

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



Université de Batna
Faculté de Technologie
Département de Génie Industriel



Thèse

Présentée au Laboratoire d'Automatique et Productique

En vue de l'obtention du diplôme de
Doctorat en Sciences en Génie Industriel

Présentée par

Adel ABDELHADI

(Magistère en Informatique option : Informatique Industrielle)

Thème

**Vers une approche des systèmes multi-agents et méthodes
d'émergence pour la maintenance systématique**

Directrice de Thèse : **Pr. L. Hayet MOUSS**

Soutenue le : 29/06/2015

Devant le jury composé de :

Président	: MOUSS Med Djamel	Prof	Université de Batna
Rapporteur	: MOUSS Leila Hayet	Prof	Université de Batna
Examineurs	: BENMOHAMMED Mohamed	Prof	Université de Constantine
	: BATOUCHE Med Chawki	Prof	Université de Constantine

Remerciements

Il serait ingrat de renier le fait qu'une thèse, reste malgré tout, une œuvre collective. La mienne est l'aboutissement de plusieurs années de travail, de patience et de persévérance. Durant toutes ces années, j'ai été assisté et conseillé par plusieurs personnes, auxquelles je dois leurs adresser mes plus sincères remerciements.

C'est à **Mme Leila Hayet Mouss**, professeur à l'Université Batna à qui revient le mérite d'avoir dirigé ce travail. Elle m'a offert d'excellentes idées et perspectives sur le travail tout au long de son évolution et n'a jamais cessé de me conseiller en me permettant de concrétiser mes idées librement. Qu'elle trouve ici l'expression de mes vifs remerciements.

Je témoigne également toute ma reconnaissance à **Mr. MOUSS Med Djamel**, professeur à l'université de Batna pour l'honneur qu'il me fait de présider ce jury.

Je tiens à exprimer toute ma gratitude à **Mr. BENMOHAMMED Mohamed**, professeur à l'université de Constantine pour avoir bien voulu juger ce travail et faire partie de mon jury de thèse.

Je tiens également à présenter mes sincères remerciements à **Mr. BATOUCHE Med Chawki**, professeur à l'université de Constantine, pour leur participation à l'évaluation scientifique de ce travail.

Je voudrai également remercier mes amis, collègues thésards, qui m'ont soutenu et encouragé.

Enfin mes remerciements vont également à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, mais également à Kadri Ouahab pour son encouragement quotidien. Je consacre une mention particulière à Touta pour sa confiance et ses encouragements.

A mon père Mohamed

A ma mère Mansoura

A ma femme Fatima

A mes enfants, Mehdi, Hidaya Kaouther et Ghadir

A mes frères, Farid, Mourad et Khaled

A mes sœurs, Saliha, Djamila, Mona et Nora

Je dédie ce travail

Adel.

Résumé

L'objectif principal de notre thèse est de proposer des approches hybride basées sur les SMA d'un côté, et l'exploitation de méthodes d'émergences tels que les Algorithmes Génétiques (AG) et les algorithmes du Système Immunitaire Artificiel (SIA), pour mieux intégrer des politiques de maintenance préventive systématique dans les ateliers d'ordonnancement. L'objectif est de minimiser les temps d'exécutions pendant le déroulement de l'ordonnancement.

Nous avons utilisé des opérateurs avancés tels que ; la clé aléatoire pour le codage, la sélection de classement, le croisement uniforme et la mutation ponctuelle unique.

Par ailleurs, nous avons défini une nouvelle procédure de calcul d'affinité dans le cadre de l'approche FSHMAIA. Cette procédure est basée sur le calcul d'un rapport de similarité des anticorps. Pour cela, nous avons proposé un algorithme basé sur quatre critères pour calculer le rapport de similarité.

Nous avons également évalué les adaptations de certaines heuristiques bien connues, y compris Johnson ($m / 2, m / 2$), NEH_H , PCDT et PLDT,

Un benchmark a été établi pour évaluer les algorithmes. Le benchmark contenu jusqu'à 100 jobs et 8 étages. Tous les résultats ont montré que le FSHMAIA donne les meilleurs résultats par rapport aux autres algorithmes, dans les trois politiques de la MP.

