phi(x)_3 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix}$$

Un deuxième exemple est la fonction noyau $K(x, y) = (x \cdot y + c)^n$ avec la valeur de c est positive. Cette fonction correspond à un produit scalaire dans l'espace des descripteurs correspondant à tous les produits d'ordre $\leq n$.

L'avantage dans l'utilisation des fonctions noyau est qu'il n'est pas nécessaire de définir explicitement la fonction ϕ . Mais le choix de la fonction noyau reste un grand dilemme. Donc, l'utilisateur est obligé de faire plusieurs exécutions en essayant à chaque fois une fonction noyau différente. C'est là, bien entendu, que se pose une autre question importante : Quelles sont les fonctions rapides et simples qui peuvent donner un résultat correspondant à $\phi(x) \cdot \phi(y)$? En réalité, ces fonctions doivent satisfaire un ensemble de conditions pour être équivalentes à un produit scalaire.

Soit donc la redescription des formes x de X par la fonction ϕ dans un espace de Hilbert de coordonnées $\phi_1(x), \dots, \phi_d(x), \dots$, selon le théorème de Hilbert-Schmidt, tout produit scalaire dans l'espace de Hilbert a la forme suivante :

$$\langle \phi_1(x), \phi_1(x') \rangle = \sum_{i=1}^{\infty} \lambda \phi_i(x) \phi_i(x') = K(x, x') \quad (3.17)$$

Théorème de Mercer

Si $K(.,.)$ est une fonction noyau continue symétrique d'un opérateur intégral

$$g(y) = Af(y) = \int_a^b K(x, y)f(y)dy + h(x)$$

vérifient :

$$\int_{X \times X} K(x, x')f(x)f(x') \geq 0$$

pour toute fonction $f \in L_2(X)$ (de carré sommable) (X étant un sous-espace compact de R^d), alors la fonction $K(., .)$ peut être développée en une série uniformément convergente en fonction des valeurs propres positives λ_i et des fonctions propres ψ_i :

$$K(x, x') = \sum_{j=1}^K \lambda_j \psi_j(x) \psi_j(x') \quad (3.18)$$

où N représente le nombre de valeurs propres positives.

Le théorème de Mercer indique si une fonction $K(., .)$ est un noyau, mais ne donne aucune information sur la redescription ϕ .

Une condition équivalente à celle de Mercer est que les matrices $K_{ij} := K(x_i, x_j)$ soient positives semi-définies pour tout échantillon d'exemples $\{x_i, \dots, x_m\}$ (ces matrices sont appelées matrices de Gram).

On peut alors décrire la fonction ϕ de redescription des entrées comme :

$$\Phi(x) = (\sqrt{\lambda_1} \psi_1(x), \sqrt{\lambda_1} \psi_1(x), \dots) \quad (3.19)$$

Même si les théorèmes présentés ci-dessus sont importants, il est très difficile de les appliquer en pratique. La solution est résidée donc dans l'utilisation des noyaux précédemment testés par

d'autres chercheurs du même domaine. Le tableau 3.1 présente les noyaux les plus utilisés dans le domaine de classification :

Tableau 3.1 : Les noyaux SVM les plus utilisés dans le domaine de classification

La solution s'exprime sous la forme : $h(x) = \sum_{i=1}^{n_{support}} a_i^* u_i K(x, x_i) + w_0^*$		
Polynomiale	$K(x, x') = (x \cdot x' + 1)^p$	La puissance p est déterminée à priori par l'utilisateur
Fonction à Base Radiale (RBF)	$K(x, x') = e^{-\frac{\ x-x'\ ^2}{2\sigma^2}}$	L'écart-type σ^2 , commun à tous les noyaux, est spécifiée à priori par l'utilisateur.
Fonctions sigmoïdes	$K(x, x') = \tanh(a(x \cdot x' - b))$	Le théorème de Mercer n'est vérifié que pour certaines valeurs de a et b .

8 Les hyper-paramètres

Avant de réaliser la phase d'apprentissage en utilisant un modèle SVM, on doit tout d'abord fixer quelques paramètres. Les valeurs de ces paramètres ne sont pas modifiables lors de la phase d'apprentissage. SVMs (les variantes de SVM) possèdent un autre ensemble de paramètres appelés hyper-paramètres qui peuvent être optimisées durant la phase d'apprentissage. Un exemple de ce type de paramètre est la constante de régularisation qui représente le facteur de Lagrangien associé au terme de pénalisation rajouté. Elle est utilisée pour régler le degré de compromis désiré entre la maximisation de la marge entre les classes et la tolérance du modèle aux erreurs d'apprentissage. Ils existent aussi des hyper-paramètres propres à chaque type de noyaux. Par exemple la forme générale d'un noyau polynomial (tableau 3.1) est donnée par l'expression suivante : $K(x, x') = (Ax \cdot x' + B)^p$

Ce type de noyau est caractérisé par les deux paramètres A et B qui prennent toujours la valeur "1". Par contre P est un hyper-paramètre qui doit être optimisé pendant la phase d'apprentissage. Un autre noyau très utilisé est le noyau Gaussien [Vapnik, 1995] (tableau

3.1) qui possède la forme suivante : $K(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$

Le hyper-paramètre σ permet de régler la largeur de la gaussienne. Si σ prend une valeur très grande, la similarité d'une observation par rapport à son entourage sera assez élevée, alors si la valeur σ est proche de 0, l'observation ne sera similaire à aucune autre. En diminuant strictement la gaussienne, un modèle qui utilise ce type de noyau peut arriver à apprendre n'importe quel ensemble d'apprentissage sans faire d'erreur [Kadri et al., 2012b].

9 SVM Multi-Classes

Comme nous l'avons vu dans les sections précédentes, l'algorithme basal de SVM ne permet pas de traiter le cas de multi-classes. Bien entendu, plusieurs recherches ont été effectuées pour proposer des algorithmes permettant de résoudre le problème de multi-classes. Mais l'inconvénient majeur de ces propositions est qu'elles sont très lentes par rapport à la version binaire. Donc, le problème de classification multi-classes par les SVMs reste toujours un domaine de recherche actif. La figure 3.5 représente un exemple de classification SVM à trois classes.

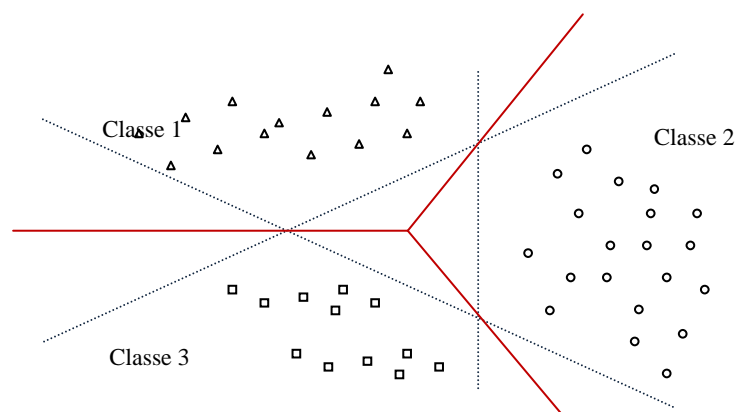


Fig. 3.5 : Exemple de classification SVM multi-classes

Dans les sections suivantes, nous présentons les différentes approches proposées pour l'extension de SVM aux problèmes de classification multi-classes [Yang et al., 2007].

9.1 SVM : Un Contre Tous

Le principe de cet algorithme est d'utiliser autant de classifieurs qu'il y a de classes. Pour l'apprentissage de chaque classifieur, toutes les observations de la classe considérée sont considérées positives (+1), alors que toutes les observations n'appartenant pas à la classe sont considérées négatives (-1). A la fin de cette phase, on dispose de k hyperplans (k égale au nombre de classes). L'inconvénient majeur de cet algorithme est qu'on peut avoir plusieurs sorties positives pour la même observation de test en utilisant les différents classifieurs. La solution pour ce problème est d'utiliser un vote majoritaire pour attribuer cet observation à

une classe. Alors, la décision d'affecter une observation x à une classe k est réalisée en utilisant la suivante : $\arg \max (w_k \cdot x + b_k)$. La critique qu'on peut donner à cette solution est que chaque classifieur SVM est entraîné indépendamment des autres. Donc, chaque classifieur peut utiliser un ensemble différent de vecteurs de support. Il en résulte que les sorties des différents classifieurs ne sont pas comparables. La figure 3.6 représente une architecture d'un système de classification en utilisant l'algorithme Un-Contre-Tous.

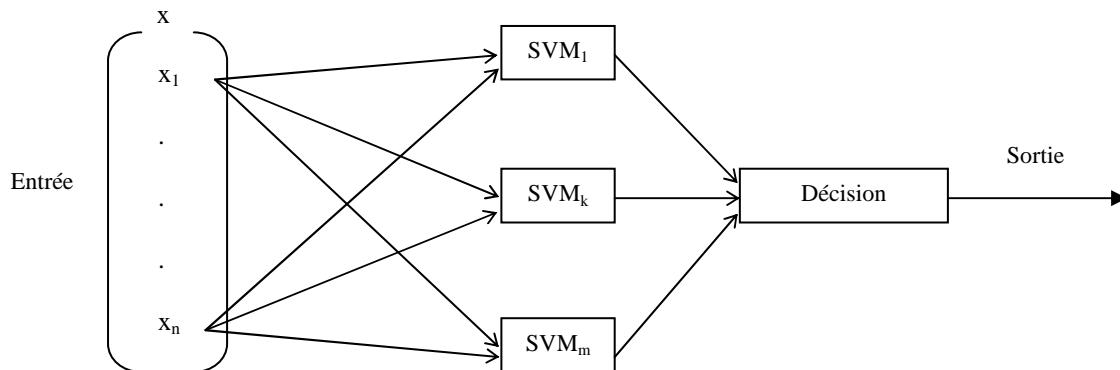


Fig. 3.6 : Architecture d'un système de classification basée sur l'algorithme Un-Contre-Tous

9.2 SVM : Un Contre Un

Ce deuxième algorithme utilise un modèle SVM pour chaque paire de classes. Alors le nombre total de modèles utilisés pour un problème de m classes est $m(m-1)/2$ modèles. Chaque modèle réalise une classification binaire et sépare les données de deux classes. Il considère les données de la première classe comme des éléments positifs et les données de la deuxième classe comme des éléments négatifs. La plupart des systèmes de classification dans cette catégorie utilisent le vote majoritaire comme critère de décision. Donc, pour classer une nouvelle observation on utilise une mesure de vote calculée à partir de la sortie de chaque modèle et l'observation est affectée à la classe qui obtient le plus grand nombre de vote. Il est possible que deux classes obtiennent le même nombre de vote. La figure 3.7 montre une observation qui appartient à une zone d'ambiguïté.

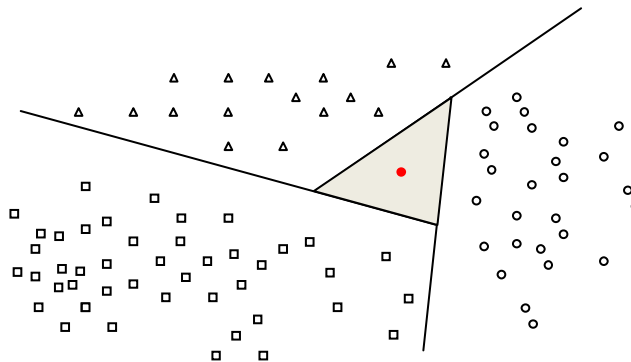


Fig. 3.7 : une observation appartient à la zone d'ambiguïté

Plusieurs solutions ont été proposées pour résoudre ce problème. On cite celle de Hsu et Lin qui utilise une décision aléatoire [Hsu et Lin, 2002].

9.3 DAG-SVM

Une autre solution pour le problème précédent a été proposée par Platt et ces collègues [Platt et al., 2000]. Dans la phase d'apprentissage, cette technique utilise le même principe que celui de la méthode un-contre-un. Donc, elle utilise $m(m-1)/2$ modèles de classification binaire. Mais, la phase de test est basée sur l'utilisation d'un graphe de décision orienté acyclique comme son nom l'indique (*Directed Acyclic Graph*). Dans cette phase, chaque modèle vote contre la classe non choisie. Le résultat de cette action est le rejet d'une classe à la fin de chaque classification binaire. ainsi, on obtient la classe désirée après $m-1$ étapes. La figure 3.8 représente un exemple de trois classes. Cette technique donne le même taux de classification que celui de la méthode un-contre-un, mais elle présente l'avantage d'être la méthode la plus rapide.

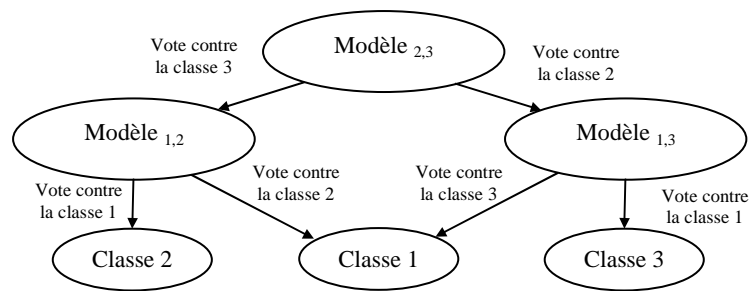


Fig. 3.8 : Une classification de trois classes par SVM-DAG

Plusieurs études comparatives ont été réalisées sur les différentes approches multi-classes de SVM. Les résultats obtenus ne favorisent aucune approche sur les autres en termes de taux de classification si on utilise un noyau linéaire. Par contre, si les hyper-paramètres d'autres noyaux sont bien choisis alors l'approche un-contre-tous est plus précise. Mais, dans tous les cas les approches basées sur le principe de un-contre-un sont plus rapides [Zhang, 2012].

10 La régression par SVM

On peut aussi utiliser les SVMs pour résoudre le problème de la régression [Vapnik, 1995]. On parle ici de SVR (Support Vector Regression). SVR donne de très bons résultats comparativement aux techniques existantes. Son principe est de trouver la fonction de coût :

$$|y - f(x)|_\varepsilon := \max\{0, |y - f(x)| - \varepsilon\} \quad (3.20)$$

où ε est un paramètre à fixer.

L'objectif alors est d'estimer la fonction de régression suivante :

$$f(x) := (w \cdot x) + b \tag{3.21}$$

Avec une valeur fixée de ε , on doit minimiser :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m |y_i - f(x_i)|_\varepsilon$$

On peut réécrire le problème précédent sous la forme d'un problème d'optimisation sous contraintes :

$$\begin{aligned} \text{minimiser } \gamma(w, \varepsilon, \xi) &= \frac{1}{2} \|w\|^2 \\ \text{sous les contraintes } &\begin{cases} ((w \cdot x_i) + b) - y_i \leq \varepsilon + \xi_i \\ y_i - ((w \cdot x_i) + b) \leq \varepsilon + \xi_i^* \\ \xi_i^*, \xi_i \geq 0 \end{cases} \end{aligned} \tag{3.22}$$

Pour tous les i allant de 1 à m , on peut dire que toute erreur moins de ε ne demande pas une valeur positive de ξ ou de ξ^* et donc la fonction objective est calculée sans prise en compte cette erreur.

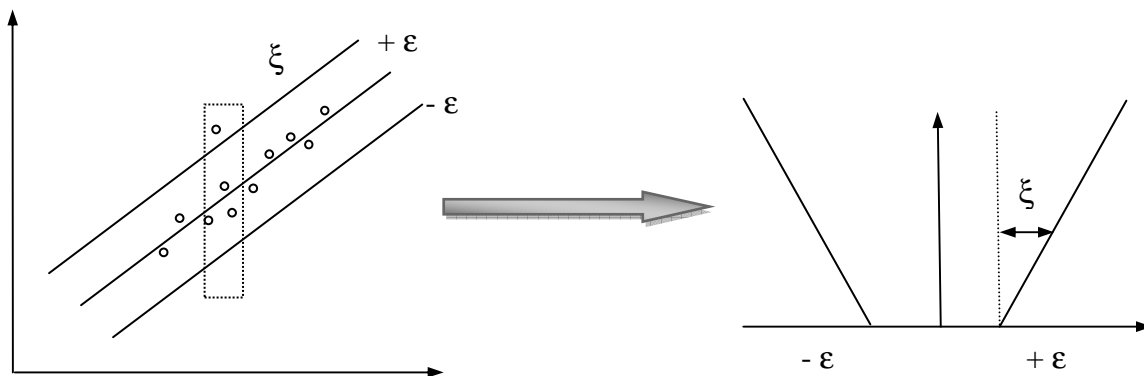


Fig. 3.9 : La régression par SVM

Pour traiter le problème de la régression non linéaire, on doit passer par un espace de redescription en utilisant des fonctions noyaux. Donc, la solution pour ce problème revient à :

Maximiser

$$\begin{aligned}
W(a, a^*) = & -\frac{1}{2} \sum_{i,j=1}^m (a_i - a_i^*)(a_j - a_j^*)k(x_i, x_j) - \varepsilon \sum_{i=1}^m (a_i^* + a_i) \\
& + \sum_{i=1}^m (a_i^* - a_i)y_i
\end{aligned}$$

sous les contraintes $\begin{cases} 0 \leq a_i^*, a_i \leq C & i = 1, \dots, m \\ \sum_{i=1}^m (a_i - a_i^*) = 0 \end{cases}$ (3.23)

L'équation de la régression prend alors la forme suivante:

$$f(x) = \sum_{i=1}^m (a_i^* + a_i) k(x_i, x) + b \quad (3.24)$$

On peut facilement calculer b si $\xi_i = 0$ et $\xi_i^* = 0$ et les contraintes (3.22) deviennent des égalités.

11 Outils logiciels

Il existe plusieurs outils logiciels qui facilitent l'utilisation de SVM. Il existe aussi des boîtes à outils et des bibliothèques Matlab. Dans cette section, nous allons introduire les trois boîtes à outils les plus utilisées qui sont : SVMTorch [Collobert et Bengio, 2001], SVMlight [Joachims, 1999] et LIBSVM [Chang et Lin, 2001]. Une liste complète d'outils validés est publiée sur le site officiel de SVM.

11.1 SVMTorch

SVMTorch est une librairie de Machine Learning qui fonctionne sur les systèmes d'exploitation Solaris, Linux et Windows. Elle a été développée par Ronan Collobert. Elle a été programmée en utilisant le langage C++ [Collobert et Bengio, 2001]. Elle utilise le même principe que celui utilisé par les deux autres outils. SVMTorch permet la classification multi-classes de données dans un grand nombre de dimensions et aussi la régression de données. Elle utilise l'approche un-contre-tous et sélectionne le meilleur score retourné par la sortie de différents classifieurs utilisés. Elle a été testée sur une base de données qui contient plus de 20000 observations. Chaque observation est composée de plus de 100 paramètres [Li et al., 2002]. SVMTorch est composée de quatre classes qui permettent la modification des méthodes existantes ou même l'ajout de nouvelles méthodes.

11.2 LIBSVM

LIBSVM a été développée par C.Chang en 2001 [Chang et Lin, 2001]. Il s'agit d'une boîte à outils SVM qui englobe les approches de classification (C-SVC et nu-SVC) et les approches de régression (epsilon-SVR, nu-SVR et one-class SVM). Elle permet une classification multi-classes. Son algorithme basal est une simplification de l'algorithme SMO de Platt et de l'algorithme SVMLight de Joachims. LIBSVM offre une interface très simple pour utiliser les différentes configurations possibles. Elle est composée de deux modules essentiels qui sont *Svmtrain* pour l'apprentissage et *Svmpredict* pour la classification.

11.3 SVMLight

La boîte à outils SVMlight a été proposée par Joachims en 1999 [Joachims, 1999]. Elle a été développée en utilisant le langage C pour la classification et la régression de données. Elle utilise un algorithme d'optimisation très rapide. Elle peut traiter des bases de données de grande taille (plus de 100.000 d'observations). SVMlight utilise une variante de l'algorithme de décomposition successive de Osuna et ces collègues [Osuna et al., 1997]. Cet algorithme est basé sur une méthode de sélection d'un ensemble actif où les multiplicateurs de Lagrange sont variables. Il utilise une décomposition successive du problème. Sa méthode d'optimisation est basée sur une technique du 'caching' et une modification incrémentale des valeurs de multiplicateurs de Lagrange. SVMlight utilise la méthode *svm_learn* pour l'apprentissage et la méthode *svm_classify* pour la classification.

12 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base de l'approche des machines à vecteurs de support. Nous avons choisi d'utiliser un classifieur SVM puisqu'il répond bien à la formalité et les exigences sollicitées par la fonction de diagnostic. Nous avons vu que l'approche SVM est bien fondée mathématiquement puisque elle a été complètement créée à partir de considérations théoriques. Nous avons commencé par une explication de différentes étapes utilisées pour la recherche d'un séparateur optimal dans les cas de données linéairement séparables et non linéairement séparables. Ainsi nous avons présentée l'astuce de fonction de noyau qui est utilisée dans le cas de données non linéairement séparables pour faire la transformation des données dans l'espace de redescription. Nous avons présenté ensuite les fonctions de noyaux les plus utilisées dans le domaine de classification de données. Aussi nous avons présenté les algorithmes SVM les plus connus pour le traitement des problèmes de

classification multi-classes. Nous avons terminé le chapitre par une présentation de quelques outils logiciels utilisés pour mettre en œuvre l'approche SVM.

Dans le prochain chapitre nous présenterons notre approche de sélection de paramètres pour le diagnostic industriel. Notre approche de sélection est basée sur l'algorithme Filter based ACO présenté dans le chapitre précédant et sur les concepts de SVM que nous avons exposés tout au long de ce chapitre.

LA SELECTION DE PARAMETRES POUR LE DIAGNOSTIC INDUSTRIEL

Résumé :

Dans ce chapitre, nous allons présenter une nouvelle approche de sélection de paramètres qui combine les techniques de méthodes filtres et enveloppantes. Notre approche profite de la rapidité des méthodes filtres et l'adaptation de paramètres sélectionnés au classifieur utilisé. Elle améliore la qualité de classification en fonction de la nature de données de la base d'apprentissage et le type de la fonction noyau utilisé. Notre approche permet aussi de régler les hyper-paramètres de la fonction de noyau. Après une description des deux catégories de méthodes de réduction de dimension, les détails de notre approche seront présentés.

1 Introduction

Le problème de sélection de paramètres a été traité depuis les années 70. Aujourd'hui, nous pouvons le considérer aussi comme un sujet d'actualité à cause de l'utilisation des bases de données de grandes dimensions et l'utilisation de la classification dans le domaine de temps réel [Dash et Liu, 1997]. Les méthodes de sélection de paramètres permettent de classer plus rapidement les données. Elles sont utilisées pour les systèmes de diagnostic industriel où le temps d'identification doit être minimisé. Ces méthodes ne changent pas les paramètres de départ et donc ne compliquent pas le système par la création des nouveaux paramètres (figure 4.1).

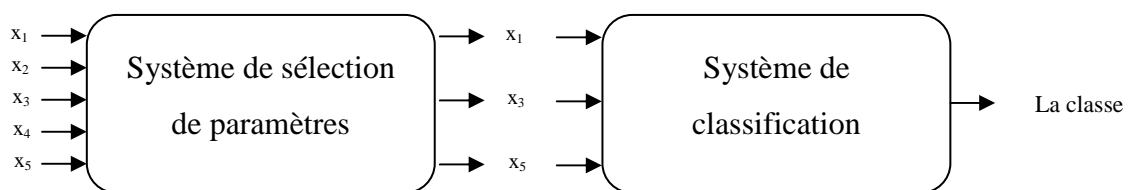


Fig. 4.1: Principe de la sélection de variables [Guérif, 2006]

L'utilisation d'un grand nombre de paramètres et l'existence des paramètres inutiles pour la description d'un système de production, limitent les performances du système de diagnostic. En réalité, il existe toujours des paramètres qui ne sont pas nécessaires dans la détection de mode de fonctionnement du système étudié. Alors, l'objectif est de sélectionner les paramètres jugés pertinents. Plusieurs travaux ont été proposés pour la réalisation de cette phase qui est indispensable dans le processus de reconnaissances de formes. Nous citons quelques exemples de ces travaux dans les sections suivantes selon le type de la méthode adoptée par les chercheurs.

2 Les étapes de la sélection de paramètres

La recherche d'un sous-ensemble de paramètres qui permet une meilleure séparation entre les classe revient à calculer la pertinence de ces paramètres. Donc, il est intéressant de trouver un critère d'évaluation qui permet de mesurer la pertinence de chaque paramètre. Il est aussi indispensable de trouver une procédure de génération qui utilise ce critère. Cette procédure permet de former le sous-ensemble de paramètres. Le dernier point qu'on doit fixer est le critère d'arrêt. Ce critère est généralement dépend de critère d'évaluation et de choix de la procédure de génération [Al-Ani, 2005].

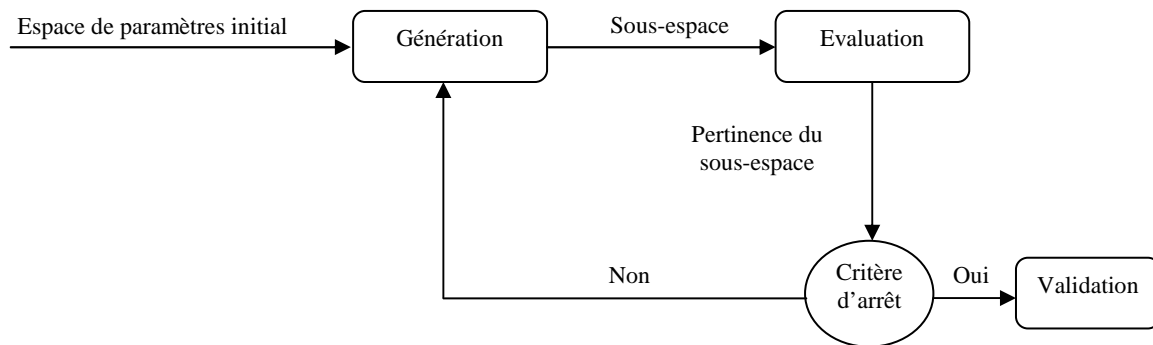


Fig. 4.2: Les différentes étapes d'une procédure de sélection d'attributs

2.1 Critères d'évaluation

Le critère d'évaluation, est comme son nom l'indique, est une mesure qui permet de déterminer la pertinence d'un paramètre. Cette pertinence est relative au domaine d'application. On peut distinguer deux types de critère d'évaluation. Le premier type calcule la pertinence de chaque paramètre d'une façon individuelle. Ce type est utilisé avec une fonction de classement pour former le sous-ensemble final. Le deuxième type calcule la pertinence d'un sous-ensemble de paramètres. Il est utilisé avec une fonction d'évaluation et une procédure de génération. Dans cette catégorie de méthodes, on peut distinguer cinq types de critères d'évaluation :

2.1.1 Les mesures d'erreur de classification

Le sous-ensemble final de paramètres est sélectionné en mesurant la qualité de classification. Donc, ce sous-ensemble est celui qui donne le meilleur taux de classification.

2.1.2 Les mesures d'information

Ce type de critère d'évaluation représente le gain d'information pour un paramètre donné. Il est calculé en utilisant la probabilité a posteriori. Donc les paramètres qui donnent des gains d'information plus importants seront sélectionnés.

2.1.3 Les mesures de consistance

Ce type de critère détermine si un ou plusieurs paramètres sont nécessaires ou pas pour faire la classification des données. Donc, le sous-ensemble final contient uniquement les paramètres indispensables à la classification.

2.1.4 Les mesures de dépendance

Dans ce type de critère, les paramètres sont sélectionnés soit en calculant le niveau de dépendance des attributs les uns par rapport aux autres, soit en calculant le niveau de dépendance qui caractérise la corrélation entre un attribut ou un sous-ensemble d'attributs et une classe.

2.1.5 Les mesures de distance

Le but d'utiliser ce type de mesure est de trouver les paramètres qui maximisent la dispersion inter-classes pour que les données de différentes classes soient bien séparées et qui minimisent la dispersion intra-classe pour que les données de la même classe soient les plus compactes possible.

Selon le type de critère, on peut distinguer deux approches pour évaluer la pertinence d'un sous-ensemble de paramètres sélectionnés

2.2 Les méthodes filtres

Ces méthodes sont rapides par rapport aux autres méthodes [Guyon et Elisseeff, 2003]. Elles appliquent le critère d'évaluation uniquement sur les données sans l'utilisation de classifieur. Donc, le sous-ensemble final de paramètres est trouvé avant l'application de la phase d'apprentissage. Généralement, ces méthodes donnent un taux de reconnaissance faible puisqu'elles sont indépendantes du classifieur utilisé. Nous avons proposé un algorithme de ce type dans le deuxième chapitre.

2.3 Les méthodes enveloppantes

Les méthodes enveloppantes, connues en anglais wrappers [Kittler, 1978], sont des méthodes récentes. Elles sont plus simples que le premier type. Elles utilisent le taux d'erreur de classification comme critère d'évaluation. Donc, le classifieur fait partie de la procédure de sélection de paramètres. Il est donc évident que le sous-ensemble choisi dépend du classifieur utilisé. Alors, si le type de classifieur est mal choisi, il peut conduire à une mauvaise sélection de paramètres. L'inconvénient majeur de ces méthodes est qu'on doit refaire la classification des données pour chaque sous-ensemble de paramètres, ce qui demande un long temps d'exécution particulièrement si le nombre de paramètre est élevé.

2.4 Les méthodes Intégrées :

Quelques ouvrages citent un troisième type de méthodes qui sont les méthodes intégrées "embedded" [Boz, 2002]. Elles sont des méthodes hybrides qui combinent les avantages des méthodes filtres et enveloppantes. Elles sélectionnent les paramètres pendant la phase d'apprentissage. Le sous-ensemble final de paramètres est choisi en optimisant le critère d'évaluation utilisé. Initialement, plusieurs sous-ensembles de paramètres sont sélectionnés en appliquant une méthode filtre. Ensuite, un taux de classification est calculé pour chaque sous-ensemble. Donc, le résultat final est le sous-ensemble de paramètres qui a donné le meilleur taux de classification.

3 La procédure de génération

En pratique, le processus de sélection de paramètres se compose d'une séquence d'itérations. Pendant chaque itération, on utilise un sous-ensemble de paramètres différent qui est généré par une procédure de génération. Ce sous-ensemble est évalué en utilisant l'un des critères précédemment présentés. Le résultat de cette évaluation est utilisé pour la génération d'un nouveau sous-ensemble qui sera utilisé dans l'itération suivante. L'ensemble de paramètres utilisé dans la première itération peut être vide ou contient tous les paramètres ou bien un ensemble de paramètres générés aléatoirement.

Si le nombre de paramètres qui caractérise un système de production est d , alors le nombre de sous-ensembles qui peuvent être générées par la procédure de génération est $2^d - 1$. En pratique, il est très difficile d'évaluer tous les sous-ensembles quand le nombre d est très grand.

On peut distinguer trois types de méthodes de génération de paramètres qui seront expliquées brièvement dans les points ci-après.

3.1 La génération complète

Si on utilise une procédure de génération complète alors on doit appliquer une procédure de recherche exhaustive. Pour réduire le nombre de sous-ensembles évalués, on peut utiliser des heuristiques. La simple idée est d'appliquer un processus de backtracking qui permet de revenir en arrière si la procédure commence de générer des sous-ensembles avec des taux moins intéressants. On peut citer comme exemple de méthodes qui utilisent une génération complète la méthode Branch and Bound [Narendra et Fukunaga, 1977]. Cette méthode est basée sur la création d'un arbre de solution et la connaissance de nombre de paramètres du sous-ensemble final. Les nœuds de cet arbre représentent des sous-ensembles dans lesquels

quelques paramètres sont éliminés. La valeur maximale est initialisée en évaluant le nœud racine. Ensuite, cette valeur est mise à jour si l'évaluation d'un nouveau nœud donne un meilleur résultat. L'inconvénient majeur de cette méthode est qu'on ne sait pas la taille du sous-ensemble final. La solution pour ce problème est de fixer un certain seuil. Somol et ces collègues ont proposé une variante de Branch and Bound. Leur algorithme est plus rapide et utilise un système de pondération basé sur une fonction de prédiction. Cet algorithme offre la possibilité d'avoir un compromis entre rapidité d'exécution et qualité de solution. Il existe une autre proposition qui utilise une méthode séquentielle afin de composer une solution initiale [Somol et al., 2001].

3.2 La génération aléatoire

Cette méthode est basée sur une génération aléatoire de sous-ensembles. La construction de solutions ne se base pas sur l'ajout ou l'enlèvement de paramètres. Cette méthode de recherche ne s'arrête pas même si un optimum local est trouvé. Donc, on doit fixer un nombre maximal d'itérations comme un critère d'arrêt pour terminer la recherche et limiter le temps de calcul. Elle appartient à l'ensemble des méthodes non optimales car elle ne recherche pas tous les sous-ensembles de paramètres du système étudié. Ce type de méthodes converge généralement vers un optimum local, pour des problèmes mal définis ou multimodaux. On peut citer un exemple de ces méthodes les algorithmes génétiques (AG) développés par Yang et Honavar, qui présentent des qualités intéressantes pour la résolution de problèmes de sélection de paramètres [Yang et Honavar, 1998]. Le principe général d'un algorithme génétique peut être résumé dans le tableau 4.1 :

TABLEAU 4.1 : PRINCIPE GENERALE D'UN ALGORITHME GENETIQUE

- Choix du codage
 - Génération aléatoire de la population initiale
 - Evaluation : calcul de la fonction d'adaptation (critère)
 - **Répéter**
 - Sélection
 - Croisement
 - Mutation
 - Evaluation
 - **Jusqu'à** satisfaction du critère d'arrêt
-

Initialement, on doit représenter les différents sous-ensembles possibles de paramètres sous une forme utilisable pour un algorithme génétique. Généralement, il existe trois types de codage qui sont le codage binaire, le codage réel et le codage en base n. Dans le cadre de notre application, nous avons utilisé un codage binaire.

La population initiale est composée de P sous-ensembles. L'initialisation de ces sous-ensembles est réalisée d'une façon aléatoire. Les valeurs du gène (0 ou 1) sont choisies aléatoirement selon une distribution uniforme. Ces valeurs indiquent la présence ou l'absence d'un paramètre dans le sous-ensemble en question. Le nombre de sous-ensembles (individus) doit être fixé afin de trouver un compromis entre le temps d'exécution et la qualité du résultat. Chaque individu appartenant à la population initiale est un vecteur de d gènes qui représente le nombre total de paramètres. Ce vecteur a la forme suivante:

$$V_k = (g_{k1}, g_{k1}, \dots, g_{kj}, \dots, g_{kd}) \text{ avec } g_{kj} = \begin{cases} 1 & \text{si } \alpha_j \text{ est présent} \\ 0 & \text{sinon} \end{cases} \quad (4.1)$$

Chaque gène g_{kj} code la présence ou l'absence du paramètre α_j dans le vecteur V_k .

Le vecteur final donnera simultanément les meilleurs paramètres discriminants et un nombre d' qui représente la taille du sous-ensemble final. Les algorithmes génétiques utilisent une fonction d'adaptation afin d'évaluer la pertinence de chaque individu. On peut utiliser comme une fonction d'adaptation pour l'algorithme génétique la fonction suivante :

$$F(V_k) = \text{trace}(S_{Wk}^{-1} \cdot S_{Bk}).$$

A la fin de chaque itération, on applique une fonction de sélection qui ajoute les meilleurs éléments à la génération suivante. Pour ce faire, Elle utilise la fitness de chaque individu. Une autre fonction qu'on doit appliquer est le Croisement. L'étape de croisement au sein d'un algorithme génétique est capitale puisque elle permet d'explorer des nouvelles espaces de recherche. Elle se base sur des échanges de gènes entre deux solutions précédentes, pour créer deux nouvelles solutions. Un autre opérateur important est la mutation. Le but de cet opérateur est de créer la diversité dans la population. Sans l'utilisation de cet opérateur, les AG vont créer que de clones de la première solution trouvée. Donc, La mutation sert à générer des solutions non proposées par le croisement ou la sélection. Son principe est de remplacer la valeur d'un gène sélectionné. Par exemple dans un codage binaire, on remplace la valeur 1 par la valeur 0 et vice versa. On applique cet opérateur sur un nombre limité de paramètres pour ne pas influencer la convergence de l'algorithme vers la solution optimale. Les opérateurs de

croisement et de mutation peuvent créer des solutions non valides (contient que des 0 ou des 1). La solution est d'utiliser une procédure qui contrôle les solutions créées et réexécute la procédure de génération jusqu'à l'obtention d'une solution valide. Pour ne perdre les meilleures solutions trouvées dans les premières itérations, on doit les recopier toujours dans la génération suivantes. On peut utiliser comme un critère d'arrêt l'évolution de l'algorithme. Par exemple, si une itération ne donne pas une meilleure solution par rapport aux solutions trouvées dans les itérations précédentes alors l'algorithme est stoppé. Pour ne pas arrêter l'algorithme rapidement, on peut fixer un nombre minimal d'itérations qu'on doit exécuter. Cette dernière action peut permettre aussi d'éviter de tomber dans un minima local.

3.3 La génération séquentielle

Ce type de méthode se base sur l'ajout ou la suppression d'un ou plusieurs paramètres au fur et à mesure des itérations. Il existe deux types de méthodes de génération séquentielle :

3.3.1 La méthode de génération constructive

Cette méthode est connue en anglais sous le nom "Sequential Forward Selection" (SFS) [Blum et Langley, 1997]. Initialement, elle utilise un ensemble vide de paramètres. Ensuite, elle ajoute des paramètres à chaque itération. Les paramètres ajoutés sont ceux qui optimisent la fonction d'évaluation. Généralement, cette méthode donne une solution différente de la solution optimale trouvée par une recherche exhaustive.

3.3.2 La méthode de génération destructive

La méthode de génération destructive est connue en anglais sous le nom "Sequential Backward Selection" (SBS) [Kittler, 1978]. Elle utilise un principe inverse de la méthode précédente. Elle commence d'un ensemble qui contient tous les paramètres. Ensuite, à chaque itération elle élimine le paramètre le moins pertinent selon la fonction d'évaluation. Celui qui a provoqué une mauvaise séparabilité entre classes. Cette méthode présente l'avantage d'être très simple mais elle ne permet pas d'explorer tous les sous-ensembles de paramètres possibles et aussi elle est très lente par rapport à la procédure SFS. Il existe des variantes de cette méthode qui consistent à ajouter plusieurs paramètres à la fois à chaque itération pour accélérer la recherche. Cette méthode n'est pas optimale puisqu'elle ne permet pas de revenir en arrière et d'ajouter des paramètres déjà supprimés. Il existe des approches qui combinent les principes de SFS et SBS. On cite par exemple : L'algorithme SFFS (Sequential Forward Floating Selection) qui permet d'utiliser après chaque opération Forward des opérations

Backward pour que le sous-ensemble de paramètres correspondant améliore la fonction d'évaluation. Un autre exemple est l'algorithme SBFS (Sequential Backward Floating Selection) qui utilise le même principe mais il inverse les deux étapes [Boz, 2002].

4 Critère d'arrêt

Le critère d'arrêt est un élément essentiel de la procédure de sélection de paramètres. Il fixe le nombre d'itérations et par conséquent la taille du vecteur optimal qui n'est pas connue a priori. On peut distinguer trois types de critères d'arrêt :

- Critère d'arrêt basé sur l'évaluation : ce type de critère dépend des valeurs retournées par la fonction d'évaluation. Il calcul la différence entre les valeurs des deux dernières itérations. Si le résultat obtenu est nul ou négatif alors on peut arrêter la procédure de recherche. Dans une approche Wrapper, on peut utiliser le rapport qui existe entre le taux de classification et la taille du vecteur optimal. Ce type de critère a été utilisé par plusieurs chercheurs, on cite par exemple : Vandembroucke dans la sélection des paramètres de texture couleur pertinents et Maenpaa dans la classification d'images de texture couleur [Vandembroucke et al., 2000].
- Critère d'arrêt basé sur le paramètre sonde : ce type de critère est proposé par Stoppiglia. Il choisit un paramètre parmi les paramètres initiaux. Ensuite, si ce paramètre est sélectionné alors la procédure de recherche est arrêtée [Stoppiglia, 1997].
- Critère basé sur les faux paramètres. Ce type de critère est créé par Tusher. Il utilise des combinaisons aléatoires des valeurs des paramètres initiaux. Ensuite, tous les paramètres sont utilisés par une procédure de recherche constructive. La procédure est arrêtée quand le rapport entre le nombre de faux paramètres et vrais paramètres atteint un certain seuil [Tusher et al., 2001].

5 La validation

Cette étape ne fait pas partie de la procédure de sélection de paramètres. Elle a pour but de vérifier la validité du sous-ensemble de paramètres sélectionnés en réalisant plusieurs évaluations sur des bases de données mesurées à partir du système étudié ou générées artificiellement en utilisant des fonctions mathématiques. Pour réaliser cette étape, on divise l'ensemble de données en deux sous-ensembles (apprentissage et test). On utilise le premier

pour réaliser l'apprentissage du modèle. Ce sous-ensemble est composé de données étiquetées (données avec leurs classes). Dans le deuxième sous-ensemble qui est composé de données de test, on ne connaît pas les classes de ces données. Suivant la répartition des données entre l'ensemble d'apprentissage et l'ensemble de test, on peut distinguer plusieurs approches de validation [Kalakech, 2011].

- La méthode Holdout : la moitié de données est utilisée pour l'apprentissage et l'autre moitié est utilisée pour le test.
- La méthode de resubstitution : le même ensemble est utilisé dans la phase d'apprentissage et dans la phase de test.
- La méthode p-validation croisée : la base de données est partitionnée en p sous-ensembles de même taille. Ensuite, on exécute la procédure de validation p fois. A chaque fois, on utilise un seul sous-ensemble et les autres sous-ensembles sont utilisés comme des données de test.
- La méthode du Leaving-one-out est un cas particulier de la méthode p-validation croisée où l'ensemble des données est divisé en n parties. L'ensemble de test comprend alors à chaque fois une seule donnée. Cette méthode est coûteuse en temps de calcul.

Pour réaliser cette étape, il existe plusieurs classieurs qu'on peut utiliser. On cite par exemple : k-plus proches voisins, réseaux de neurones ou SVM.

6 Extraction des paramètres

6.1 Principe

L'extraction des paramètres consiste à créer des p nouveaux paramètres à partir des d paramètres initiaux tel que $P < d$. les p nouveaux paramètres sont des projections des d paramètres initiaux dans un espace réduit. Les p paramètres contiennent les informations pertinentes que les d paramètres peuvent fournir.

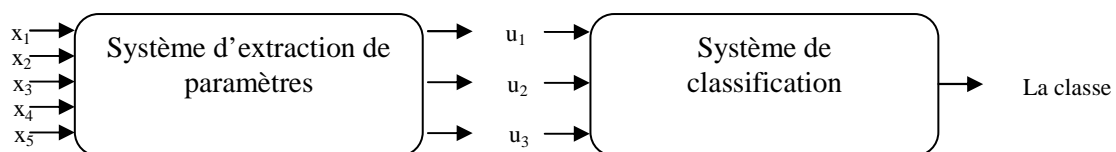


Fig. 4.3 : Principe d'extraction de variables [Guérif, 2006]

Il existe quatre types de méthode d'extraction qui sont [Kalakech, 2011] :

- Les méthodes linéaires supervisées, par exemple l'Analyse Factorielle Discriminante (AFD) ;
- Les méthodes linéaires non-supervisées, par exemple l'Analyse en Composantes Principales (ACP);
- Les méthodes non linéaires supervisées comme l'Analyse Factorielle Discriminante à noyau ;
- Les méthodes non linéaires non supervisées, par exemple Linear Embedding (LLE).

L'inconvénient majeur de ces méthodes est leur temps de calcul. Elles posent aussi un problème de difficulté dans l'interprétation des nouveaux paramètres par l'utilisateur. Dans le point suivant, on va présenter la méthode la plus utilisée en extraction de paramètres qui est l'Analyse en Composantes Principales.

6.2 L'analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP) est une ancienne méthode qui calcule les axes de projection d'un nuage de points [Pearson, 1901]. Elle utilise ces axes de projection pour dessiner des directions orthogonales suivant lesquelles la variance des données est maximisée. Donc, On peut utiliser ACP pour l'extraction des données en gardant simplement un sous-ensemble des composantes principales. Pour simplifier l'interprétation de la projection de données imposée par cette méthode, plusieurs recherche dans ce sens ont conduit à la création de quelques méthodes telles que : varimax, quartimax et equamax qui utilisent une rotation orthogonale des axes et les méthodes oblimin et promax qui utilisent des rotations obliques [Guérif, 2006]. L'inconvénient majeur de la méthode ACP est qu'elle ne permet pas de résoudre les problèmes non linéaires puisqu'elle utilise uniquement les relations linéaires qui existent entre les paramètres et ne peut donc pas fournir une projection fidèle pour une distribution non-linéaire. Il existe des variantes de cette méthode qui ont été développées pour résoudre le cas des non linéaire [Kadri et al., 2012a]. Ces variantes utilisent de noyaux de Mercer de la même façon que dans les SVMs (voir le chapitre précédent).

7 Procédure de sélection proposée

Dans les sections précédentes, nous avons présenté les éléments essentiels d'une procédure de sélection et nous avons montré l'intérêt d'utiliser ce type de procédure dans le processus de diagnostic. Nous proposons dans cette section un algorithme hybride de sélection de paramètres basé sur l'algorithme Filter based ACO et les séparateurs à vaste marge (SVM)

[Kadri et al., 2012b]. Notre algorithme sélectionne un sous-ensemble réduit de paramètres depuis un ensemble qui contient un grand nombre de paramètre en utilisant la méthode Wrapper (la figure 4.4).

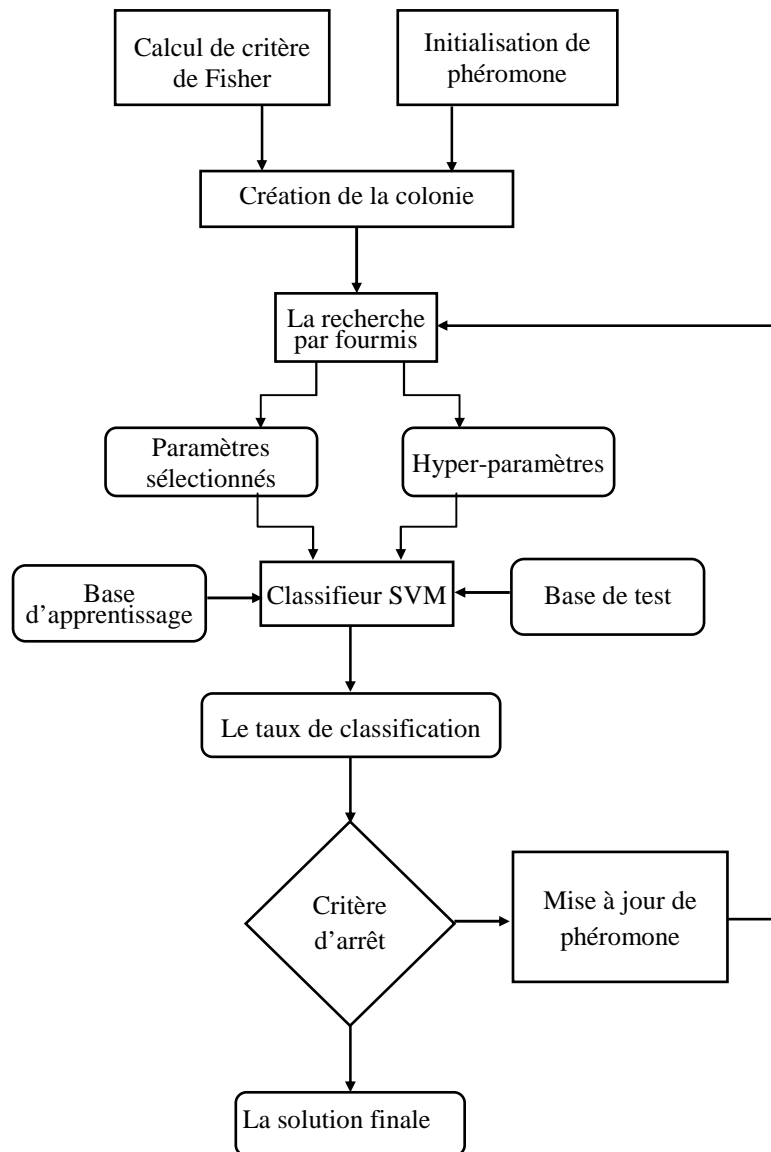


Fig. 4.4: Organigramme de l'algorithme Hybrid wrapper/filter-based ACO-SVM

L'utilisation d'un algorithme de colonie de fourmis a imposé le choix d'une méthode de génération aléatoire. Nous avons utilisé le rapport de Fisher comme un facteur heuristique pour pousser les fourmis à choisir les paramètres pertinents (voir la section 6.3 dans le deuxième chapitre). L'algorithme Hybrid wrapper/filter-based ACO-SVM utilise aussi un facteur phéromonal qui représente la quantité de phéromone disposée par la colonie sur un arc. Ce facteur favorise le choix d'un paramètre en se basant sur l'historique de la colonie. Le taux d'erreur de la classification est utilisé comme un critère d'évaluation. Il est calculé après une classification réalisée en utilisant un classifieur SVM. Ce classifieur utilise un noyau

Gaussien. Nous avons utilisé le facteur phéromonal de l'algorithme Hybrid wrapper/filter-based ACO-SVM pour choisir les valeurs des hyper-paramètres C et γ du noyau. Les valeurs possibles des hyper-paramètres sont définies dans des tableaux d'une seule dimension.