L'originalité de cette thèse réside dans l'utilisation des SMA avec les AG et les SIA dans l'intégration des politiques de maintenance préventive systématique dans un ordonnancement de type flow shop hybride.

Mots clés : *Systèmes multi agents, Emergence, maintenance systématique, Système immunitaire artificiel, Algorithme génétique, Ordonnancement, Flow shop hybride.*

Abstract

The main objective of our thesis is to propose hybrid approaches based on MAS on one side, and the exploitation of emergences methods such as Genetic Algorithms (GA) and the algorithms of Artificial Immune System (AIS) to better integrate systematic preventive maintenance policies in scheduling workshops.

The objective is to minimize the time of executions during the course of scheduling.

We use advanced operators; the random key for encoding, the selection of classification, the uniform crossover and single point mutation.

Furthermore, we have defined a new affinity calculation procedure within the FSHMAIA approach. This procedure is based on the calculation of similarity ratio for each antibody. For this, we have proposed an algorithm based on four criterions for calculating the ratio of similarity.

We also evaluated the adaptations of some well-known heuristics, including Johnson ($m / 2, m / 2$) NEH_H , PCDT and PLDT,

A benchmark has been established to evaluate the algorithms. The benchmark content up to 100 jobs and 8 stages. All results have shown that the FSHMAIA gives the best results compared to other algorithms in the three policies of PM.

The originality of this thesis lies in the use of MAS with GA and AIS in the integration of systematic preventive maintenance policies in a hybrid flow shop scheduling.

Keywords: *Multi Agent system, Emergence, Systematic maintenance, Artificial Immune System, Genetic Algorithm, Scheduling, Hybrid flow shop.*

ملخص

الهدف الرئيسي من هذه الأطروحة هو اقتراح نهج الهجين مبني على أساس نظام متعدد الوكلاء (MAS) من جهة ، واستغلال طرق التولد (EMERGENCES) مثل الخوارزميات الوراثة (GA) و خوارزميات الجهاز المناعي الاصطناعي (AIS) لتحسين دمج سياسات الصيانة الوقائية المنتظمة في ورشات عمل الجدولة . الهدف هو تقليل وقت العمل خلال الجدولة.

لقد استخدمنا عوامل متقدمة مثل: استعمال المفتاح عشوائي للترميز، الاختيار التصنيفي، التهجين الموحد وطفرة النقطة الواحدة.

علاوة على ذلك، فإننا قد حددنا إجراء حساب تقارب جديد ضمن نهج FSHMAIA ، ويستند هذا الإجراء على حساب وجود الأجسام المضادة لنسبة التشابه. لهذا، اقترحنا خوارزمية على أساس أربع معايير لحساب نسبة التشابه.

لقد قمنا كذلك بتقييم التعديلات لبعض الخوارزميات المعروفة، مثل جونسون ($Johnson, m/2, m/2$)، NEH_H ، PCDT و PLDT،

وقد أنشئنا معيارا لتقييم الخوارزميات، حيث أن محتوى التقييم يصل إلى 100 عمل و 8 طوابق. وقد أظهرت النتائج أن نهج FSHMAIA يعطي أفضل النتائج مقارنة بخوارزميات أخرى في السياسات الثلاث من الصيانة الوقائية المنتظمة.

أصالة هذه الرسالة تكمن في دمج نظام متعدد الوكلاء مع الخوارزميات الوراثة و خوارزميات الجهاز المناعي الاصطناعي في دمج سياسات جدولة الصيانة وقائية في ورش التدفق الهجين.

كلمات البحث: نظام متعدد الوكلاء، التولد، الصيانة الروتينية، الجهاز المناعي الاصطناعي، الخوارزمية الوراثة، الجدولة، ورشة التدفق الهجين.