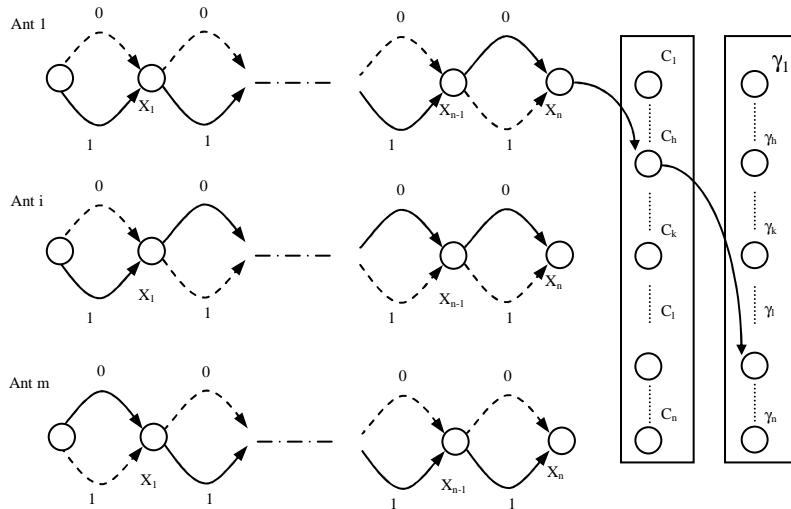


Fig. 4.5: Les solutions finales proposées par l'algorithme Hybrid wrapper/filter-based ACO-SVM

On peut résumer l'algorithme dans les points suivants :

1. Initialiser la phéromone sur le réseaux
2. Calculer le facteur heuristique en utilisant la formule (2.9).
3. Lancer la procédure de recherche en utilisant la probabilité de déplacement définie par la formule (2.8).
4. Evaluer les solutions trouvées par Hybrid wrapper/filter-based ACO-SVM en utilisant le classifieur SVM et garder la solution optimale;
5. Mise à jour la phéromone sur le réseau en utilisant la formule (2.12);
6. Tester si le critère d'arrêt est atteint. si oui, alors la procédure est terminée sinon aller à l'étape 3.

8 Conclusion

Dans ce chapitre, nous avons discuté le problème de la sélection de paramètre et sa position dans le processus de diagnostic. Nous avons présenté en détail les deux approches utilisées pour la réalisation de la tâche de réduction de la dimension du vecteur d'état qui sont la sélection de paramètres et l'extraction de paramètres. Rappelons que notre thèse s'inscrit dans

le cadre du diagnostic des systèmes dynamiques et complexes. Donc, nous avons traité ce problème puisqu'il représente la première étape dans le processus de diagnostic industriel. Nous n'avons pas utilisé une approche basée sur l'extraction de paramètres à cause de la complexité qui caractérise ce type de techniques lorsque le nombre de paramètres soit relativement grand et aussi à cause de la difficulté de l'interprétation de paramètres créés. Notre approche utilise la technique de sélection de paramètres pour réduire la taille du vecteur d'état. Elle n'impose pas des efforts supplémentaires pour l'interprétation de paramètres sélectionnés. Nous avons utilisé un classifieur robuste pour faire l'évaluation du sous-ensemble de paramètres sélectionnés. Notre approche permet aussi la sélection des valeurs des hyper-paramètres du classifieur SVM les plus adaptés aux données du système étudié.

Dans le chapitre suivant, nous nous proposons d'appliquer les approches développées pour faire le diagnostic de deux systèmes industriels.

APPLICATION ET VALIDATION DES APPROCHES PROPOSEES

Résumé :

Dans ce chapitre, Nous avons réalisé deux types de classification non supervisée à partir des bases de données construites lors de l'étape d'acquisition. Pour exposer l'efficacité et les avantages de chacun des deux algorithmes (AntTreeStock et LF), nous avons répéter l'exécution de ces algorithmes plusieurs fois en utilisant plusieurs bases de données. A la fin de ce chapitre, Nous intéressons à la sélection de paramètres en utilisant un classifieur SVM. Toujours dans la même section, nous allons discuter notre choix de la fonction noyau en comparant les performances de plusieurs noyaux.

1 Introduction

Dans ce chapitre, nous allons mettre en œuvre et évaluer l'efficacité des algorithmes de classification AntTreeStoch et Lumer & Faieta proposés dans le chapitre 2 et l'algorithme de sélection de paramètres Hybrid wrapper/filter-based ACO-SVM présenté dans les chapitres 2 et 4. Les expérimentations sont effectuées en utilisant des données simulées et des données réelles qui représentent deux systèmes industriels dynamiques. Le premier système est un four rotatif de SCIMAT (Société des ciments de Aïn Touta) qui sera présenté au début de ce chapitre. Il assure la phase de clinkérisation dans le processus de fabrication du ciment. Ensuite nous allons détailler le deuxième système qui est un échangeur à plaques de ULA (unité de laiterie Aurès). Cet appareil est pour but la pasteurisation du lait qui représente une étape indispensable dans le processus de production de lait.

2 Description du premier procédé

2.1 Présentation de SCIMAT

L'origine du Groupe industriel et commercial des ciments de l'Est remonte à la restructuration de la société mère SNMC (Société nationale des matériaux de construction) qui a donné naissance à GICA. Cette entreprise fut transformée, à son tour, en entreprise publique économique sous la forme de société par actions au capital social de 150 millions de DA, en 1990. En 1998 et après plusieurs recapitalisations, GICA devient un groupe industriel et commercial au capital social de 15 milliards de DA.

Cette cimenterie d'une capacité de 2.500.000 t/an (2 fours) utilise le procédé par voie sèche. Elle est composée de plusieurs unités. Les carrières d'où sont extraites les matières premières (argile et calcaire). Les stations de concassage destinées à réduire la taille des blocs de matières (calcaire et argile 1000 t/h, les ajouts 150 t/h). L'atelier de broyage de cru est composé de deux broyeurs de 140 t/h chacun. L'atelier de cuisson regroupe deux fours dont le débit clinker est de 1560 t/j. Le broyage de ciment comprend deux broyeurs de 100 t/h chacun. Les expéditions du ciment s'effectuent à partir de deux stations, une pour les camions et une pour les wagons [Mouss, 2006].

3 La fabrication du ciment

La fabrication du ciment consiste à faire broyer et refroidir un produit appelé clinker. Ce dernier se compose d'un mélange de 80% de calcaire (essentiellement CaCO_3) et 20% d'argile (dont 14% de SiO_2 , 4% de Al_2O_3 et 2% de Fe_2O_3) porté à 1450 °C, température de fusion. Le ciment est nommé aussi liant hydraulique, puisqu' il forme avec l'eau une pâte plastique liante, capable de réunir, en durcissant, des matières variées [Ogbeide, 2010]. La figure 5.1 présente les différentes étapes pour la fabrication de ciment.

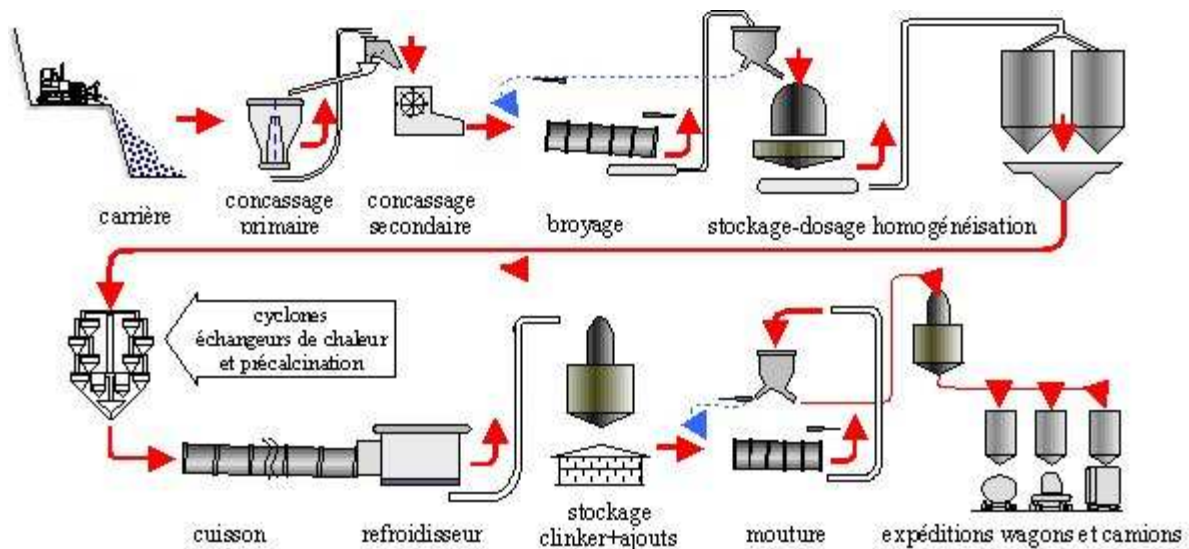


Fig. 5.1: Représentation schématique du principe de fabrication du ciment

3.1 La Clinkérisation

Selon plusieurs chercheurs, l'atelier de cuisson représente le sous-système central dans le processus de fabrication du ciment. Dans le cas de la SCIMAT, cet atelier est composé de deux lignes identiques et complètement indépendantes. Chaque ligne est constituée de trois échangeurs : le préchauffeur à cyclones, le four rotatif et le refroidisseur à ballons (la figure 5.2) [Rezki et al., 2011].

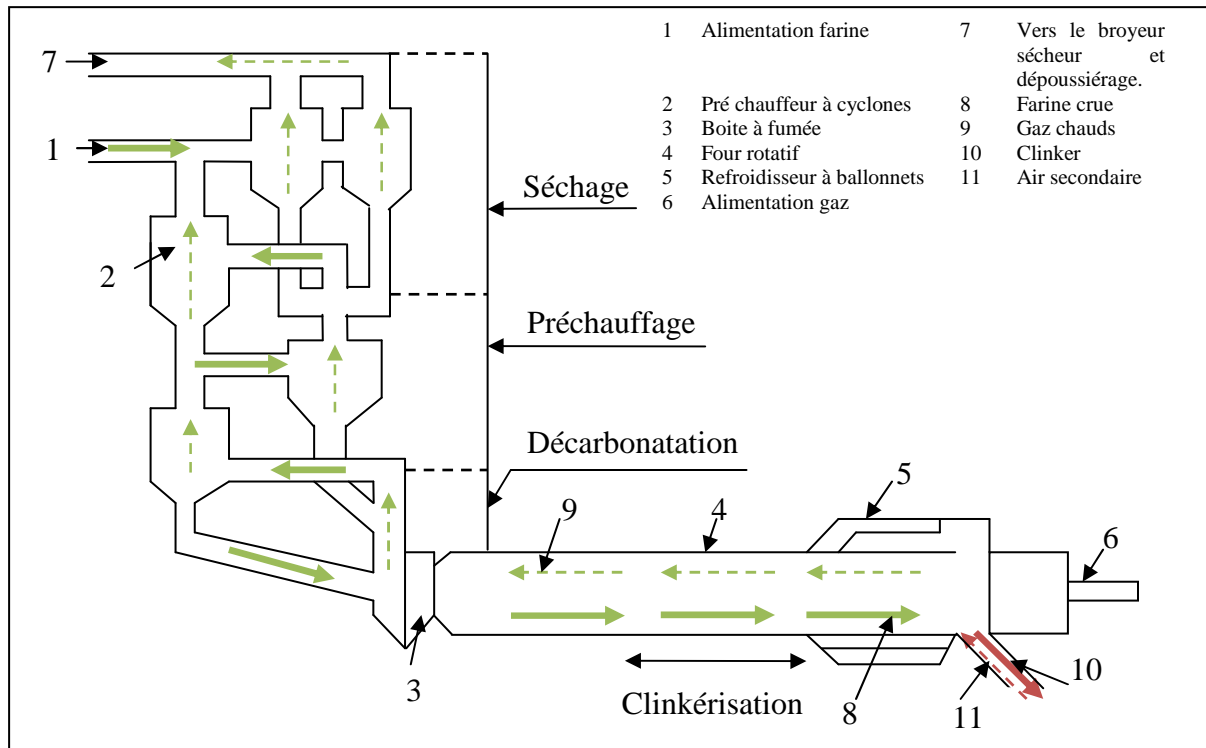


Fig. 5.2 : installation de cuisson

3.1.1 Le four rotatif

Le four rotatif de la cimenterie est constitué d'un cylindre d'acier dont la longueur est de 68 m et le diamètre est de 4,6 m. Il est recouvert à l'intérieur par des briques réfractaires qui protègent les tôles des températures élevées (1850°C pour les gaz et 1450°C pour la matière). Il est légèrement incliné (3%) et il tourne sur lui-même à une vitesse comprise entre 0,67 et 2 tours par minute. Ses caractéristiques permettent une progression lente du cru vers la partie basse, à la rencontre de la flamme et à contre courant des gaz chauds. Le clinker se fait à une température proche de 1450°C. Cette température est obtenue grâce à une flamme, située à l'extrémité basse du four, alimentée par du charbon broyé, fuel lourd, gaz, ou encore en partie avec des combustibles de substitution provenant d'autres industries, tels que le coke de pétrole, les pneus usagés, les farines animales, les huiles usagées. Suite à la réaction chimique se produisant à l'intérieur du four, la farine laisse place au clinker que l'on doit faire refroidir [Phoummavong, 2012] [Mouss, 2006].

3.2 Analyse de fonctionnement d'un four rotatif

Le four rotatif est soumis à des sollicitations ayant pour origine les charges mécaniques mises en œuvre ou appliquées, la charge thermique, des défauts de géométrie.

Les charges mécaniques, mises en œuvre ou appliquées, peuvent être :

- Uniformément réparties : poids propre de la virole, poids du réfractaire, poids de la matière ;
- Concentrées : poids de la couronne, poids sur les extrémités du four (joints, satellites), force et couple d'entraînement ;
- Evolutives : croûtage, anneaux.

Les sollicitations dues à la charge thermique sont :

- La variation de température sur la longueur du four ;
- Une distribution de température non homogène dans une section transversale.

Ces sollicitations engendrent des contraintes et des déformations qui sont évaluées par le calcul lors de la conception, de manière à rester dans des limites acceptables.

On peut citer comme défauts de géométrie :

- Les alignements imparfaits des supports (horizontal et vertical) ;
- La construction défectueuse au montage (axe de virole non rectiligne) ;
- Les déformations permanentes créées par des surchauffes.

La durée de vie des éléments importants du four est calculée en années, sur une base de 300 jours de marche par an, 24 h /24, soit 7 200 h. Les bandages peuvent subir des casses et fissures entre 0 et 15 ans de service (8 % des cas). Au-delà de 15 ans, une dégradation par écaillage apparaît dans 5 % des cas tandis que le risque de casse ou fissure est réduit à 2 %. Globalement 85 % des bandages ont une durée de vie de 25 ans. Les viroles courantes ont des durées de vie de l'ordre de 20 à 30 ans ; mais des phénomènes de corrosion de la virole, liés à l'usage de matières premières ou combustibles particuliers, peuvent apparaître et conduire en quelques années à des remplacements de tronçons de virole. De même, une conduite de four, sans ménagement, engendre des déformations permanentes irréversibles que l'on peut assimiler à des accidents. Ces déformations ne seront supprimées que par remplacement des viroles. En outre, la virole de sortie de four est très sollicitée et peut être quelquefois remplacée dans les 3 à 5 ans [Phoummavong, 2012].

3.3 Les paramètres de la partie Clinkérisation

Plusieurs paramètres peuvent caractériser le fonctionnement de cette partie du système. Ces paramètres décrivent les différents phénomènes complexes que nous avons présentés dans les sections précédentes. Dans le cas de notre système, chaque composant possède plusieurs modes de dysfonctionnement causés par le changement de la valeur d'un ou de plusieurs paramètres. Donc, le système de diagnostic utilisé doit être capable de détecter le ou les paramètres qui ont causé l'état de dysfonctionnement du composant.

Le tableau 5.1 présente le vecteur de paramètres qui caractérise mieux les différents états de fonctionnement et de dysfonctionnement du système selon l'avis de l'expert. Grâce à ce vecteur, on peut connaître par exemple : la qualité de clinker, le niveau de production et la consommation d'énergie [Rezki et al., 2012].

Tableau 5.1 : Principaux paramètres de la partie Clinkérisation

Sous système			Variable code air	Point de consigne	Désignation	
C U I S S O N / C L I N K É R I S A	Préchauffeur à cyclone	A50	O ₂ C	0.1<O ₂ C<0.6%	Teneur O ₂ sortie cyclone A50	
			COC	0<COC<0.25%	Teneur CO sortie cyclone A50	
			A50T1	300<A50T1<350°C	Température gaz sortie cyclone A50.	
			A50P1	40<A50P1<60mbar	Pression sortie cyclone A50.	
		A52	A52T1	530<A52T1<555°C	Température gaz sortie cyclone A52.	
			A52P2	25<A52P2<35mbar	Pression cyclone A52.	
		A53	A53T1	700<A53T1<760°C	Température gaz cyclone A53	
			A53T2	700<A53T2<720°C	Température matière cyclone A53	
			A53P1	18<A53P1<20mbar	Pression cyclone A53.	
		A54	A54T1	820<A54T1<860°C	Température gaz cyclone A54	
			A54T2	820<A54T2<830°C	Température matière cyclone A54	
			A54P2	8<A54P2<10mbar	Pression cyclone A54t	
		Boite à fumée		O ₂ F	0.1<O ₂ F<0.6%	Teneur O ₂ boite à fumée.
				COP	0<COP<0.25%	Teneur CO boite à fumée.

T I O N		W01P3	$1 < W01P3 < 2 \text{ mbar}$	Pression sortie boîte à fumée.
		W01P1	$0.5 < W01P1 < 1 \text{ mbar}$	Pression entrée boîte à fumée.
		W01T1	$1200 < W01T1 < 1400^\circ\text{C}$	Température gaz entrée boîte à fumée.
		V31F1	$850 < V31F1 < 893 \text{ Nm}^3/\text{h}$	Début gaz.
	Four	V01F1	$3280 < V01F1 < 5056 \text{ Nm}^3/\text{h}$	Début gaz.
		CCS	$810 < CCS < 1050$	Consommation calorifique spécifique.
		W01X1	$20 < W01X1 < 60\%$	Moment four
		W01S1	$1 < W01S1 < 3 \text{ t/mm}$	Vitesse four.
		W01T4	$1200 < W01T1 < 1500^\circ\text{C}$	Température de cuisson
		TV	$1000 < TV < 1800^\circ\text{C}$	Température virole four
		V07P1	$28 < V07P1 < 30 \text{ mbar}$	Pression air primaire
		PAL	$1250 < PAL < 1350 \text{ g/l}$	Poids au litre
		FCAO	$1 < FCAO < 3\%$	Chaux libre
		Refroidisseur	U01T1	$100 < U01T1 < 150^\circ\text{C}$
K01P1	$0.5 < K01P1 < 1 \text{ mbar}$		Pression air secondaire	
K01T1	$500 < K01T1 < 900^\circ\text{C}$		Température air secondaire	

3.3.1 Modes de fonctionnement étudiés

Une grande partie des informations concernant les différents modes de fonctionnement est obtenue par une analyse de type *Analyse des modes de défaillance, de leurs effets et de leur Criticité* (AMDEC). L'AMDEC est une technique d'analyse exhaustive et rigoureuse de travail en groupe, très efficace par la mise en commun de l'expérience et de la compétence de chaque participant du groupe de travail. Cette méthode fait ressortir les actions correctives à mettre en place. Dans notre cas, elle a été réalisée par le groupe du travail composé du chef de production et l'ensemble de travailleurs de la salle de contrôle [Mouss, 2006]. A partir de l'AMDEC (voir annexe A), nous pouvons tirer quatre modes de défaillance qui sont présentés sur le tableau 5.2 :

Tableau 5.2: les différentes classes de fonctionnement

N° classe	1	2	3	4
Nom classe	Marche perturbée	Déplacement de zone	Production des incuits	Perte de calories (l'énergie)

4 Description du deuxième procédé

Dans cette deuxième partie, nous nous intéressons à un autre secteur industriel qui est l'industrie agro-alimentaire. Nous avons pris comme cas d'étude la Laiterie des Aurès et nous avons essayé d'appliquer nos propositions sur le sous-système de pasteurisation. Donc, l'objectif de cette section est d'étudier ce sous-système pour identifier ses différents modes de défaillance et tous les paramètres qui ont une influence sur l'état de fonctionnement du système. A fin de parvenir à ce but, nous avons construit son AMDEC [Kadri et al., 2011a].

4.1 Présentation de laiterie des Aurès

L'unité de laiterie Aurès Batna est une branche du Groupe Industriel des Productions Laitières GIPLAIT (groupe industriel de la production du lait). GIPLAIT a été créé le 10 Mai 1998, à l'issue de la restructuration des Ex. Offices Régionaux (ORLAC, OROLAIT, ORELAIT). Doté d'un capital social de 2,501 Milliards de Dinars, il a pour mission la production et la commercialisation des laits et des produits laitiers. Son CA avoisine les 20 Milliards de Dinars et son effectif est de 4.099 salariés, dont 3.741 permanents. Les capacités de production installées sont de l'ordre de 1,4 Milliards de litre de lait équivalent. La Société mère située à Alger, est gérée par un Directoire et Quatre Directions Centrales (Administration et Ressources Humaines, Finances et Comptabilité, Développement et Restructuration, Contrôle de Gestion). GIPLAIT regroupe dix-neuf (19) usines de production en activité, érigées en Filiales et une société de *trading* chargée des importations pour assurer l'approvisionnement de ses différentes usines en matières premières principales. L'unité Aurès, créée en 1991, se trouve dans la zone industrielle Kechida. Doté d'un capital social de 250.000.000 DA, La laiterie Aurès fonctionne par roulement de huit heures consécutives, trois équipes de travail afin d'assurer un fonctionnement continu sur les 24 h. Ce système d'organisation d'horaires de travail a permis une production journalière d'environ 220.000 litres de produits laitiers. Les divers produits élaborés se répartissent dans les catégories suivantes: 1) les laits : pasteurisé (lait de consommation), stérilisé, fermenté (leben), 2) les yaourts : étuvé, brassé, 3) la pâte fraîche (fromage), et l'huile de beurre (smen) [Kadri et al., 2012a].

La laiterie contient plusieurs ateliers qui sont : l'atelier de réception de lait cru, l'atelier de préparation du lait, l'atelier des produits laitiers, l'atelier de conditionnement de LPC, l'atelier de conditionnement de PL, l'atelier de CIP et le laboratoire du contrôle de qualité.

4.2 Production du lait

Le lait est un produit très fragile. Sa durée de conservation est très courte. La grande demande en lait et l'implication des nouvelles technologies de transformation ont changé les modes de production, de transformation et de conditionnement du lait en vue de son approbation par les consommateurs. Dans le processus de production, on applique des traitements physiques sur le lait cru pour qu'il devienne un lait de consommation. Le processus de production englobe plusieurs étapes comme la clarification, la standardisation, l'homogénéisation et bien évidemment la pasteurisation [Office, 2001].

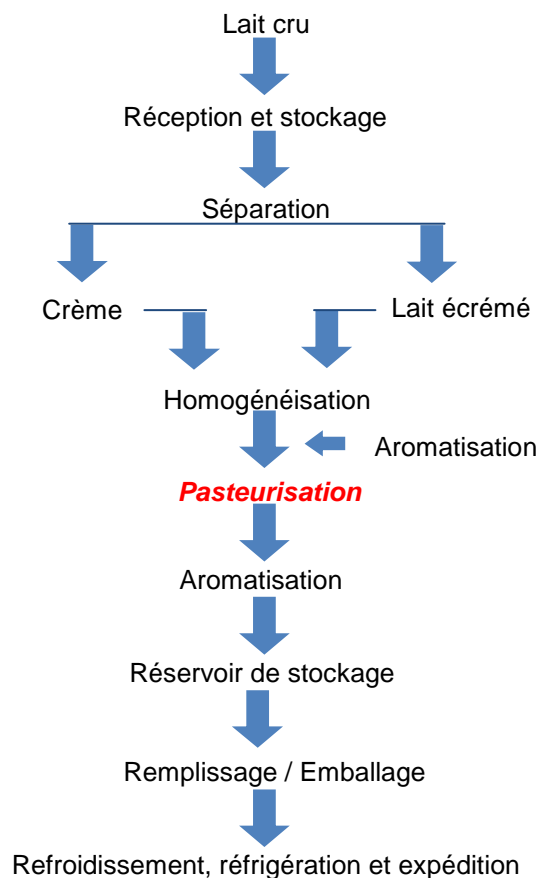


Fig. 5.3 : Le cheminement de lait dans le processus de fabrication

4.3 Pasteurisation

4.3.1 Principe

Dans tous les procédés de fabrication du lait, la pasteurisation constitue leur point commun. Elle consiste à appliquer un traitement thermique sur le lait pour faire tuer les microorganismes pathogènes et un grand nombre de microorganismes d'altération. Elle permet d'une part, d'assurer la salubrité du produit et d'autre part, d'améliorer sa

conservabilité. La pasteurisation est utilisée dans le processus de production de plusieurs produits comme le lait pasteurisé et le beurre pasteurisé. Dans cette étape, le lait est soumis à une température égale à 63°C pendant 30 minutes, ou 73°C pendant 16 secondes. Mais, pour avoir une longue conservation du lait pasteurisé, on doit utiliser un traitement plus rigoureux en température et en temps. Dans ce dernier type de traitement, il y a le risque de changer le goût du lait ou de diminuer sa valeur nutritive. La pasteurisation peut être aussi utilisée pour améliorer la saveur du lait en intégrant au pasteurisateur une chambre à vide qui permet d'extraire certains composés volatils. Aujourd'hui, la pasteurisation la plus utilisée est celle qui est basée sur un procédé en continu, à haute température (HTST : High Temperature, Short Time), au moyen de pasteurisateurs en plaques. Cette méthode présente l'avantage d'être la plus économique, précise et fiable [Office, 2001] [Mouss, 2006].

4.3.2 Le pasteurisateur

On utilise souvent un échangeur de chaleur à plaques comme pasteurisateur. Il est constitué d'un assemblage de plaques cannelées montées et serrées sur un bâti. Chaque paire de plaques forme un canal qui alterne pour le fluide et pour le produit. La distance entre deux plaques obtenue par des joints est variable. Elle peut aller de 3 à 7 mm pour les produits les plus visqueux. La surface d'échange est obtenue par la surface des plaques et par leur nombre. Ce type d'échangeur de chaleur a l'avantage de prendre moins de place au sol que les autres.

Le pasteurisateur utilisé fonctionne à grands débits et il est complètement automatisé. Il représente un appareil qui peut réchauffer ou refroidir en continu un produit pompable. Il est basé sur le principe de faire circuler parallèlement le produit à un fluide thermique. Ce dernier est généralement à contre-courant du produit. Il comporte plusieurs sections ainsi qu'un chambreur :

- Le chambreur : est un tube dont le volume permet, selon le débit, de maintenir la température de pasteurisation durant le temps nécessaire.
- Section de récupération de chaleur Préchauffage ou pré-refroidissement : dans cette section, le produit entrant est préchauffé et le produit sortant est pré-refroidi. Aucune énergie extérieure n'est nécessaire, ce qui permet d'économiser beaucoup d'énergie.
- Section de chauffage : cette section doit amener le produit préchauffé dans la section de récupération à la température de pasteurisation. Cette mesure de la température de pasteurisation doit être précise et constante.

- Section de refroidissement : une fois pré-refroidi dans la section de récupération de chaleur, le produit est refroidi dans cette section. Le produit final devra être constamment refroidi puisque la pasteurisation ne le rend pas stérile.

4.3.2.1 Le processus

Après une phase de prétraitement, le lait arrive au niveau du bac tampon « T01 ». Lorsqu'on débute une phase de pasteurisation, la pompe « P01 » est actionnée, elle aspire alors le lait pour l'envoyer à travers la vanne V2 vers l'échangeur de chaleur à plaques. Le lait ainsi chauffé arrive au niveau de la vanne « V5 » commandé par le contrôleur « R-Cont2 » qui envoie le lait en fonction de sa température (entre 90° et 95 °C) vers l'échangeur pour y être refroidit sinon vers le tank « T01 » pour un nouveau cycle de chauffage. Une fois refroidi, le lait arrive au niveau de la vanne «V7 » commandé par le contrôleur « R-Cont-3 ». Si la température du lait est comprise entre 4° et 6°C, le lait sera envoyé vers la vanne «V9 ». Dans le cas contraire, le lait est réinjecté pour un nouveau refroidissement. On peut résumer les différentes opérations de pasteurisation dans le tableau 5.3 [Office, 2001] [Mouss, 2006] [Kadri et al., 2012a]:

Tableau 5.3 : Les différentes opérations de pasteurisation

Opération	Désignation	But	Observation
OP1	Production d'eau chaude	Chauffer l'eau qui va à son tour chauffer le lait	La vapeur arrive au niveau de l'échangeur va chauffer l'eau
OP2	Chauffage du lait	Détruire les pathogènes	la température de pasteurisation est environ de 75 à 90 °C
OP3	Préchauffage du lait	le lait chaud va servir pour son propre pré chauffage	Récupération de la chaleur, consommation d'énergie
OP4	Refroidissement du lait	Refroidit le lait	Produit final

4.3.3 Les principaux paramètres du processus de pasteurisation

Comme nous l'avons décrit précédemment, Les performances du système de diagnostic sont étroitement liées à la pertinence des paramètres du processus retenu. Selon l'expert, l'ensemble de paramètres présentés ci-dessous décrivent parfaitement l'état de fonctionnement du processus de pasteurisation et portent plusieurs informations concernant la qualité du lait, le niveau de production et la consommation d'énergie [Mouss, 2006] [Kadri et al., 2012a].

Tableau 5.4 : Les principaux paramètres du processus de pasteurisation

Sous système	paramètre	valeur de consigne	Désignation
Section de préchauffage du produit	T_e	$6 < T_e < 8 \text{ } ^\circ\text{C}$	Température d'entrée du produit
	D_e	$9000 < D_e < 12000 \text{ Kg/h}$	Débit d'entrée du produit
Section de chauffage de l'eau	D_a	11000 Kg/h	Débit d'eau
	T_a	$92 < T_a < 97 \text{ } ^\circ\text{C}$	Température d'eau chaude
	D_v	104 Kg/h	Débit du vapeur
	T_v	$110 \text{ } ^\circ\text{C}$	Température du vapeur
	P_r	$3 < P_r < 4 \text{ bars}$	Pression d'air comprimé
Section de chauffage du produit	T_c	$75 < T_c < 90 \text{ } ^\circ\text{C}$	Température de chauffage du produit
Section de refroidissement	D_{ag}	19900 Kg/h	Débit d'eau glace
	T_{ag}	$6 < T_{ag} < 10 \text{ } ^\circ\text{C}$	Température d'eau glace
	T_r	$6 < T_r < 8 \text{ } ^\circ\text{C}$	Température de refroidissement du produit
	D_s	$9000 < D_s < 12000 \text{ Kg/h}$	Débit de sorti du produit

4.3.4 Modes de fonctionnements étudiés

Tout système de diagnostic doit nous renseigner sur l'état de fonctionnement du système tout en assurant à la fois la fonction de détection (fonctionnement normal ou pas) et la détermination des causes. Pour ce deuxième procédé, nous avons aussi construit un AMDEC. A partir de l'AMDEC (voir annexe B), nous pouvons tirer 7 modes de défaillance qui sont présentés sur le tableau 5.5. En pratique, il existe d'autres classes de défaillance mais on se limite à ce nombre puisque le même type d'erreur peut avoir lieu pour les différentes sections du système [Mouss, 2006] [Kadri et al., 2012a].

Tableau 5.5 : Les modes de fonctionnements du processus de pasteurisation

Classe	1	2	3	4	6	7
Description	Fuite entre deux plaques	Bouchage	Perte d'efficacité de l'échange de chaleur	Variation du débit du produit en amont	Fermeture intempestive	arrêt de la pompe

5 L'algorithme AntTreeStoch

5.1 Utilisation de Netlogo

Nous avons choisi d'utiliser le logiciel de simulation Netlogo puisqu'il nous donne une gamme de liberté pour créer des modèles propres à la classification, l'optimisation,... dans les divers domaines : en sociologie, biologie, médecine, physique, chimie, mathématiques, informatique, économie et psychologie-sociale [Wilensky, 1999]. Il a d'abord été créé en 1999 par *Uri Wilensky* au *Center for Connected Learning and Computer-modelisation*, puis à l'Université Tufts à Boston. NetLogo né de *StarLogoT*, qui a été rédigé par *Wilensky* en 1997. En 2000, la *CCL* est déménagé à l'Université *Northwestern*, dans la région de Chicago. On peut le définir comme un environnement de programmation pour la modélisation et la simulation des systèmes naturels complexes et aussi les phénomènes collectifs. Il se base sur des agents mobiles capables d'effectuer de nombreuses opérations simultanément avec une grande précision. Il est aussi bien adapté à la modélisation des systèmes multi agents « par exemple : les fourmis, les termites, les abeilles... ». En Netlogo, on peut identifier 4 types d'Agents : *Patch* qui possède une position fixe et il est capable de créer une *turtle* et de contenir plusieurs *turtles*. Le deuxième type est l'agent *Turtle* qui est un agent réactif capable de se déplacer dans son monde. Le troisième type est *Link* qui est capable de connecter deux *turtles*. Le quatrième type d'agent est *l'Observer* qui est unique et il ne fait pas partie du monde de *turtle*. Il est capable de créer et de manipuler les *turtles*.

5.2 Les données de test

Les résultats obtenus par AntTreeStoch sont évalués en utilisant quatre bases de données. Les bases de données (BD1 et BD3) concernent le système de clinkérisation. La base BD1 est composée de 500 observations réparties sur deux classes qui représentent deux modes de fonctionnements (normal et anormal) (la figure 5.7). Chaque observation est composée de 30 paramètres. La base BD3 est composée de 200 observations réparties sur quatre classes qui représentent quatre modes de dysfonctionnements (marche perturbée, déplacement de zone, production des incuits, perte de calories (l'énergie)) [Kadri et al., 2011a]. La base de données BD2 concerne les données qui caractérisent le système de pasteurisation. Cet ensemble est composé de 300 observations qui représentent deux classes de fonctionnement (normal et anormal) et chaque observation contient 7 paramètres. La quatrième base est l'ensemble IRIS. IRIS est une base de données introduite par Fisher en 1936. Elle est constituée de 150 enregistrements qui représentent 3 classes. Chaque enregistrement est composé de 4

paramètres. L'intérêt d'utiliser IRIS est qu'on connaît à l'avance la classe réelle de chaque information et donc on peut évaluer la qualité de classification de l'algorithme [Frank et Asuncion, 2010].

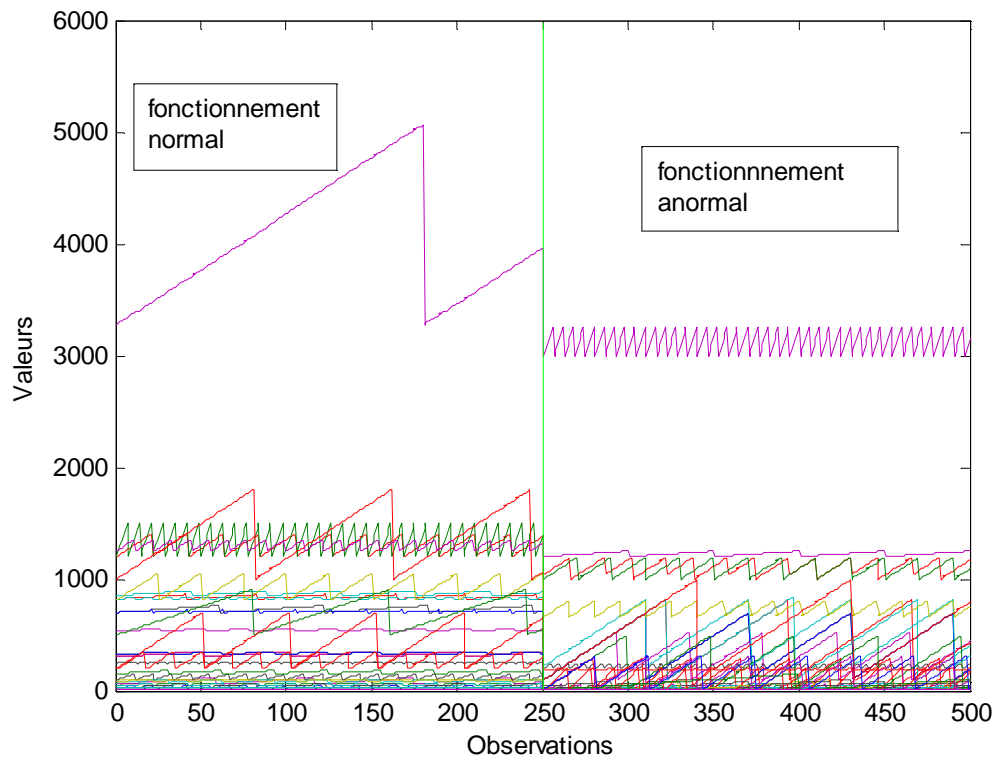


Fig. 5.4 : La base de données BD1 du système de clinkérisation

5.3 Les paramètres de AntTreeStoch

Comme tout autre algorithme, avant de passer à la phase de classification. Il faut fixer quelques paramètres. Ce problème représente l'une des inconvénients des méthodes biomimétiques. Puisque les valeurs des paramètres sont liées au nombre d'individus et la distribution des données sur la plage de représentation.

Notre modèle propose une interface simple pour configurer les différents paramètres de l'algorithme. Il permet aussi de réaliser plusieurs simulations en changeant à chaque fois la taille de vecteur d'état, le nombre de connexions possibles par nœud et la variable population qui indique la taille de la base de données (la figure 5.8).

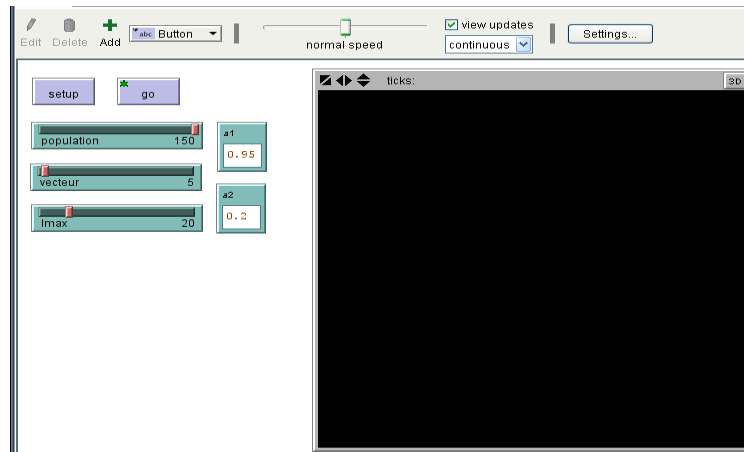


Fig. 5.5 : Le modèle multi-agents de l'algorithme AntTreeStoch

Le tableau 5.6 représente les valeurs des paramètres d'AntTreeStoch pour les quatre bases de données.

Tableau 5.6 : Les Paramètres d'AntTreeStoch

Paramètre	Description	Valeur initiale				Évolution
		BD1	BD2	BD3	IRIS	
$\alpha 1$	Evolution des valeurs des seuils (SSim)	0.95	0.97	0.95	0.92	Stable
$\alpha 2$	Evolution des valeurs des seuils (SDissim)	0.2	0.07	0.2	0.1	Stable
SSim(fi)	Seuil de similarité	Sim moy - max (fi)				SSim(fi) * $\alpha 1$
SDissim(fi)	Seuil de dissimilarité	Sim moy - min (fi)				SDissim(fi) + $\alpha 2$

Ces paramètres sont fixés après l'exécution de plusieurs simulations en utilisant comme entrée un ensemble restreint de données. Ces paramètres permettent une bonne distribution des individus sur l'arbre et accélèrent l'exécution de l'algorithme.

5.4 Les mesures d'évaluation

Les résultats de classification obtenus par une implémentation séquentielle et une autre parallèle à base d'un système multi-agents sont comparés en termes de nombres de cycles de temps et de nombre de classes obtenues. Nous avons pris une autre mesure d'évaluation qui est la pureté d'une classe. Elle donne le pourcentage des données bien classées.

$$p = \frac{1}{N} \times \sum_{i=1}^{C_t} \max M_i \quad (5.1)$$

Où p représente la pureté d'une classe, M représente la matrice de confusion et N est le nombre total de données.

Et finalement, nous allons voir l'influence du nouveau paramètre S_{id} sur la taille de l'arbre.

5.5 Les résultats

Le tableau 5.7 représente les résultats obtenus d'une implémentation séquentielle et une autre parallèle de l'algorithme AntTreeStoch. Sachant que les données des trois bases ne sont pas triées pour montrer l'indépendance entre l'ordre de données et la qualité de classification.

Tableau 5.7 : Résultats obtenus par une implémentation séquentielle et une autre parallèle de l'algorithme AntTreeStoch

<i>Base de données</i>	<i>Implémentation</i>	<i>Nombre de classes</i>	<i>Taille de l'arbre</i>	<i>Taux d'erreur</i>	<i>Nombre de mouvements</i>
IRIS	Séquentielle	3	5	0.27 ± 0.1	≈ 450
	Parallèle SMA	3	5	0.31 ± 0.4	≈ 400
BD1	Séquentielle	2	10	0.32 ± 0.4	≈ 2800
	Parallèle SMA	2	9	0.35 ± 0.4	≈ 2400
BD2	Séquentielle	2	8	0.31 ± 0.1	≈ 1500
	Parallèle SMA	2	8	0.33 ± 0.1	≈ 1300
BD3	Séquentielle	4	8	0.26 ± 0.1	≈ 1400
	Parallèle SMA	4	7	0.33 ± 0.1	≈ 1200

D'après le tableau 5.7, nous remarquons que l'implémentation à base d'un système multi-agents influe négativement sur la qualité de classification avec un taux d'erreur plus élevé. Mais comme prévu, l'implémentation parallèle a amélioré la vitesse d'exécution de l'algorithme.

Un des inconvénients de l'algorithme AntTreeStoch est qu'il nécessite le calcul de distances de la fourmi f_i (donnée à classer) à toutes les fourmis f_+ qui représentent l'ensemble de nœuds visités par la fourmi f_i avant d'être classée. Ce traitement demande un temps de calcul considérable si la base de données à classer est relativement grande.

Il est donc préférable de trouver un moyen pour accélérer cette étape. Dans ce but nous avons proposé (voir chapitre 2) un nouveau paramètre appelé S_{id} pour minimiser le nombre des nœuds de l'arbre. Le tableau 5.8 montre le nombre des nœuds et la taille de l'arbre avant et après l'ajout du paramètre S_{id} [Kadri et al., 2010a].

Tableau 5.8 : L'arbre final obtenu par AntTreeStoch avec et sans le paramètre S_{id}

Base de données	Algorithme	Nombre de classes	Taille de l'arbre	Taux d'erreur	Nombre de nœuds
IRIS	Basale	3	5	0.27 ± 0.1	150
	Version améliorée	3	4	0.27 ± 0.1	80
BD1	Basale	2	10	0.32 ± 0.4	500
	Version améliorée	2	8	0.32 ± 0.4	453
BD2	Basale	2	8	0.31 ± 0.1	300
	Version améliorée	2	6	0.31 ± 0.1	237
BD3	Basale	4	8	0.26 ± 0.1	200
	Version améliorée	4	6	0.26 ± 0.1	120

Le tableau 5.8 montre que le paramètre S_{id} réduit la taille de l'arbre et le nombre de nœuds. Par conséquent, le temps de reconnaissance d'une nouvelle information devient relativement petit. Faire réduire la taille de l'arbre va faciliter son interprétation par l'expert et offre la possibilité de l'exploiter par une application auxiliaire pour prédire un dysfonctionnement.

Comme nous l'avons évoqué dans le premier chapitre, un des inconvénients des méthodes de diagnostic est le fait qu'une nouvelle observation située entre deux classes peut être rejetée en ambiguïté ou en distance alors que celle-ci appartient à l'un des modes de fonctionnement définis par ces classes. Le problème réside dans la taille de l'ensemble d'apprentissage puisque cette base de données n'englobe pas tous les états possibles du système. Donc, il est indispensable de prendre en considération l'évolution des modes de fonctionnement pour avoir un diagnostic sans erreur. Notre algorithme offre cette possibilité en exploitant les données récemment identifiées pour classer une nouvelle observation.

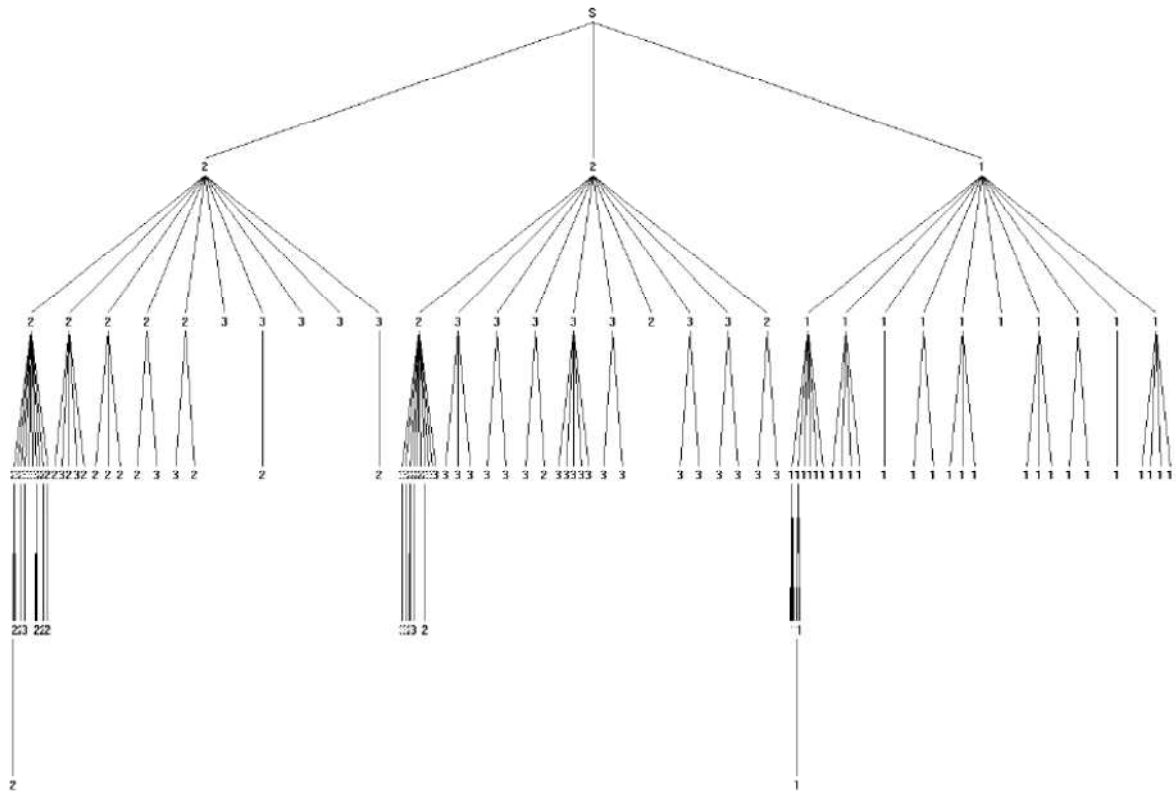


Fig. 5.6 : Exemple d'arbre obtenu par AntTreeStoch sur la base IRIS

La figure 5.9 montre un exemple d'exécution de l'algorithme AntTreeStoch sur la base de données IRIS. La propriété de parallélisme qui se trouve dans la plateforme Netlogo permet d'avoir des résultats différents du même scénario à chaque exécution.

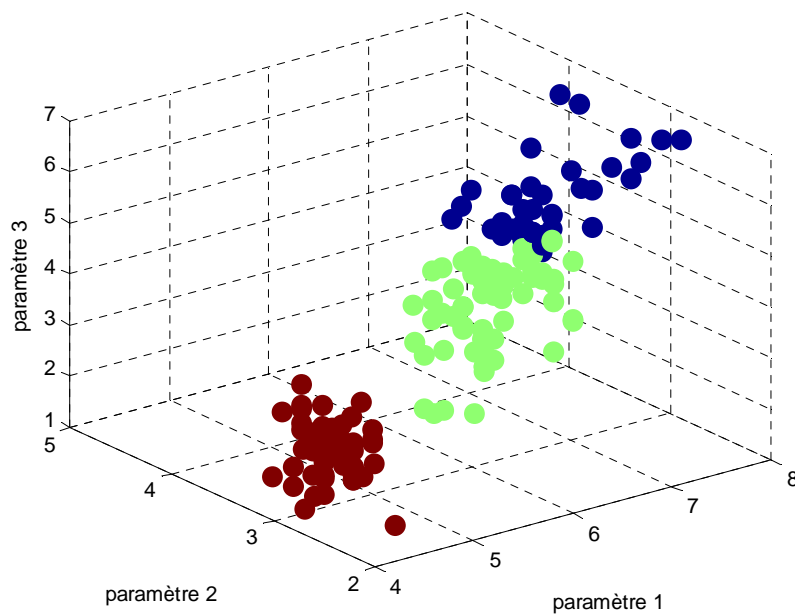


Fig. 5.7 : Vue en 3D de la base IRIS

Dans ce cas, on remarque que l’algorithme arrive à trouver le nombre exact de classes mais avec un chevauchement entre les classes 2 et 3. Ce résultat est causé par la petite distance qui se trouve entre les deux classes (figure 5.10).

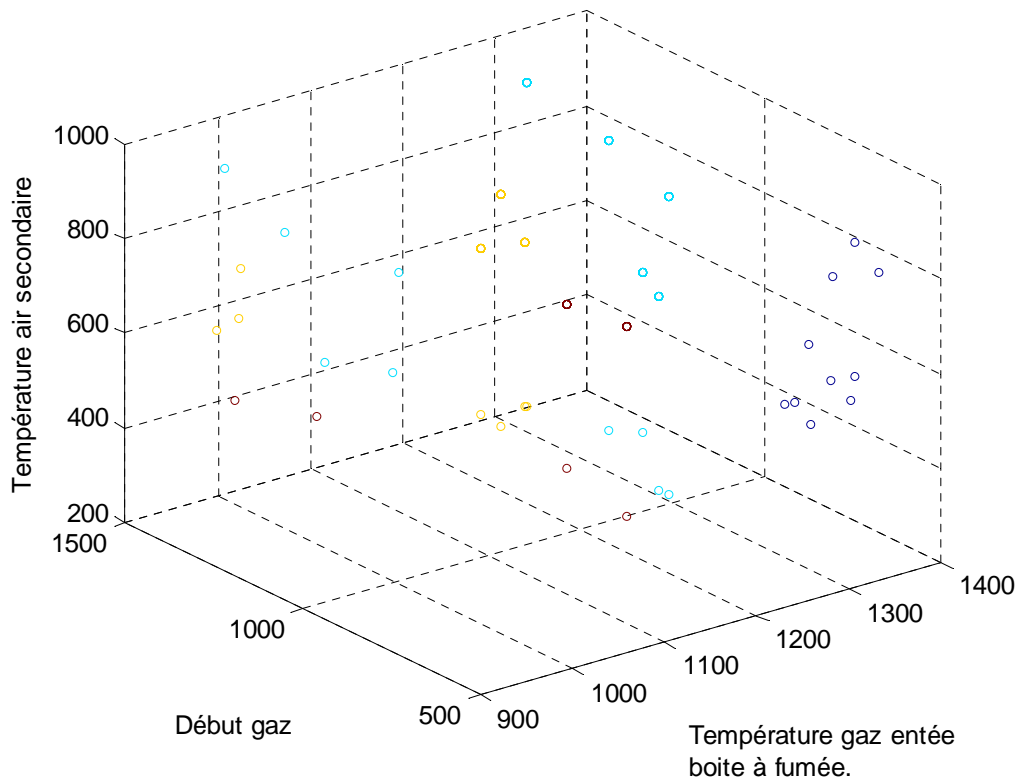


Fig. 5.8: Visualisation des quatre modes de dysfonctionnement du premier procédé en utilisant 3 paramètres

La figure 5.11 représente le résultat d’un diagnostic du premier procédé en utilisant uniquement trois paramètres. La figure montre quatre modes de dysfonctionnement mais trois de ces quatre classes ne sont pas suffisamment éloignées. Pour obtenir un bon diagnostic, il est donc nécessaire d’utiliser une méthode de sélection de paramètres pour prendre en compte tous les paramètres discriminants. Cette dernière est réalisée dans la suite de ce chapitre.

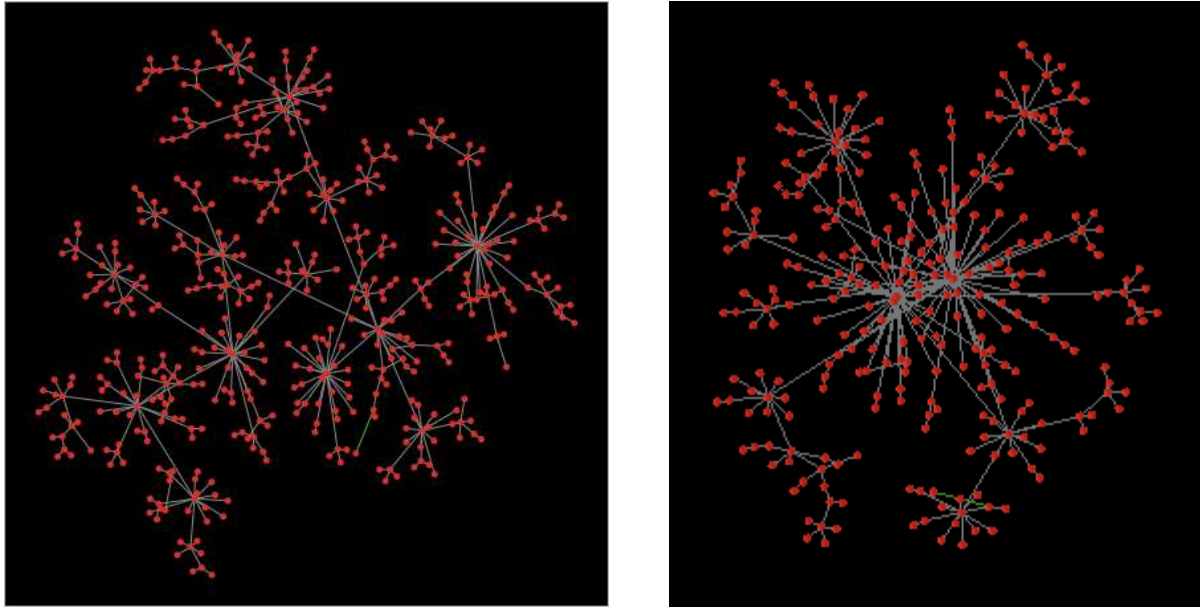


Fig. 5.9 : Visualisation 3D de l'arbre de classification de la base BD1 (500 nœuds) et de la base BD2 (300 nœuds)

Un autre avantage du modèle développé est qu'il offre une image globale sur le comportement du système (figure 5.12). Il est simple d'observer l'arbre de fourmis en ajoutant des étiquettes aux nœuds qui représentent les racines de chaque classe. De cette façon, la probabilité de passer d'un mode de fonctionnement à un autre est relative à la distance qui existe entre les classes correspondantes. Cette opération est très simple à réaliser et ne demande pas une grande base de données [Kadri et al., 2010a].

6 Application de l'algorithme Lumer & Faieta

6.1 Configuration de l'algorithme

Avec ce deuxième algorithme, nous avons utilisé les mêmes bases de données. Mais cette fois, nous allons présenter deux scénarii de simulation pour voir l'influence de paramètres de l'algorithme sur la qualité de classification. Nous n'allons pas calculer le taux de classification puisque ce type d'algorithme est utilisé uniquement pour estimer le nombre de classes. Notre modèle de simulation offre une interface graphique qui facilite l'identification des classes et offre la possibilité d'arrêter l'exécution lorsqu'on juge que la classification est satisfaisante (figure 5.13).

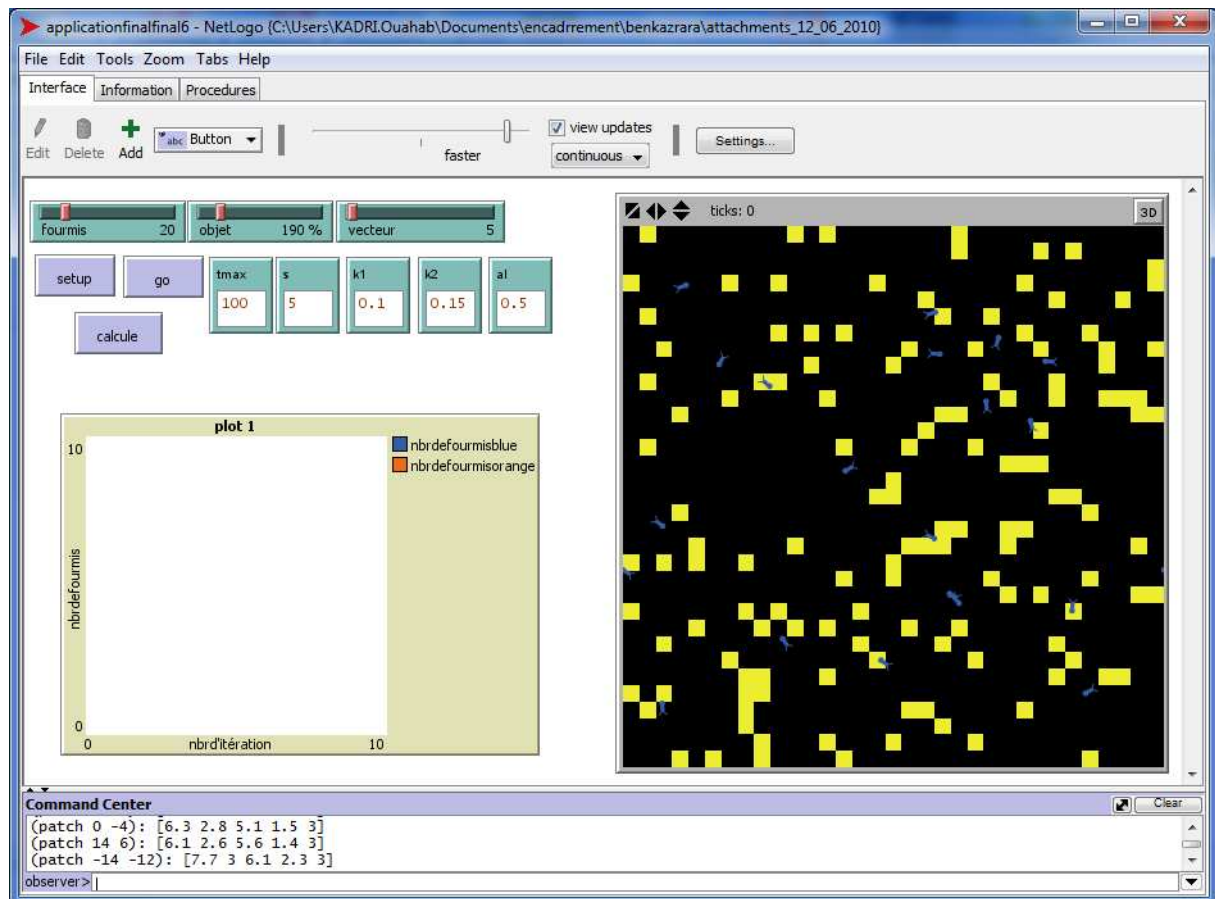


Fig. 5.10 : Interface de configuration de LF

Pour tous les scénarii de simulation, nous avons utilisé les paramètres d'entrées suivants :

- Tmax : représente le critère d'arrêt, on l'utilise pour spécifier le nombre maximal d'itération ;
- S : fixe le nombre de comparaisons en indiquant le nombre de fourmis voisines ;
- K1 : permet de contrôler la forme de la densité de p_p ;
- K2 : permet de contrôler la forme de la densité de p_d .

Pour réaliser ces deux scénarii, nous avons modifié la valeur du paramètre de région et les contraintes de ramassage et de dépôt. Dans le premier scénario, nous avons configuré notre modèle comme suit: Tmax=3000, S = 3, K1 = 0.1, K2= 0.15, A1= 0.5 et le degré de classification: 50%. Concernant le deuxième scénario, les nouvelles valeurs des paramètres sont comme suit : S = 5 et le degré de classification : 1% et nous avons gardé les mêmes valeurs pour les autres paramètres.

6.2 Résultats des scénarii :

La première configuration a donné une classification acceptable mais non optimale. Ce résultat est étroitement lié aux valeurs initiales des paramètres de l'algorithme. Nous pouvons expliquer l'influence de paramètres sur le résultat final par les points suivants :

Quand une fourmi se déplace sur la grille, elle calcule la similarité entre l'objet trouvé et son voisinage. Si cet objet est un élément isolé alors la distance euclidienne égale à zéro. Autrement dit la fonction $f(o_i)$ de similarité moyenne devient $f(o_i) = 1/S2 \approx 0$. Dans ce cas, la probabilité que l'objet soit ramassé est maximale. Dans le cas contraire, c'est-à-dire, il existe plus d'un objet dans le voisinage de l'objet trouvé. Ici, la fourmi calcule sa distance euclidienne et la fonction de similarité moyenne qui détermine l'action de la fourmi (l'objet soit ramassé ou non).

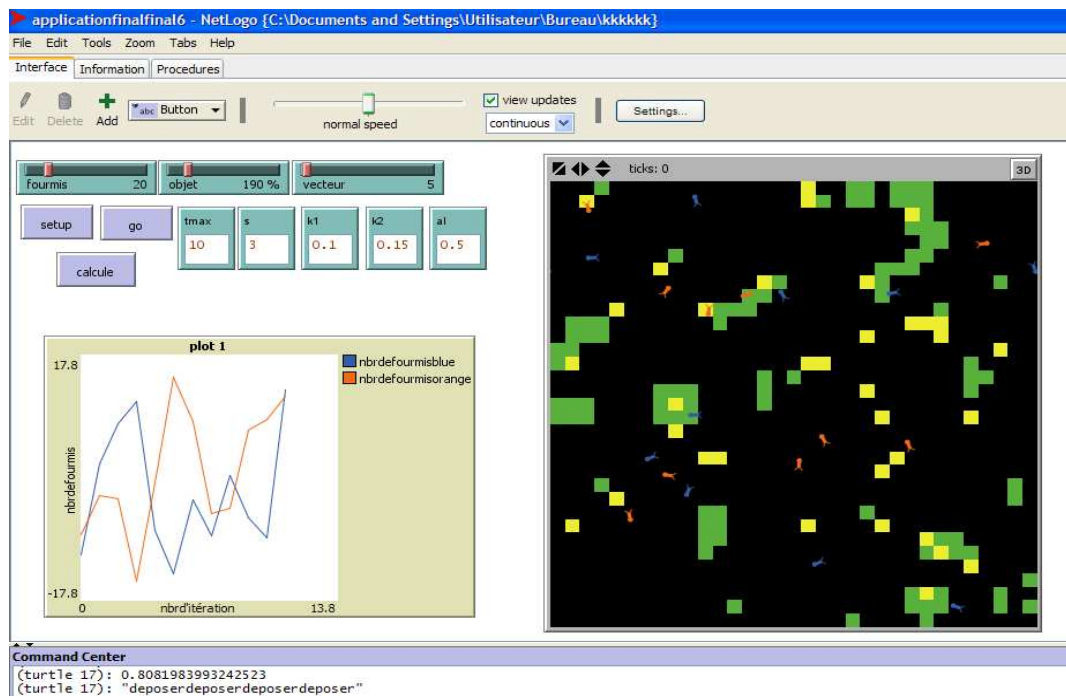


Fig. 5.11 : Modèle de simulation de l'algorithme Lumer & Faieta

A travers l'exécution du scénario1, il apparaît que les agents de l'ensemble exécutent le code simultanément, où chaque agent se déplace sur la grille. Si l'agent est dans un emplacement qui contient un objet où il y aurait d'autres objets qui lui sont similaires, il les classe puis il continue à se déplacer, sinon l'agent ramasse l'objet et continue à se déplacer jusqu'au moment où il trouverait des objets qui lui sont similaires. Dans ce cas, il les classerait dans son entourage. Nous constatons que les agents classent et regroupent les objets isolés si la région est restreinte.

Dans ce deuxième scénario, nous avons supposé que la valeur de paramètre S est supérieure ou égale à 5. Cette deuxième configuration a conduit à une mauvaise classification de données (figure 5.15).

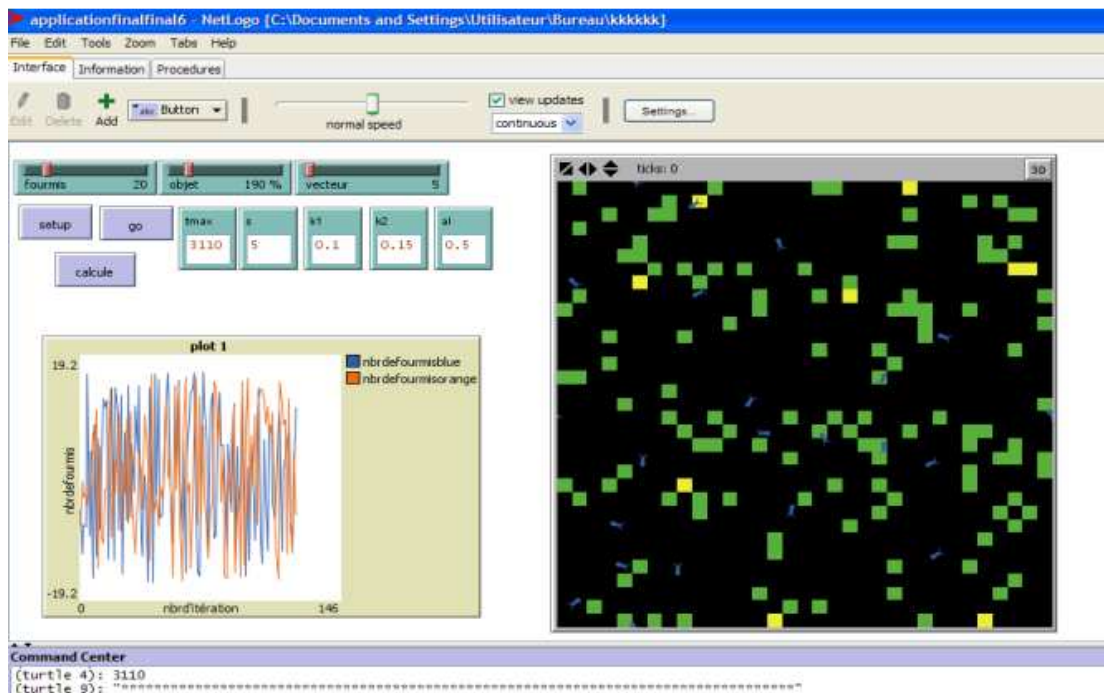


Fig. 5.12 : Résultat de la simulation de l'algorithme Lumer & Faieta avec ($S \geq 5$)

Les résultats obtenus dans toutes les configurations en utilisant les trois bases de données ont montré que l'algorithme génère un nombre de classes très éloigné du nombre réel de classes. Mais, la présentation graphique du résultat aide l'expert à détecter l'apparition d'une nouvelle classe de dysfonctionnement. Donc, on peut déduire que l'utilisation d'une région restreinte conduit toujours à une classification rapide et relativement optimale [Kadri et al., 2011b].

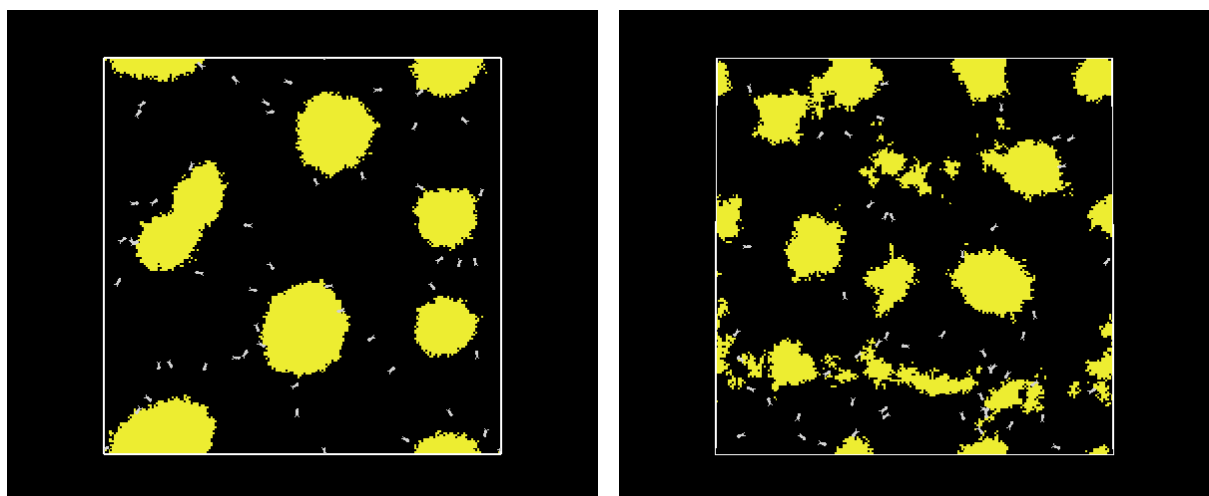


Fig. 5.13 : Simulation de l'algorithme LF avec et sans des fourmis rapides (BD1)

Comme le montre la figure 5.16, la simulation de l'algorithme LF sans l'utilisation des fourmis rapides donne un très grand nombre de classes par rapport au nombre réel de modes de fonctionnement du système.

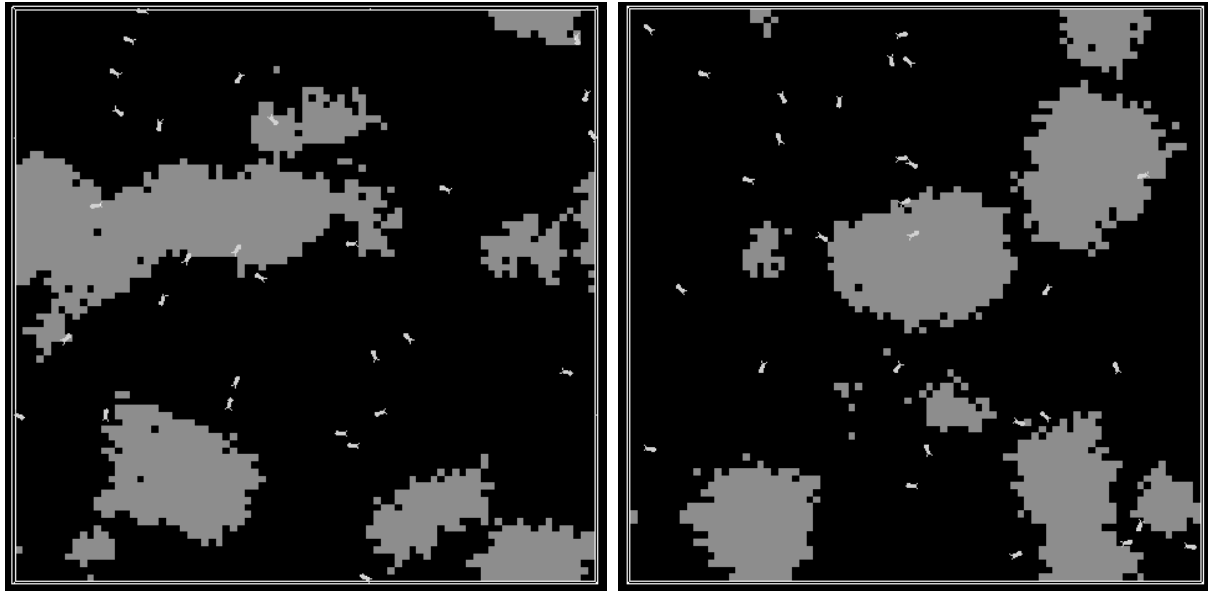


Fig. 5.14 : Simulation de l'algorithme LF avec et sans des fourmis rapides (BD2)

Généralement, l'ajout de fourmis rapides a permet d'avoir de bons résultats mais pas toujours puisque la vitesse de chaque fourmi est définie d'une façon aléatoire et il est possible que la colonie ne contienne que quelques fourmis lentes. Dans ce cas, les fourmis peuvent construire plusieurs classes qui représentent le même mode de fonctionnement du système (figure 5.17).

7 Application de l'algorithme Hybrid wrapper/filter-based ACO-SVM

7.1 Les données de test

Pour ce dernier algorithme nous avons utilisé cinq bases de données (RCK1 (*Rotary Cement kiln*), RCK2, BD2 et Vehicle). Nous avons utilisé la base de données BD2 pour réaliser la phase de configuration du classifieur. Les données de Vehicle ont été recueillies en 1987 par Siebert [Frank et Asuncion, 2010]. Vehicle est constituée de 800 enregistrements qui représentent 4 classes et chaque enregistrement est composé de 18 attributs. La base de données RCK1 est composée de 200 observations qui représentent 4 modes de fonctionnement et la base de données RCK2 est composée de 500 observations qui représentent 2 modes de fonctionnement. Les détails des bases de données sont présentés dans le tableau 5.9 :

Tableau 5.9 : les détails de bases de données utilisées dans les tests

Base de données	Classe	Description	Base d'apprentissage	Base de test
RCK1 30 attributs	1	Marche perturbée	31	30
	2	Déplacement de zone	20	20
	3	Production des incuits	21	21
	4	Perte de calories (l'énergie)	29	28
Vehicule 18 attributs	1	Double decker bus	100	100
	2	Chevrolet van	100	100
	3	Saab 9000	100	100
	4	Opel Manta 400	100	100
RCK2 47 attributs	1	Fonctionnement normal	125	125
	2	Fonctionnement anormal	125	125

Le tableau 5.9 montre que nous avons utilisé presque le même nombre de vecteurs dans les deux bases de données (apprentissage et test). Nous avons utilisé cette répartition pour montrer l'efficacité de l'algorithme proposé.

Contrairement à l'algorithme précédant, nous avons utilisé la même configuration dans tous les tests réalisés. L'initialisation de paramètres de l'algorithme Hybrid wrapper/filter-based ACO-SVM est présentée dans le tableau 5.10 :

Tableau 5.10 : Les paramètres de l'algorithme de sélection

Paramètre	Valeur	Description
N_A	20	Nombre d'Agents
F_a	0.2	Taux de comportement aléatoire
ρ	0.3	Taux d'évaporation
S_Min_FP	Min FH	Seuil minimal de phéromone
S_Max_FP	Max FH	Seuil maximal de phéromone

Ces paramètres sont fixés après l'exécution de plusieurs simulations en utilisant comme entrée un ensemble restreint de données. Ces paramètres permettent une exécution rapide de l'algorithme.

7.2 Le facteur heuristique FH

Le facteur heuristique FH est pris en considération que par les fourmis qui ont un comportement lié à la probabilité PS. Les fourmis qui ont un comportement aléatoire sont

utilisées pour découvrir des nouveaux espaces de recherche. La figure 5.18 représente les valeurs du facteur heuristique FH en utilisant la base de données BD2.

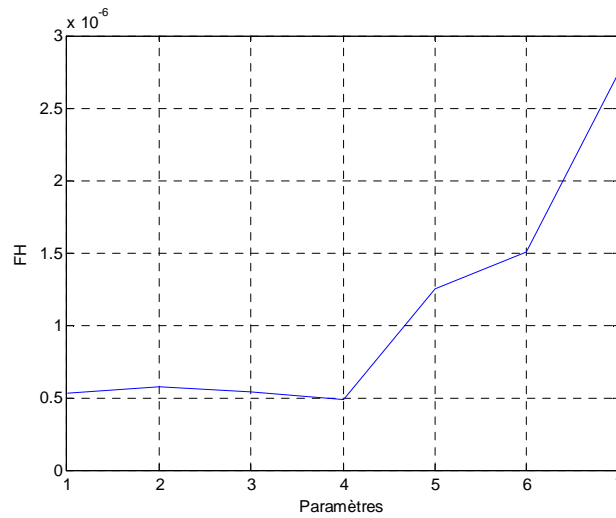


Fig. 5.15 : Le facteur heuristique FH de la base BD2

D'après la figure 5.18, on constate que la valeur de FH7 va favoriser les fourmis de choisir le chemin 1 qui signifie la présence du septième paramètre dans le sous-ensemble final.

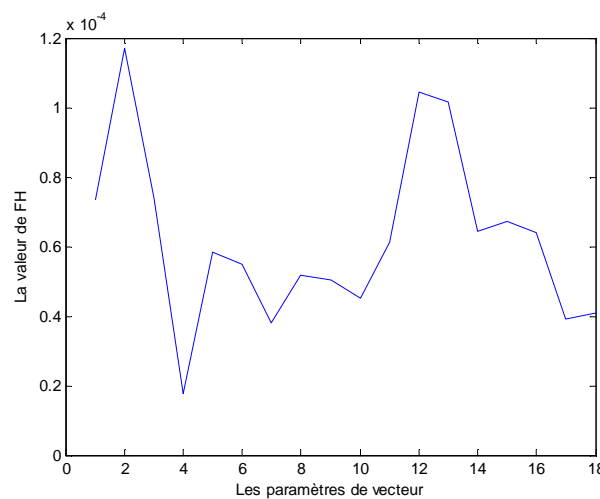


Fig. 5.16 : Le facteur heuristique FH de la base Vehicle

Selon la figure 5.19, nous remarquons que le deuxième paramètre de la base de données Vehicle représente le paramètre le plus pertinent puisqu'il possède la plus grande valeur de FH. En conséquence il va être présent dans la solution finale.

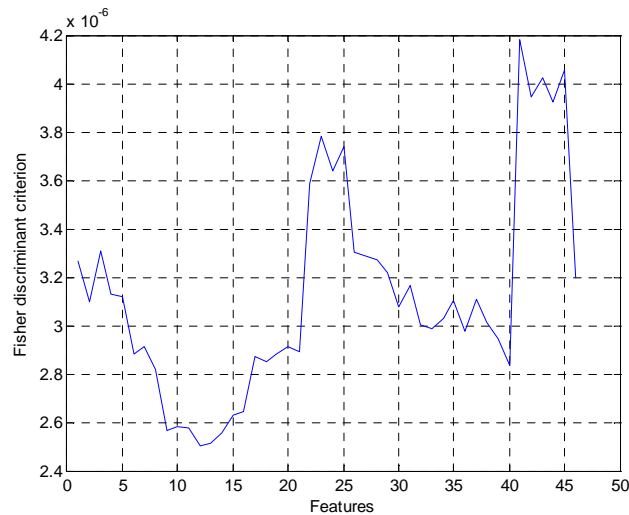


Fig. 5.17 : Le facteur heuristique FH de la base RCK1

La figure 5.20 représente le facteur heuristique FH de la base de données RCK1. Nous remarquons que le paramètre numéro 12 n'a pas d'influence sur la classification de données. Donc, il ne sera pas présent dans l'ensemble final de paramètres sélectionnés.

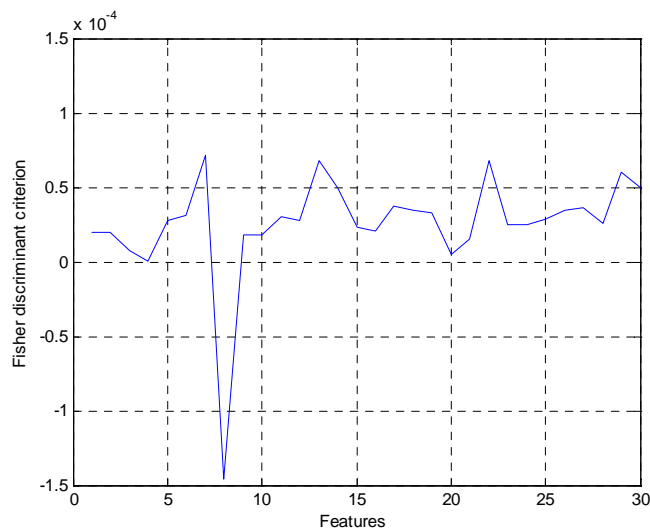


Fig. 5.18 : Le facteur heuristique FH de la base RCK2

Nous pouvons faire la même remarque sur la base de données RCK2 (figure 5.21).

7.3 La configuration du classifieur

Pour la configuration du classifieur, nous avons utilisé l'ensemble de données qui caractérise le système de pasteurisation. Cet ensemble est composé de 300 observations qui représentent deux classes de fonctionnement et chaque observation contient 7 attributs. Nous avons utilisé aussi quelques bases de données binaires téléchargées à partir de site officiel de UCI

(University of California, Irvine). Le tableau 5.12 résume les différentes caractéristiques de ces bases de données :

Tableau 5.11 : La description de bases de données

Nom de la base de données	Nombre de paramètres	Nombre de classes	Nombre de données
Ex8a	2	2	863
Four_classe	2	2	862
Heart_scale	13	2	270

Nous avons utilisé comme critère d'évaluation de la qualité de classification, le nombre d'observations bien classées sur le nombre total d'observations.

7.3.1 Réglage des hyper-paramètres

Nous avons utilisé dans la phase d'apprentissage l'hyper-paramètre C qui sert à fixer le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la marge. Il s'agit d'un facteur de coût positif qui pénalise les exemples des informations malles classées. En pratique, le comportement du SVM est sensible à la valeur de C uniquement si les données d'apprentissage ne sont pas séparables. Dans ce cas, il existe des valeurs critiques qui peuvent compromettre la performance du classifieur. Une très grande valeur de C peut conduire à une minimisation de la fonction objective par le SVM et elle ne soit plus convexe et empêcherait sa convergence. Une très faible valeur de C tend à diminuer la capacité du classifieur. Le deuxième hyper-paramètre est le coefficient gamma σ qui caractérise la fonction noyau. Le tableau 5.13 résume quelques résultats obtenus en changeant les valeurs de couple (C, σ) pour le noyau gaussien.

Tableau 5.12 : Résultats obtenus pour les différentes valeurs de couple (C, σ)

C	1	10	100,	1	1,	0.5	0.5,
σ	0.07	0.07	0.07	=1	10	10	100
T	95.666	99.333	99.333	99.333	99.333	67	66.333
N	287	298	298	298	298	201	199
t	0.0618	0.0103	0.10552	0.0119	0.0123	0.0121	0.0117
n	279	244	243	287	293	283	274

Sachant que :

- T : taux de classification ;
- N : nombre d'observations bien classées ;
- t : temps d'exécution ;
- n : nombre de vecteurs de support.

En changeant le type de la fonction noyau, nous obtenons les résultats suivants :

- Pour le noyau linéaire, le taux de classification obtenu est 70.7143 % ;
- Pour le noyau gaussien(RBF), le taux de classification obtenu est 99.3333% ;
- Pour le noyau précalculé le taux de classification obtenu est 67.1429 %.

7.3.2 Les résultats de classification

Nous avons testé les performances du classifieur en utilisant le critère de qualité présenté ci-dessus. Nous avons utilisé comme une configuration de base la fonction noyau RBF et pour les hyper-paramètres (C, σ) les valeurs sont (10, 0.07) [Benathmene et Bounouara 2011] .

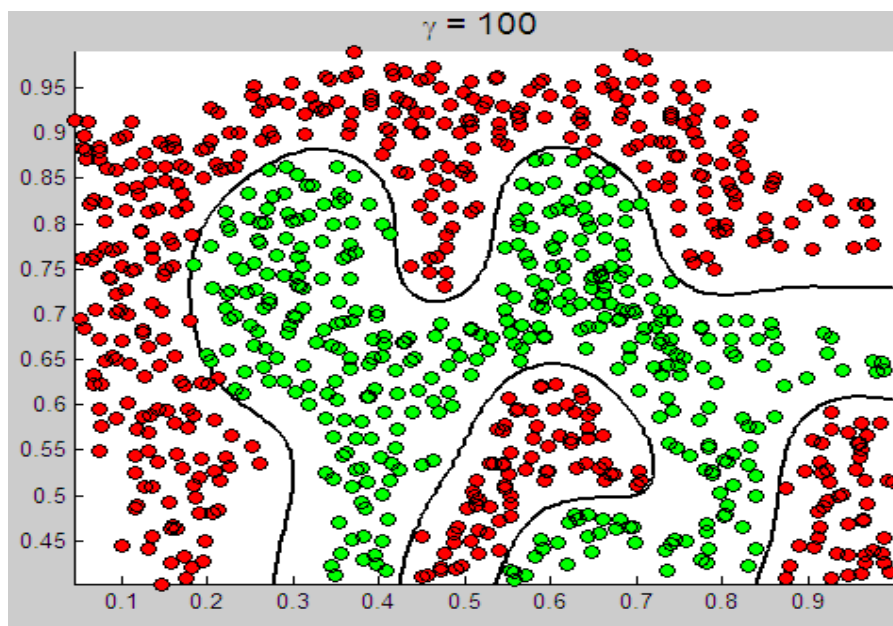


Fig. 5.19 : Résultat de classification de la base de données Ex8a

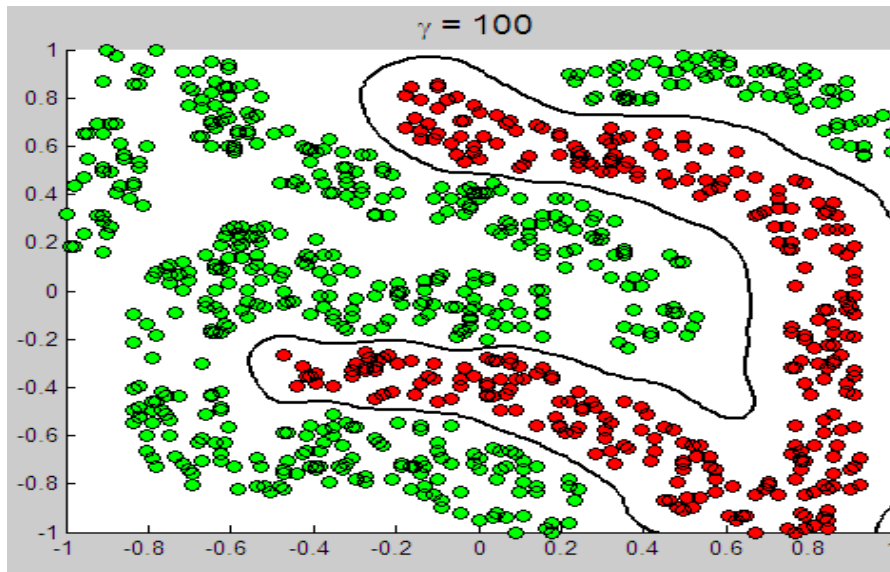


Fig. 5.20 : Résultat de classification de la base de données Four_classe

Dans toutes les figures, nous n'avons utilisé que deux paramètres. Mais dans la construction du modèle SVM, nous avons utilisé tous les paramètres des bases de données.

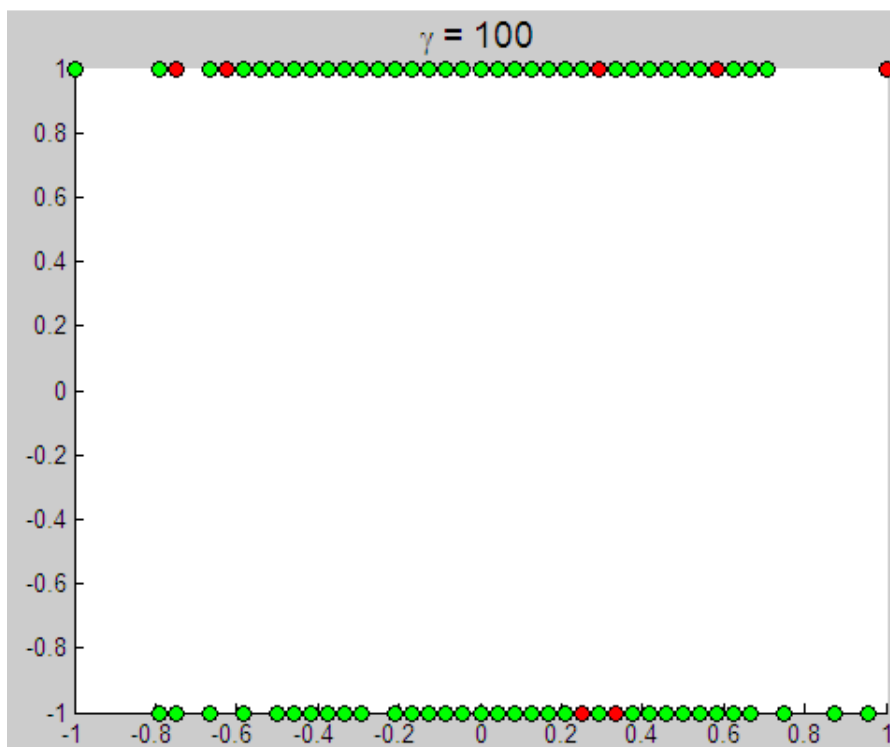


Fig. 5.21 : Résultat de classification de la base de données Heart_scale

La figure 5.24 montre que les observations de la base de données du système de pasteurisation de lait ne sont pas linéairement séparables.

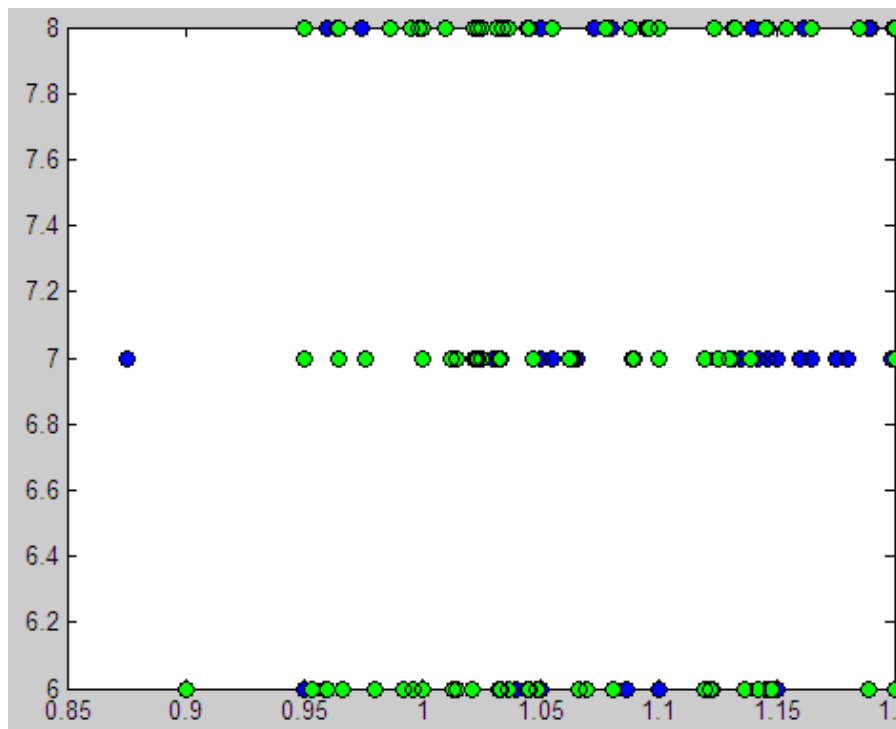


Fig. 5.22 : Résultat de classification de la base de données du système de pasteurisation de lait

Pour les trois premières bases de données, nous avons obtenu un taux de classification égale à 100%. Et pour la base de données réelle, on a obtenu un taux de classification égale à 99.33%.

7.4 Les résultats

Nous avons testé les performances de l’algorithme Hybrid wrapper/filter-based ACO-SVM en utilisant le critère d’évaluation présenté dans la section 4.4. Le tableau 5.14 montre la qualité de classification en utilisant :

- Le meilleur paramètre discriminant obtenu à partir des figures 5.18-5.21;
- Le meilleur sous-ensemble de paramètres engendré par l’algorithme ;
- L’ensemble de tous les paramètres.

Pour l’implémentation, nous avons utilisé l’environnement Matlab avec la boîte à outils LIBSVM. Les algorithmes utilisés dans les expériences qui suivent sont Hybrid wrapper/filter-based ACO-SVM, Filter-based ACO et Filter-based GA.

Nous avons développé l’algorithme Filter-based GA en utilisant la boîte à outils GATool de matlab. Les résultats de Hybrid wrapper/filter-based ACO-SVM ont été comparés avec ceux de Filter-based ACO et Filter-based GA.

Les valeurs de paramètres de l'algorithme génétique utilisés sont résumées dans le tableau 5.13.

Tableau 5.13 : Les entrées de GATOOL

Fitness function			@init11
Number of variables			18
Options	Population	Population type	Bit string
		Population size	18
	Fitness scaling	Scaling function	Rank
	Selection	Selection function	Tournament.
	Reproduction	Elite count	2
		Crossover fraction	0.8
	Mutation	Mutation function	Uniforme.
	Crossover	Crossover function	Two point.
	Migration	Direction	Both.
		Fraction	0.2
		Interval	20
	Hybrid function	Hybrid function	None.
	Stopping criteria	Generations	100
		Time limit	inf
		Fitness limit	-inf
		Stall generations	50
		Stall time limit	20
Display to command window	Level of display	Off	
Vectorize	Fitness function is vectorized	Off	

Pour les algorithmes Hybrid wrapper/filter-based ACO-SVM, Filter-based ACO, nous avons utilisé les paramètres présentés dans le tableau 5.7. Tous les tests sont réalisés sur une machine *Intel Pentium Dual Core T4400 2.2 GHz* avec 3 GB de RAM.

Tableau 5.14 : Les performances de classification en utilisant les différentes entrées

Algorithme	Paramètres	RCK1		RCK2		BD2	
		Taux d'erreur	F (V)	Taux d'erreur	F (V)	Taux d'erreur	F (V)
Hybrid wrapper/filter-based ACO-SVM	Sous-ensemble généré	11 %	0.4717	10 %	0.5078	13 %	0.7418
Filter based ACO	Sous-ensemble généré	13 %	0.6537	15 %	0.0210	17 %	0.0221
Filter-based GA	Sous-ensemble généré	11 %	0.7717	15 %	0.0210	17 %	0.0221
Tous les algorithmes	Un seul paramètre	75 %	0.0325	46 %	0.0061	72 %	0.0091
	Tous les paramètres	07 %	0.7875	10 %	0.0218	8 %	0.7806

Les résultats présentés dans le tableau 5.14 représentent les meilleures valeurs trouvées dans 20 exécutions. La meilleure valeur pour le couple (C, γ) en utilisant l'algorithme Hybrid wrapper/Filter based ACO-SVM pour les trois bases de données a été $(2^3, 2^{-5})$.

Le tableau 5.14 montre qu'on obtient un taux d'erreur acceptable avec le sous-ensemble généré par notre algorithme. On remarque aussi que la valeur de F(V) reflète bien la qualité de classification.

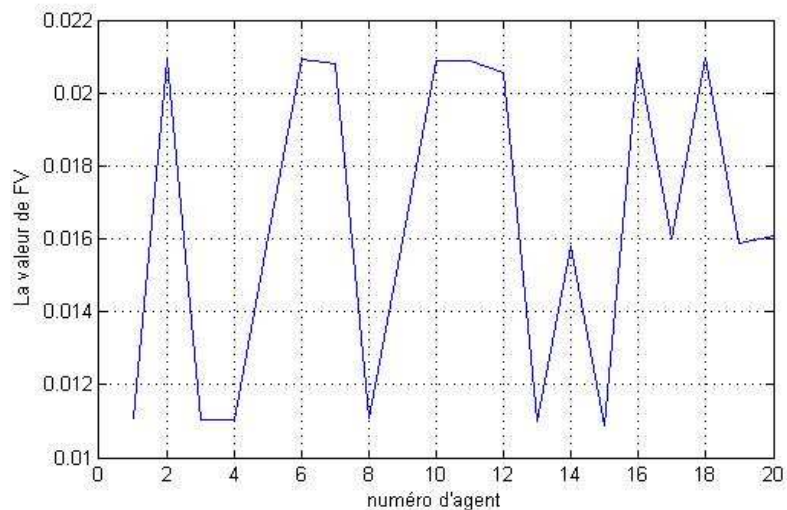


Fig. 5.23 : Les valeurs de FV dans la dernière itération (Vehicle)

La figure 5.26 montrent que huit agents trouvent la meilleure solution à partir de la sixième itération. Donc, ces résultats prouvent que la vitesse de convergence de notre algorithme est élevée.

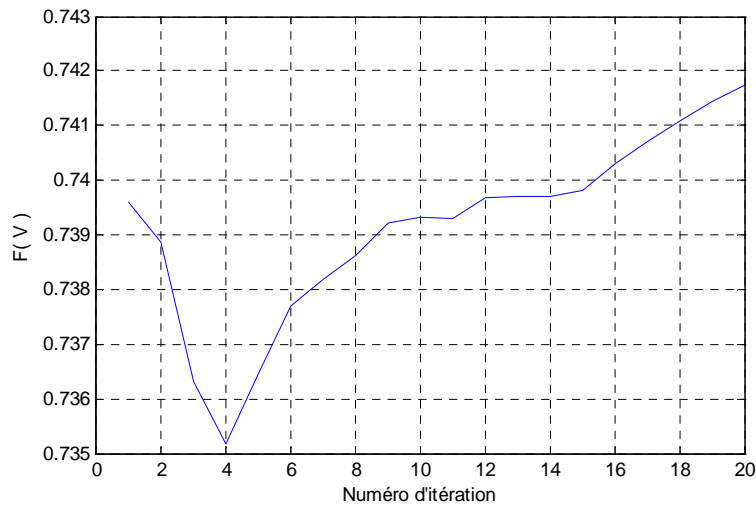


Fig. 5.24 : La meilleure solution obtenue dans chaque itération (BD2)

La figure 5.27 montre que les fourmis commencent à trouver une bonne solution à partir de la première itération. Nous remarquons aussi que l’algorithme converge vers une solution optimale à partir de la cinquième itération.

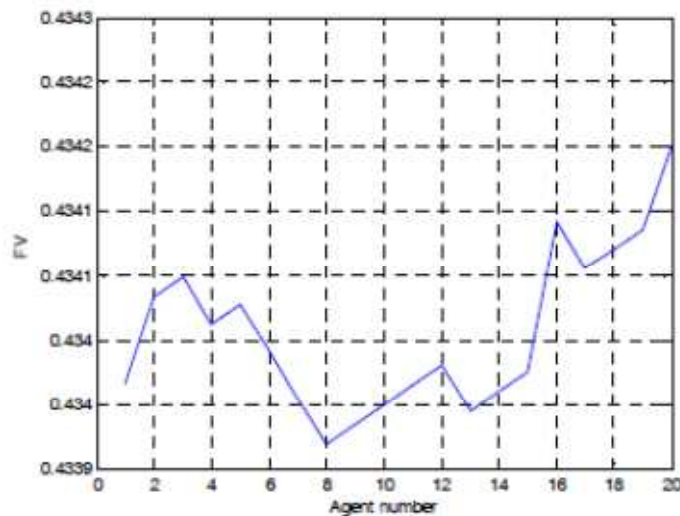


Fig. 5.25 : La valeur de FV obtenue par chaque agent dans la dernière itération (RCK1)

Les résultats trouvés montrent que plus de 80% d’agents trouvent une bonne solution dans la dernière itération. La densité de phéromone est la cause principale pour ce résultat.

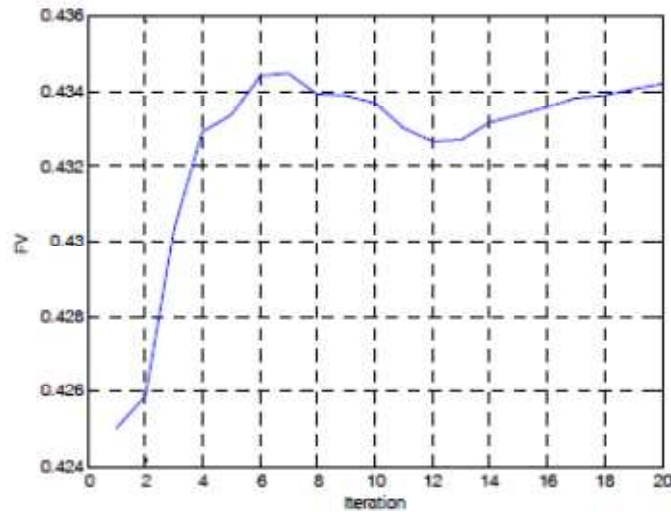


Fig. 5.26 : Le meilleur résultat obtenu à la fin de chaque itération (RCK1)

La figure 5.29 montre que nous trouvons la meilleure solution après uniquement 6 itérations. Cette solution représente le résultat optimal puisque nous trouvons cette même solution dans plusieurs itérations.

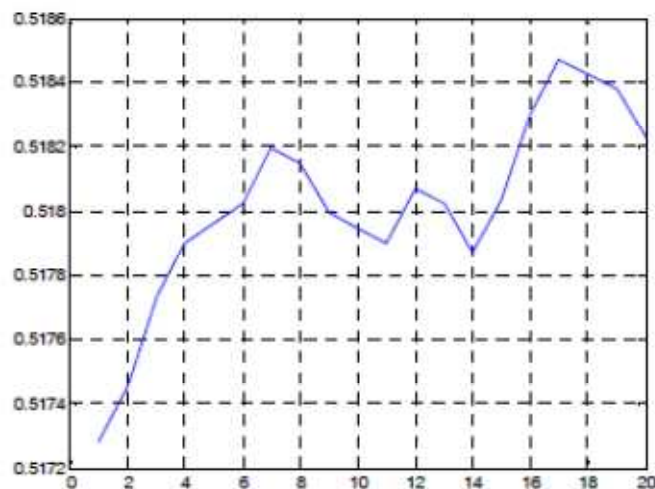


Fig. 5.27 : Le meilleur résultat obtenu à la fin de chaque itération (RCK2)

La figure 5.30 montre que nous trouvons une bonne solution après uniquement 5 itérations. Ce résultat montre la rapidité de notre algorithme. Nous pouvons encore améliorer la rapidité de notre algorithme en réduisant le nombre d'agents utilisés.

Selon le tableau 5.15, nous déduisons que notre algorithme arrive à illuminer un grand nombre de paramètres de RCK. Mais le sous-ensemble de paramètres sélectionnés n'est pas

nécessairement le même dans tous les exécutions. Nous avons trouvé le même résultat en appliquant l'algorithme Filter-based GA [Rezki et al., 2011] [Kadri et al., 2012b].

Tableau 5.15 : Description des paramètres sélectionnés

N° paramètre	Symbole paramètre	Description paramètre.	N° paramètre	Symbole paramètre	Description paramètre.
P1	W01P3	Pression sortie boîte à fumée.	P ₁₆	W01P1	Pression entrée boîte à fumée.
P2	COC	Teneur CO sortie cyclone A50	P ₁₇	W01T1	Température gaz entrée boîte à fumée.
P3	A50T1	Température gaz sortie cyclone A50.	P ₁₈	V31F1	Début gaz.
P4	A50P1	Pression sortie cyclone A50.	P ₁₉	V01F1	Début gaz.
P ₅	A52T1	Température gaz sortie cyclone A52.	P ₂₀	W01X1	Moment four
P ₆	A52P2	Pression cyclone A52.	P ₂₁	CCS	Consommation calorifique spécifique.
P ₇	A53T1	Température gaz cyclone A53	P ₂₂	W01S1	Vitesse four.
P ₈	A53T2	Température matière cyclone A53	P ₂₃	TV	Température virole four
P ₉	A53P1	Pression cyclone A53.	P ₂₄	W01T4	Température de cuisson.
P ₁₀	A54T1	Température gaz cyclone A54	P ₂₅	V07P1	Pression air primaire.
P ₁₁	K01P1	Pression air secondaire.	P ₂₆	PAL	Poids au litre.
P ₁₂	A54P2	Pression cyclone A54t.	P ₂₇	U01T1	Température clinker.
P ₁₃	O ₂ F	Teneur O ₂ boîte à fumée.	P ₂₈	FCAO	Chaux libre.
P ₁₄	COP	Teneur CO boîte à fumée.	P ₂₉	A54T2	Température matière cyclone A54
P ₁₅	O ₂ C	Teneur O ₂ sortie cyclone A50	P ₃₀	K01T1	Température air secondaire.

Les résultats présentés dans les figures 5.26-5.30 et le tableau 5.14 montrent que notre algorithme (Hybrid wrapper/filter-based ACO-SVM) est très précis. Autrement dit, il donne la solution optimale qui n'est pas toujours le cas avec un autre algorithme. En réalité, les résultats obtenus sur les bases de données du système de clinkérisation montrent que notre algorithme converge toujours vers la solution optimale dans toutes les exécutions.

8 Conclusion

Dans ce chapitre, nous avons présenté les performances des différents algorithmes proposés pour le diagnostic des systèmes complexes. La première partie du chapitre a été dédiée à l'étude de deux procédés industriels à savoir, le système de clinkérisation et le système de pasteurisation. Suite à une analyse détaillée des deux procédés, nous avons pu déterminer les différents modes de fonctionnement de chaque système et les différents paramètres qui les caractérisent.

Dans la deuxième partie, nous avons appliqué les algorithmes AntTreeStoch et Lumer & Faieta pour effectuer une classification. Celle-ci a été réalisée sur plusieurs bases de données qui représentent l'historique de fonctionnement des deux systèmes étudiés.

L'étape de sélection de paramètres a été réalisée en utilisant l'algorithme Hybrid wrapper/filter-based ACO-SVM. Avant l'application de cet algorithme, nous avons discuté la configuration des hyper-paramètres du classifieur SVM et le choix de la fonction noyau en utilisant plusieurs bases de données. Les résultats obtenus montrent que les paramètres sélectionnés par notre algorithme fournissent des taux de bonne classification comparables à ceux obtenus par les paramètres sélectionnés par les algorithmes Filter-based ACO et Filter-based GA.

CONCLUSION GENERALE

Conclusion générale

Les recherches menées au cours de cette thèse nous ont permis de développer plusieurs modules de diagnostic pour des systèmes complexes et dynamiques. Ces modules sont basés sur trois algorithmes de colonie de fourmis qui sont *AntTreeStoch*, *Lumer & Faieta* et *Binay ant colony*. Ces algorithmes ont été choisis pour leur simplicité et leur vaste domaine d'application. Toutefois ils ne peuvent pas être utilisés sous leur forme basale pour le développement des modules de diagnostic puisqu'ils présentent plusieurs limites :

- Pour l'algorithme *AntTreeStoch*,
 - L'exécution séquentielle de l'algorithme influe négativement sur la qualité de classification puisque le choix de la branche que va prendre une fourmi quelconque dépend des fourmis déjà classées. Donc la qualité de classification dépend fortement du tri initial à cause de l'absence totale de la propriété de parallélisme dans les déplacements des fourmis.
 - Un autre point faible d'*AntTreeStoch* est la non signification de déplacement aléatoire de fourmis dans le cas où la similarité entre deux fourmis est inférieur à un certain seuil. Ce déplacement aléatoire d'une fourmi va l'éloigner de la zone de recherche réelle.
 - Un autre inconvénient majeur de l'algorithme *AntTreeStoch* est la taille énorme de l'arbre engendré qui rend son interprétation difficile voire impossible si le nombre de données d'apprentissage est énorme. Aussi la croissance de la taille de l'arbre rend la phase de classification très longue, ce qui contredit l'objectif de l'utilisation des méta-heuristiques.
- Pour l'algorithme *Lumer & Faieta*,
 - Le déplacement des fourmis avec la même vitesse sur la grille résulte la création d'un grand nombre de classes.
 - Un autre inconvénient de cet algorithme est l'absence des traces de déplacements de fourmis. Le résultat de cet absence est que chaque fourmi peut exécuter des actions inutiles en transportant le même objet plusieurs fois.
- Pour l'algorithme *Binay ant colony*,

- La réutilisation de l'algorithme génétique dans la boucle de recherche a augmenté la complexité de l'algorithme sans améliorer l'efficacité de l'algorithme.
- Un autre inconvénient est la loi de probabilité utilisé par chaque fourmi qui demande un temps de calcul considérable.

En conséquence, nous avons proposé plusieurs adaptations afin que ces algorithmes soient aptes à être utilisés dans des modules de diagnostic.

- Nous avons proposé une version parallèle de l'algorithme AntTreeStoch basée sur un système multi-agents réactifs. Cette version permet de minimiser l'influence de tri initial sur le résultat final de classification.
- Nous avons introduit aussi pour minimiser la taille de l'arbre, un nouveau paramètre appelé S_{id} qui permet aux plusieurs fourmis de se connecter à la même position.
- Toujours pour améliorer cet algorithme, nous avons modifié les déplacements des fourmis en favorisant le chemin de la fourmi la plus similaire.
- Pour l'algorithme Lumer & Faieta, nous avons accéléré la vitesse de construction de classes en ajoutant un paramètre de vitesse différent pour chaque fourmi. Pour réduire le nombre de déplacements.
- Nous avons proposé une nouvelle variable qui permet de sauvegarder les identifiants des objets déplacés par la même fourmi.
- Pour améliorer la qualité de classification, nous avons ajouté aussi à l'algorithme des indices pour signaler les classes mal construites.
- Nous avons aussi proposé un nouvel algorithme appelé « Hybrid wrapper/filter-based ACO-SVM ». Cet algorithme permet la sélection de paramètres. Il combine les techniques de méthodes filtres et enveloppantes en profitant de la rapidité du rapport de Fisher et l'adaptation de paramètres sélectionnés au classifieur SVM. Il améliore la qualité de classification en fonction de la nature de données de la base d'apprentissage et le type de la fonction noyau utilisé. Il permet aussi de régler les hyper-paramètres de la fonction noyau.

Les algorithmes proposés dans ce manuscrit ont été testés sur des bases de données issues de deux systèmes industriels : le système de clinkérisation de la cimenterie de Aïn Touta et le système de pasteurisation de la laiterie Aurès de Batna, ainsi sur quelques bases de données d'UCI.

Il sera intéressant de tester ces algorithmes sur d'autres applications issues du domaine de la reconnaissance d'images, vidéos, écriture, etc.

Nous n'avons pas pu tester nos algorithmes en ligne pour évaluer leurs performances réelles.

En conclusion, ce travail a permis de montrer l'efficacité de l'application des algorithmes de colonie de fourmis au diagnostic des systèmes industriels. Il pourra donner lieu à des études supplémentaires sur plusieurs points :

L'une des perspectives immédiates de notre travail est bien sûr d'utiliser notre algorithme de sélection de paramètres avec d'autres classifieurs tels que les réseaux de neurones.

Nous pensons qu'il serait intéressant d'utiliser plus d'une colonie de fourmis pour réaliser la classification de données et la sélection de paramètres pour améliorer la pertinence du vecteur forme.

Finalement, améliorer les algorithmes proposés pour pouvoir développer un module de diagnostic prédictif qui permet de suivre l'évolution du système et anticiper l'évolution d'un mode de fonctionnement normal vers un mode anormal. Cette fonction prédictive permet d'éviter l'apparition de dégâts irréversibles voire l'arrêt du système de production.

REFERENCES BIBLIOGRAPHIQUES

- [Adrot, 2000] O. Adrot. Diagnostic à base de modèles incertains utilisant l'analyse par intervalles: l'approche bornante, Ph.D. de Institut National Polytechnique de Lorraine, France, 2000.
- [Al-Ani, 2005] A. Al-Ani, Feature subset selection using ant colony optimization, *International Journal of Computational Intelligence*, 2 (1) (2005) 53-58.
- [Anguita et al., 2005] D. Anguita, A. Boni, S. Ridella, F. Riveccio, D. Sterpi, Theoretical and Practical Model Selection Methods for Support Vector Classifiers. in L. Wang (Ed.), *Support Vector Machines: Theory and Applications*, Springer Verlag, 2005.
- [Anguita et Bozza, 2005] D. Anguita, G. Bozza, The Effects of Quantization in Support Vector Machines with Gaussian Kernels, *Proc. of the IEEE/INNS Int. Joint Conf. on Neural Networks*, Montreal, Canada, August 2005.
- [Azzag et al., 2004] H. Azzag, C. Guinot, G. Venturini. AntTree : web document clustering using artificial ants. *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, p. 480-484, IOS Press 22- 27 août. Valence. Espagne.
- [Benathmene et Bounouara 2011] Benathmene Imen et Bounouara Rima, La classification binaire et multi-classes par les machines à vecteur de support pour le problème de diagnostic industriel Application « L'unité de laiterie Aurès Batna », Mémoire de fin d'études, Université de Batna, 2011.
- [Blum et Langley, 1997] A. L. Blum and P. Langley, Selection of relevant features and examples in machine learning artificial, *Artificial Intelligence*, 97 (1997) 245- 271.
- [Boser et al., 1992] B. E. Boser, I. M. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144-152, 1992.
- [Boz, 2002] O. Boz, Feature subset selection by using sorted feature relevance, *International Conference on Machine Learning and Application*, Las Vegas City (2002) 147-153.
- [Burges, 1998] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* (2) 1-43. 1998.
- [Casimir, 2003] Roland Casimir. Diagnostic Des Défauts Des Machines Synchrones Par Reconnaissance Des Formes, Ph.D l'école doctorale électronique, électrotechnique, automatique de Lyon, France, 2003.
- [Chang et Lin, 2001] Chang C.-C., Lin C.-J. LIBSVM: a library for support vector machines, apport technique, national taiwan university, 2001. Logiciel disponible en ligne (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>).
- [Chen et al., 2001] Yunqiang Chen, Xiang Zhou, and Thomas S. Huang. One-Class Svm For Learning In Image Retrieval. In *Proc. IEEE Int'l Conf. on Image Processing 2001*, Thessaloniki, Greece.
- [Chen et al., 2002] Chen, M., Kiciman, E., Fratkin, E., Fox, A., and Brewer, E. "Pinpoint: Problem Determination in Large, Dynamic Internet Services." In *Dependable Sys. and Networks (DSN)*, 2002.
- [Colas, 2008] Sonia Colas, Outils d'amélioration de l'accessibilité du web pour les personnes visuellement handicapées, Ph.D de l'université François-Rabelais, Tours, France, 2008.

- [Collobert et Bengio, 2001] R. Collobert and S. Bengio. SVM Torch: Support Vector Machines for Large-Scale Regression Problems. *Journal of Machine Learning Research*, 1:143-160, 2001.
- [Colorni et al., 1991] Colorni A., Dorigo M., Maniezzo V. An Autocatalytic Optimizing Process, Technical Report n. 91-016 Politecnico di Milano, 1991.
- [Dash et Liu, 1997] M. Dash and H. Liu, Feature selection methods for classifications, *Intelligent Data Analysis*, 3 (1) (1997) 131-156.
- [Dazzi et al., 2001] Dazzi, D., Taddei, F., Gavarini, A., Uggeri, E., Negra, R., and Pezzarossa, A., (2001). The Control of Blood Glucose in the Critical Diabetic Patient: A Neuro-Fuzzy Method, *Elsevier Science Journal of Diabetes and Its Complications*, 15, 80-87.
- [Deneubourg et al., 1990] Deneubourg J.-L., Aron, S., Goss, S., et Pasteels, J.-M. The self-organizing ; exploratory pattern of the argentine ant, *Journal on insect Behavior*, 3: 159-168, 1990.
- [Dorigo, 1992] M. Dorigo. Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy, 1992.
- [Duda, 2001] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern classification*, Second Ed. Wiley Interscience Publication (2001).
- [Fasel et Luetin, 2003] B. Fasel and J. Luetin, “Automatic facial expression analysis: a survey,” *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.
- [Ferber, 1995] J. Ferber. *Les systèmes multi-agents : vers une intelligence collective*, Interedition, 1995.
- [Frank et Asuncion, 2010] A. Frank, and A. Asuncion. *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. 2010.
- [Fukunaga, 1990] Keinosuke Fukunaga. *Introduction to statistical pattern recognition* (2nd ed.) (Computer Science and Scientific Computing Series). Academic Press Professional, Inc., San Diego, CA, USA, 1990.
- [Gambardella, 1999] L. Gambardella, E. Taillard, and G. Agazzi. MACS-VRPTW : A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimisation*, pages 63--76. McGraw-Hill, London, UK, 1999.
- [Guérif, 2006] Sébastien Guérif, *Réduction de dimension en Apprentissage Numérique Non Supervisé*, Ph.D de l’université PARIS 13, France, 2006.
- [Guyon et Elisseeff, 2003] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research*, 3 (2003) 1157-1182.
- [Guyon et Elisseeff, 2003] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, vol 3, pp 1157–1182, 2003.
- [Haykin, 1999] S. Haykin. *Neural Networks*. Prentice Hall, 1999.
- [Holldöbler et Wilson, 1990] B. Holldöbler, EO Wilson. *The ants*, Belknap, Harvard University Press, Cambridge, Mass, 1990.
- [Hsu et Lin, 2002] C.W. Hsu, C.J. Lin, A comparison of methods for multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415-425. 2002.
- [Jain et Mao, 1994] A. K. Jain and J. Mao. *Imitating Life*, chapter *Neural Networks and Pattern Recognition*, pages 194-212. Computational Intelligence. IEEE Press, Piscataway, N. J., 1994.

- [Joachims, 1999] T. Joachims. Making large-scale svmlearning practical. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11. 1999.
- [Kadri et al., 2010a] Ouahab Kadri, Hayet Mouss, Adel AbdelhdiI, et Rafik Mahdaoui. Vers une optimisation de l'algorithme anttreestoch, *ICIEM'10 International Conference on Industrial Engineering and Manufacturing*. Batna–Algeria du 09 au 10 Mai 2010.
- [Kadri et al., 2010b] O. Kadri, F. Merah, H. Mouss & A. Abdelhadi. Un algorithme de colonie de fourmis pour la réduction de la taille du vecteur d'état d'un système industriel, *International Conference On Industrial Engineering and Manufacturing ICIEM'10*, May, 9-10, 2010, Batna, Algeria.
- [Kadri et al., 2010c] O. Kadri, M. D. Mouss, L.H. Mouss, F. Merah. A Hybrid Feature Subset Selection Approach based on SVM and Binary ACO. Application to Industrial Diagnosis, *International Journal of Aerospace and Mechanical Engineering*, Volume 4 Number 4 Autumn 2010.
- [Kadri et al., 2011a] Ouahab Kadri, L. Hayet Mouss, M. Djamel Mouss. La sélection de paramètres d'un système industriel par les colonies de fourmis, *Annals. Computer Science Series*, Tome 9, Fasc. 1. 2011.
- [Kadri et al., 2011b] Ouahab Kadri, L. Hayet Mouss, M. Djamel Mouss. Modélisation des ACO par l'application des SMA réactifs, *La conférence internationale : Conception et Production intégrés*, 19-21 Octobre 2011, Oujda, Maroc.
- [Kadri et al., 2012a] O. Kadri, L. H. Mouss, M. D. Mouss & A. Abdelhadi. Reconnaissance des Formes par SVM pour le Diagnostic du Système de Pasteurisation d'une Usine de Lait, *ICIEM'12 International Conference On Industrial Engineering and Manufacturing*, May, 6-7, 2012, Batna, Algeria.
- [Kadri et al., 2012b] Ouahab Kadri, L. Hayet Mouss, M. Djamel Mouss. Fault diagnosis of rotary kiln using SVM and Binary ACO. *The Journal of Mechanical Science and Technology*, vol. 26, no. 2, pp.601-608, 2012.
- [Kalakech, 2011] Mariam Kalakech, *Sélection semi-supervisée d'attributs : Application à la classification de textures couleur*, Ph.D De L'université de Lille 1, France, 2011.
- [Kittler, 1978] J. Kittler, Feature set search algorithms, *Proc. of Pattern Recognition and Signal*, C. H. Chen, editor, Sijhoff and Noordhoff, the Netherlands (1978).
- [Kittler, 1978] J. Kittler. Feature set search algorithms. In C. H. Chen, editor, *Pattern Recognition and Signal Processing*. Sijhoff and Noordhoff, the Netherlands, 1978.
- [Labroche et al., 2002] N. Labroche, N. Monmarché, et G. Venturini. A new clustering algorithm based on the chemical recognition system of ants. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 345–349, Lyon, France, july 2002. IOS Press.
- [Lebart et al., 2000] L. Lebart, A. Morineau, M. Piron, *Statistique Exploratoire Multidimensionnelle*, Dunod, 2000.
- [Lewis et Ringuette, 1994] D. D. Lewis et M. Ringuette. A comparison of two learning algorithms for text classification. In *Proc. of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, 1994.

- [Li et al., 2002] Y. Li, M.J. Pont, N.B. Jones, Improving the performance of radial basis function classifiers in condition monitoring and fault diagnosis applications where ‘unknown’ faults may occur, *Pattern Recognition Letters* (23) 569–577. 2002.
- [Lumer et Faieta, 1994] E.D Lumer, et B. Faieta, Diversity and adaptation in populations of clustering ants, in D. Cli®, P. Husbands, J. Meyer and S. Wilson (eds), *From Animals to Animats 3*, Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior, The MIT press/Bradford Books.1994.
- [Mahdaoui et Mouss, 2008] R. Mahdaoui and L. H. Mouss, NEFDIAG, A new approach for industrial diagnosis by neuro-fuzzy systems: Application to manufacturing system, *Proc. of ERIMA*, 2 (2) (2008) 144-151.
- [Mathieu et al., 2000] Mathieu, p.; Lebedeff, d.; Barlaud, M., Méthode de projection non-linéaire rapide pour la quantification vectorielle, GRETSI, Groupe d’Etudes du Traitement du Signal et des Images,1995.
- [Meunier et al., 1996] B. Bouchon Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, 84(2) :143–153, 1996.
- [Mokhtari, 2007] Aïmed Mokhtari. Diagnostic des systèmes hybrides : développement d’une méthode associant la détection par classification et la simulation dynamique, Ph.D de l’université de Toulouse, France, 2007.
- [Monmarché, 1999] N. Monmarché On Data Clustering with Artificial Ants. In:Freitas AA, editor, *Data Mining with Evolutionary Algorithms:Research Directions – Papers from the AAAI Workshop*, pages 23-26. AAAI Press, 1999.
- [Mouss et al., 2004a] Mouss D., Mouss H., and Mouss N., “DIAPASE: An Industrial Diagnosis Tool by knowldge based Approach for An Agro-Alimentary Production System” ASCC’ 04 The 5th Asian Control Conference 20-23 July, Grand Hyatt Melbourne Australia.
- [Mouss, 2006] Mouss Djamel. Diagnostic et conduite des systèmes de production par approche à base de connaissances, Thèse en Sciences, LAP, Université de Batna, Juin 2006.
- [Narendra et Fukunaga, 1977] P. M. Narendra and K. Fukunaga, A branh and bound algorithm for feature subset selection, *IEEE Transactions on Computers*, 26 (1977) 917-922.
- [Narvaez, 2007] Claudia Victoria Isaza Narvaez. Diagnostic par techniques d’apprentissage floues conception d’une methode de validation et d’optimisation des partitions, Ph.D de l’université de Toulouse, France, 2007.
- [NedjemEddine, 2004] AYAT NEDJEMEDDINE, Sélection De Modèle Automatique Des Machines A Vecteurs De Support: Application A La Reconnaissance D’images De Chiffres Manuscrits, Ph.d de l’école de technologie supérieure, Université du Québec, Canada, 2004.
- [Office, 2001] Office de l’efficacité énergétique. Energy Performance Indicator Report: Fluid Milk Plants. SMRC. Canada, 2001.
- [Ogbeide, 2010] S. O. Ogbeide, Developing an optimization model for CO2 reduction in cement production process, *Journal of Engineering Science and Technology Review*, 3 (1) (2010) 85-88.
- [Ondel et al., 2006] Olivier Ondel, Emmanuel Boutleux, and Guy Clerc. A method to detect broken bars in induction machine using pattern recognition techniques, *IEEE Transactions on Industry Applications*, vol 42, no. 4, pp 916-923, 07 2006.

- [Ondel, 2006] Olivier Ondel. Diagnostic Par Reconnaissance Des Formes : Application A Un Ensemble Convertisseur – Machine Asynchrone, Ph.D l'école doctorale électronique, électrotechnique, automatique de Lyon, France, 2006.
- [Osuna et al., 1997] E. Osuna, R. Freund, and F. Girosi. Improved training algorithm for support vector machines, 1997.
- [Ouadfel et Batouche, 2007] Salima Ouadfel, Mohamed Batouche. An Efficient Ant Algorithm for Swarm-Based Image Clustering, *Journal of Computer Science*, Vol 3, p 162-167, 2007.
- [Parzen, 1962] E. Parzen. On The Estimation Of A Probability Density Function And Mode. *Annals of Mathematical Statistics*, 33 :1065–1076, 1962.
- [Pearson, 1901] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559 - 572, 1901.
- [Phoummavong, 2012] Vimane Phoummavong, Cours de Matériaux de construction, l'Université Nationale du Laos, 2012.
- [Platt et al., 2000] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large Margin DAG's for Multiclass Classification. In: Solla, S.A., Leen, T.K., Müller, K.R.(Eds.), *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 12:547-553. 2000.
- [Plotz et Fink, 2009] T. Plotz and G. A. Fink, “Markov Models for Offline Hand-writing Recognition: A Survey,” *Int. Journal on Document Analysis and Recognition*, vol. 12, no. 4, pp. 269–298, 2009.
- [Rezki et al., 2011] Wail Rezki, L.H.Mouss, O.Kadri. La sélection des paramètres par les algorithmes génétiques, *International Conference on Electronics and Oil ICEO'11 1 Ouargla, Algeria*, Mars 1 - 2, 2011.
- [Rezki et al., 2012] Wail Rezki, L.H.Mouss, O.Kadri, La classification des observations pour le diagnostic industriel d'un four rotatif, *International Conference On Industrial Engineering and Manufacturing ICIEM'12*, May, 6-7, 2012, Batna, Algeria
- [Saint-Jean, 2001] Christophe Saint-Jean. Classification paramétrique robuste partiellement supervisée en reconnaissance des formes, Ph.D de l'université de La Rochelle, France, 2007.
- [Somol et al., 2001] Petr Somol, Pavel Pudil and JiYí Grim, Branch & Bound Algorithm with Partial Prediction for Use with Recursive and Non-Recursive Criterion Forms, *Advances In Pattern Recognition, Lecture Notes in Computer Science Volume 2013/2001*
- [Sophian et al., 2003] Sophian, A., Tian, GY, Taylor, D. and Rudlin, J., “A Feature Extraction Technique Based on Principal Component Analysis for Pulsed Eddy Current NDT ”, *NDT & E International*, Vol. 36, 2003, pp. 37-41.
- [Stoppiglia, 1997] Hervé Stoppiglia, Méthodes statistiques de sélection de modèles neuronaux ; applications financières et bancaires, Ph.D de l'université Pierre et Marie Curie - Paris VI, France, 1997.
- [Tusher et al., 2001] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98(9):5116 – 5121, April 2001.
- [Vandenbroucke et al., 2000] N. Vandenbroucke, L.Macaire, J.-G. Postaire. Unsupervised color texture features extraction and selection for soccer images segmentation. In *IEEE International*

Conference on Image Processing (ICIP'00), volume 2, pages 800–803, Vancouver (Canada), septembre 2000.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[Weiqing et al., 2006] X. Weiqing, Y. Chenyang and W. Liuyi, Binary ant colony evolutionary algorithm, *International Journal of Information Technology*, 12 (3) (2006) 10-20.

[Wilensky, 1999] Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/-netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[Wooldridge et Jennings, 1995] Wooldridge, M., Jennings, N. R. (1995) Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152.

[Yang et al., 2007] Y. Yang, D. Yu and J. Cheng, A fault diagnosis approach for roller bearing based on envelop spectrum and SVM, *Measurement*, 40, 943-50. 2007.

[Yang et Honavar, 1998] J. H. Yang and V. Honavar, Feature subset selection using a genetic algorithm, *IEEE Intelligent Systems*, 13 (2) (1998) 44-49.

[Youn, 2010] E. Youn, Support vector-based feature selection using Fisher's linear discriminant and Support Vector Machine, *Expert Systems with Applications*, 37 (9) (2010) 6148-6156.

[Zhang, 2012] J.H. Zhang. Optimization of Kernel Function Parameters SVM Based on the GA. *Advanced Materials Research*, Trans Tech Publ. 2012.

ANNEXES

LAP D'AUTOMATIQUE ET PRODUCTIVE	ANALYSE DES MODES DE DEFAILLANCE ET DE LEURS EFFETS ET CRITICITE								AMDEC			
	Système : la cimenterie d'Ain-TouTa Sous système : partie Clinkérisation					Date d'analyse : 30/07/2010			PHASE DE FONCTIONNEMENT : En service.			
	Composant	Fonction	Modes de défaillance	Repère	Cause possible	Durée d'indis- ponibilité	Criticité				Information de l'opérateur	Activité compensation
F							G	N	C			
Préchauffeur à cyclone	Préparer la farine cru	Marche perturbé.	1	↑↓ du teneur O2 sortie cyclone A50	10 min	1	1	2	2	Alarme analyseur gaz	-L'opérateur contrôle les variables (A50T1, A50P1, A54T1, A54T2, A54P2, W01P3, W01P1). -demande l'intervention de l'instrumentiste.	-Système filtrage d'alarme. -l'intervention de l'instrumentiste
			2	□ teneur CO sortie cyclone A50	10 min	1	2	2	4	Alarme analyseur gaz	Arrêté électrofiltre.	l'intervention de l'instrumentiste
			3	□ tempé- rature gaz sortie cyclone A50.	120 min	1	1	2	2	Alarme température	□ A50T1sinon agir sur ces variables (O2C, W01T1, V31F1, V01F1, W01S1) et contrôler la variable A54T2.	Surveillant
			4	↑ Pressio n sortie cyclone A50.	70 min	1	1	2	2	Alarme pression.	- ↓↑ la variable O2C - intervention de l'opérateur sur le système.	Surveillant.

LAP D'AUTOMATIQUE ET PRODUCTIQUE	ANALYSE DES MODES DE DEFAILLANCE ET DE LEURS EFFETS ET CRITICITE							AMDEC				
	Système : la cimenterie d'Ain-TouTa Sous système : partie Clinkérisation					Date d'analyse : 30/07/2010		PHASE DE FONCTIONNEMENT : En service.				
	Composant	Fonction	Modes de défaillance	Repère	Cause possible	Durée d'indis- ponibilité	Criticité				Information de l'opérateur	Activité compensation
F							G	N	C			
Préchauffeur à cyclone	Préparer la farine cru	Marche perturbé.	5	↓ tempéra- ture gaz sortie cyclone A52.	120 min	1	1	1	1	Alarme température	↓ la variable A52T1.	/
			6	↓ pressio- n cyclone A52	10 min	2	3	1	6	Alarme pression.	-commandé la variable A52P2. -intervention de l'opérateur sur le système.	l'intervention de Surveillant.
			7	↓ tempéra- ture gaz cyclone A53.	180 min	1	1	1	1	Alarme température	↓ la variable A53T1.	Surveillant
			8	↓ tempéra- ture matière cyclone A53.	120 min	1	1	2	2	Alarme température	↓ la variable A53T2.	Surveillant
			9	↑↓ Pressio- n cyclone A53.	70 min	2	1	1	2	Alarme pression.min ou max	- ↓↑ la variable O ₂ C - intervention de l'opérateur sur le système.	/

LAP D'AUTOMATIQUE ET PRODUCTIQUE	ANALYSE DES MODES DE DEFAILLANCE ET DE LEURS EFFETS ET CRITICITE									AMDEC		
	Système : la cimenterie d'Ain-TouTa Sous système : partie Clinkérisation					Date d'analyse : 30/07/2010				PHASE DE FONCTIONNEMENT : En service.		
	Composant	Fonction	Modes de défaillance	Repère	Cause possible	Durée d'indisponibilité	Criticité				Information de l'opérateur	Activité compensation
F							G	N	C			
Préchauffeur à cyclone	Préparer la farine crue	Marche perturbé.	10	↓ la température gaz cyclone A54	120 min	2	1	2	4	Alarme température gaz	L'opérateur ↓ A54T1 ou bien agir sur les variables (O ₂ C, W01T1, V31F1, V01F1, W01S1). et contrôlé la variable A54T2	-l'intervention de l'instrumentiste
			11	↓ la température matière cyclone A54	90 min	1	1	2	2	Alarme température matière	L'opérateur ↓ A54T2 ou bien agir sur les variables (O ₂ C, A54T1, W01T1, V31F1, V01F1, W01S1)	l'intervention de l'instrumentiste
			12	↑↓ Pression cyclone A54.	30 min	1	1	2	2	Alarme pression.	- ↓↑ la variable O ₂ C - intervention de l'opérateur sur le système.	Surveillant.
Boîte à fumée	Calcination de la farine.	Déplacement de zone.	13	↑↓ du teneur O ₂ boîte à fumée	10 min	2	1	3	6	Alarme analyseur gaz	-L'opérateur agir sur les variables (V31F1, V01F1).	-Système filtrage d'alarme. -l'intervention de l'instrumentiste

LAP D'AUTOMATIQUE ET PRODUCTIQUE	ANALYSE DES MODES DE DEFAILLANCE ET DE LEURS EFFETS ET CRITICITE								AMDEC			
	Système : la cimenterie d'Ain-TouTa Sous système : partie Clinkérisation					Date d'analyse : 30/07/2010			PHASE DE FONCTIONNEMENT : En service.			
Composant	Fonction	Modes de défaillance	Repère	Cause possible	Durée d'indis- ponibilité	Criticité				Information de l'opérateur	Activité compensation	Besoin de l'opérateur
						F	G	N	C			
Boite à fumée	Calcinat ion de la farine.	Déplacem ent de zone.	15	↑ teneur CO boite à fumée	10 min	1	1	2	2	Alarme analyseur gaz	Arrêté électrofiltre.	l'intervention de l'instrumentiste
			16	↑↓ Pressio n sortie boite à fumée.	70 min	2	2	1	4	Alarme pression.	L'opérateur agir sur les deus variable (O ₂ C et O ₂ F) et contrôler la variable W01P1. - intervention de l'opérateur sur le système.	Surveillant.
			17	↑↓ Pressio n entrée boite à fumée.	60 min	1	1	1	1	Alarme pression.	-L'opérateur agir sur les variables (O ₂ C, O ₂ F, W01P3, W01S1). -intervention de l'opérateur sur le système.	Surveillant.
			18	↑ tempéra ture gaz entrée boite à fumée	120 min	1	1	2	2	Alarme température gaz	L'opérateur W01T1 ou bien agir sur les variables (V31F1, V01F1)	l'intervention de l'instrumentiste

LAP D'AUTOMATIQUE ET PRODUCTIVE	ANALYSE DES MODES DE DEFAILLANCE ET DE LEURS EFFETS ET CRITICITE									AMDEC		
	Système : la cimenterie d'Ain-TouTa Sous système : partie Clinkérisation					Date d'analyse : 30/07/2010				PHASE DE FONCTIONNEMENT : En service.		
	Composant	Fonction	Modes de défaillance	Repère	Cause possible	Durée d'indis- ponibilité	Criticité				Information de l'opérateur	Activité compensation
F							G	N	C			
Boîte à fumée	Calcinat ion de la farine.	Déplacem ent de zone.	19	↑↓ débit gaz	1 min	1	1	2	2	Afficheur débit gaz	↓ la variable V01	Surveillant.
Four	Clinkéri- sation	Mauvaise Clinkérisat ion	20	↑↓ débit gaz	1 min	1	1	1	1	Afficheur débit gaz	L'opérateur réglé les variables (A50T1, A54T1, A54T2, O ₂ F, W01T1, CCS, W01X1).	Surveillant.
			21	moment four.	60 min	2	3	1	6	Afficheur moment four.	L'opérateur réglé les variables (O ₂ F, V01F1, CCS, W01S1).et contrôlé les deux variables (PAL et FCAO).	/
			22	vitesse four.	60 min	1	1	2	2	Afficheur vitesse four.	L'opérateur réglé les variables (O ₂ F).et contrôlé les variables (A50T1, A54T1, A54T2, W01P1, CCS, W01X1, PAL, FCAO).	/

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 1/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Echangeur de chaleur : section de préchauffage et de chauffage du produit (échangeur à plaques)	circulation, préchauffage et chauffage du produit laitier	fuite (entre deux plaques, entre le bâti fixe et la bride de tubulure,...)	Joints endommagés, perforation, fissures (corrosion), chute brutale de la température	diminution du débit du produit		3	3	1	9	
		bouchage (ou blocage)	corps étrangers, encrassement			3	3	1	9	
		perte d'efficacité de l'échange de chaleur (la puissance thermique échangée diminue)	encrassement, dépôts	diminution de la température de chauffage du produit.		3	3	2	18	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 2/ 9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Echangeur de chaleur : section de chauffage de l'eau	circulation de la vapeur destinée au chauffage de l'eau	fuite (entre deux plaques, entre le bâti fixe et la bride de tubulure,...)	Jointes endommagés, perforation, fissures (corrosion), chute brutale de la température	diminution du débit du produit		2	3	1	6	
		bouchage (ou blocage)	corps étrangers, encrassement			2	3	1	6	
		perte d'efficacité de l'échange de chaleur (la puissance thermique échangée diminue)	encrassement, dépôts	diminution de la température de chauffage du produit.		2	3	2	12	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 3/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Echangeur de chaleur : section de refroidissement	circulation et refroidissement du produit laitier	fuite (entre deux plaques, entre le bâti fixe et la bride de tubulure,...)	Joint endommagés, perforation, fissures (corrosion), chute brutale de la température	diminution du débit du produit		1	4	2	8	
		bouchage (ou blocage)	corps étrangers, encrassement			1	4	2	8	
		perte d'efficacité de l'échange de chaleur (la puissance thermique échangée diminue)	encrassement, dépôts	diminution de la température de chauffage du produit.		1	4	2	8	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 4/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Echangeur de chaleur : section de chauffage	circulation et chauffage de l'eau destinée au chauffage du produit laitier	fuite (entre deux plaques, entre le bâti fixe et la bride de tubulure,...)	Joint endommagés, perforation, fissures (corrosion), chute brutale de la température	diminution du débit du produit		3	3	1	9	
		bouchage (ou blocage)	corps étrangers, encrassement			3	3	1	9	
		perte d'efficacité de l'échange de chaleur (la puissance thermique échangée diminue)	encrassement, dépôts	diminution de la température de chauffage du produit.		3	3	2	18	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 5/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Echangeur de chaleur : section de refroidissement	circulation de l'eau glacée destinée au refroidissement du produit laitier	fuite (entre deux plaques, entre le bâti fixe et la bride de tubulure,...)	Joint endommagés, perforation, fissures (corrosion), chute brutale de la température	diminution du débit du produit		3	3	1	9	
		bouchage (ou blocage)	corps étrangers, encrassement			3	3	1	9	
		perte d'efficacité de l'échange de chaleur (la puissance thermique échangée diminue)	encrassement, dépôts	diminution de la température de chauffage du produit.		3	3	2	18	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 6/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
vanne réglante commandée par un servomoteur pneumatique	régulation de la pression de la vapeur en fonction de la température de chauffage du produit laitier Tc, la position de la vanne est fixée par le système de régulation	blocage en position	défaut mécanique défaut dû au système de régulation défaut électrique (servomoteur)	la pression de la vapeur garde une valeur constante, pas de compensation d'une variation de la température de chauffage du produit		3	3	1	9	
		fermeture (ouverture) intempestive	défaut mécanique défaut dû au système de régulation perte de pression d'air de régulation	sollicitation du contrôle, arrêt du débit du produit		3	3	1	9	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE							AMDEC PROCESSUS		
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 7/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
pompe centrifuge	circulation du produit laitier	arrêt de la pompe	forte cavitation, défaut mécanique, défaut électrique (moteur), perte d'alimentation électrique	arrêt du débit du produit		1	2	1	2	
Régulateur de débit FC	adaptation du passage du produit à travers le régulateur au débit fixé pour le produit	dérive de la valeur de consigne	erreur humaine (erreur de l'opérateur dans le réglage de la valeur de consigne)	stabilisation du débit du produit à une valeur anormale engendrée par la dérive		2	1	3	6	
		défaillance à la sollicitation	Défaillance première du régulateur	pas de compensation d'une variation du débit du produit (en amont du traitement)		2	1	3	6	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse		Page 8/9		
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
Circuit d'alimentation en vapeur	alimentation en vapeur pour le chauffage de l'eau	débit insuffisant de vapeur		pas de compensation d'une diminution de Tc, recyclage et donc arrêt du débit du produit		1	2	1	2	
Circuit d'alimentation en eau glacée	alimentation en eau glacée pour le refroidissement du produit laitier	débit insuffisant d'eau glacée		augmentation de la température de refroidissement, réglage		1	2	1	2	
Circuit d'alimentation en lait prétraité (destinée aux produits laitiers : LPC, LS)	alimentation du groupe pasteurisateur destiné à la pasteurisation des produits laitiers	variation du débit du produit en amont	dysfonctionnement de la pompe de refoulement du produit des réservoirs de stockage,...	variation du débit du produit compensable par la régulation		2	2	3	12	

LAP	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE						AMDEC PROCESSUS			
	Système : Production du lait Sous système : Pasteurisation du lait			Phase de fonctionnement : MARCHE NORMALE		Date de l'analyse	Page 9/9			
Elément	Fonction	Mode de défaillance	Cause de la défaillance	Effet de la défaillance	Détection	Criticité				Observation
						F	G	N	C	
vanne d'alternance	permettre le passage du produit laitier dans la section de préchauffage	erreur de fermeture	erreur de l'opérateur	diminution de la température de chauffage du produit		1	3	1	3	

1 Présentation générale

Dans cette annexe, nous allons présenter un modèle multi-agents basé sur les algorithmes génétiques pour la sélection de paramètres. La figure A.1 présente un exemple d'application sur la base de données IRIS.

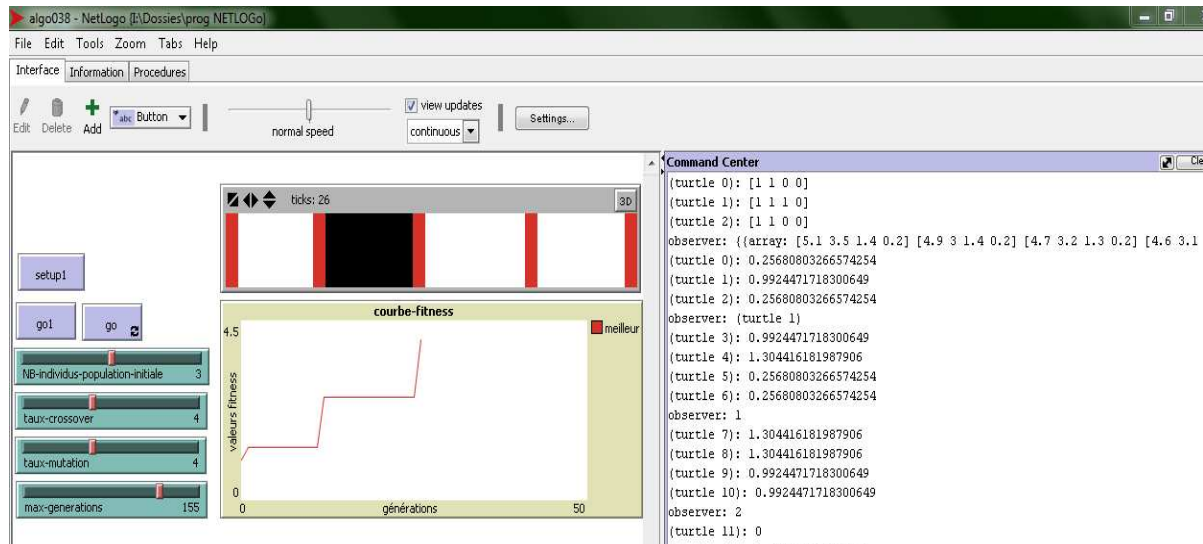


Fig. A.1 : Interface graphique du modèle

L'idée principale est de trouver une chaîne de bits des uns et de zéros. Ensuite, nous éliminons les colonnes qui correspondent aux zéro dans le tableau qui contient les données IRIS. La solution trouvée est évaluée en utilisant une fonction d'adaptation. La valeur retournée reflète la qualité de la solution.

2 Description du modèle :

Le bouton SETUP1 permet de créer une population initiale aléatoire de solutions, le nombre des solutions égales est indiquée par la variable interface *NB-INDIVIDUS-POPULATION-INITIALE*. L'environnement NetLogo nous permet de représenter chaque solution par un agent Turtle sachant que chaque agent reste jusqu'à la prochaine génération.

Le bouton GO1 permet d'évaluer tous les individus de la population initiale en utilisant la fonction d'adaptation. Cette étape permet aussi de lire les données de fichier IRIS dans un tableau nommé « Bits ».

Tableau A.1: La procédure de configuration

```
to setup1
  clear-all
  create-turtles NB-individus-population-initiale
  [
    set case2 n-values 4 [one-of [0 1]]
  ]
  hide-turtle
  set num-turt 0
  cas-particulier
end
```

Tableau A.2 : La procédure d'initialisation

```
to go1
  set gen 0
  set identiti-turt 0
  lire-donnees "iris.txt
  repeat NB-individus-population-initiale [ go2 ]
  determiner-la-meilleur
  courbe
  show meilleur
end
```

Tableau A.3 : La procédure d'affichage

```
to go2
  ask turtle identiti-turt [ set ds case2 ]
  affichage
  set stopturtle 0
  go3
end
```


Tableau A.4 : La fonction d'adaptation

```

to go3
ask turtle identiti-turt
[
  set ds case2 ifelse (length (remove 0 ds) = 1) or (length (remove 0 ds) = 0) or
(length (remove 0 ds) = 4) [set fitness 0 show fitness]
[
  calcule-centr-de-gravite-generale
  calcule-centr-de-gravite-classe
  intra-classe
  inter-classe
  inverse-matrice
  multiplication-intraclasse-interclasse
  trace-matrice
  set identiti-turt identiti-turt + 1
]
]
End

```

Le résultat de cette fonction est la création d'une nouvelle génération à partir de l'ancienne génération. Sur cette nouvelle génération, on applique les opérations de sélection, croisement, mutation, et d'évaluation.

Tableau A.5 : Création d'une nouvelle génération

```

to go
  if (gen = max-generations)or( ds1 > 42) [ stop
]
  cree-prochaine-generation
  set gen gen + 1
  show gen
  tick
  courbe
end

```

La procédure précédente permet aussi de tester le critère d'arrêt.

Tableau A.6 : La procédure de croisement

```

to croisement
  let point-de-croisement random (5)
  set fils-bits1 (sentence (sublist select-turt1 0 point-de-croisement) (sublist select-turt2 point-de-croisement 4))
  set fils-bits2 (sentence (sublist select-turt1 0 point-de-croisement) (sublist select-turt2 point-de-croisement 4))
end

```

L'algorithme génétique s'exécute jusqu'à ce qu'une solution est trouvée ou le nombre de génération maximal déterminé par le curseur MAX-GENERATIONS est atteint. La meilleure solution dans chaque génération est affichée dans la vue qui est divisée en quatre régions séparées par une bordure rouge, chaque région représente un agent Patch.

Tableau A.7 : La procédure de mutation

```

to mutation
  set case2 map [ifelse-value (random-float 100.0 < taux-mutation) [1 - ?] [?]]
case2
end

```

L'algorithme génétique s'exécute jusqu'à ce qu'une solution est trouvée ou le nombre de génération maximal déterminé par le curseur MAX-GENERATIONS est atteint. La meilleure solution dans chaque génération est affichée dans la vue qui est divisée en quatre régions séparées par une bordure rouge, chaque région représente un agent Patch.

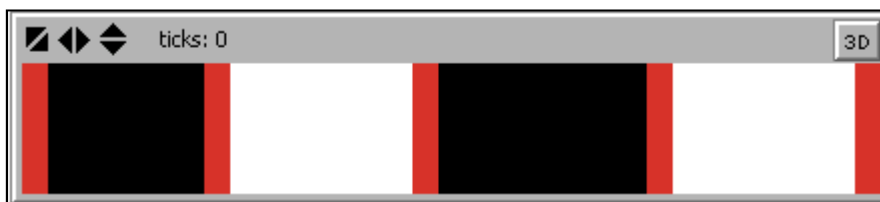


Fig. A.2 : Présentation graphique de la solution