Table des matières

Table des matières.....	8
Liste des figures.....	12
Liste des tableaux.....	14
Liste des abréviations et des sigles.....	15

Introduction générale

1.1 Introduction.....	18
1.2 Contexte de travail.....	19
1.3 Organisation de la thèse.....	20
1.4 Contributions	21

Chapitre 1 : Contexte et Problématique

1.1 Introduction.....	25
1.2 Les SMA dans le domaine industriel.....	25
1.3 Quelques études faites sur les méthodes d'émergence	29
1.4 Les applications des systèmes immunitaires artificiels.....	30
1.5 Etat de l'art sur le problème d'ordonnancement Flow shop.....	31
1.6 Critiques et limites des travaux existants.....	32
1.7 Synthèse.....	33

Chapitre 2 : Système Multi-Agents

2.1 Introduction.....	37
2.2 Agents.....	37
2.2.1 Définitions d'un agent.....	37
2.2.2 Architecture d'un agent.....	38
2.2.3 Propriétés des agents	39
2.2.4 Classes d'agents	40

2.2.4.1	Agents réactifs.....	41
2.2.4.2	Agent cognitif.....	41
2.2.5	Normes de communication entre les agents	42
2.2.6	Agents et Objets	43
2.3	Systèmes multi-agents.....	43
2.3.1	Historique des systèmes multi agents.....	43
2.3.2	Définition d'un système multi agents.....	44
2.3.3	Types de systèmes multi agents	46
2.3.4	Propriété des systèmes multi-agents	46
2.3.5	Plates-formes multi-agents	47
2.4	Conclusion	48

Chapitre 3 : Emergences et Métaheuristiques

3.1	Introduction.....	51
3.2	Emergence et Auto-Organisation.....	51
3.3	Emergence	51
3.3.1	Historique.....	51
3.3.2	Définition d'émergence.....	52
3.3.3	Caractéristiques de l'émergence	52
3.3.4	L'émergence dans le domaine d'informatique	53
3.3.5	Axes de recherches scientifiques de l'émergence.....	55
3.3.6	Exemples de phénomènes émergents	56
3.4	Auto-organisation	56
3.4.1	Historique et Définition.....	57
3.4.2	Caractéristiques de l'auto-organisation	58
3.4.3	Combinaison entre l'émergence et l'auto-organisation	59
3.5	Stigmergie	60
3.5.1	Définition	60
3.6	Auto-organisation et SMA	61
3.7	Quelques techniques Métaheuristiques.....	62
3.7.1	Algorithmes génétiques.....	62
3.7.2	Réseaux neuronaux dynamiques	63
3.7.3	Algorithmes de colonies de fourmis	64
3.7.4	Optimisation par essaims particulaires	64
3.8	Conclusion	66

Chapitre 4 : Algorithmes du Système Immunitaire Artificiel

4.1	Introduction.....	69
4.2	Systèmes immunitaires biologiques.....	69
4.2.1	Historique et définitions.....	69
4.2.2	Notions du système immunitaire	69
4.2.3	Immunité innée et adaptative.....	71
4.2.4	Réponses immunitaires.....	71
4.2.5	Caractéristiques du système immunitaire.....	72
4.2.6	Méthodes immunitaires	73
4.2.6.1	Méthode de la sélection Négative/Positive.....	73
4.2.6.2	Méthode de la sélection clonale.....	73
4.2.6.3	Méthode du danger.....	74
4.2.7	Développement d'une réponse immunitaire.....	75
4.3	Systèmes immunitaires artificiels.....	76
4.3.1	Définition du SIA	76
4.3.2	Spécification des SIA	76
4.4	Algorithmes du SIA	79
4.4.1	Algorithme de la sélection négative/positive.....	79
4.4.2	Algorithme de la sélection clonale.....	80
4.4.3	Algorithme de réseaux immunitaire.....	82
4.5	Systèmes immunitaires et algorithmes immunitaires.....	83
4.6	Conclusion.....	83

Chapitre 5 : Méthodes de résolution du problème d'ordonnancement de type flow shop

5.1	Introduction.....	87
5.2	Maintenance préventive systématique.....	87
5.2.1	Politique I: Maintenance préventive à des intervalles de temps fixes.....	87
5.2.2	Politique II: Modèle de la période optimale	89
5.2.3	Politique III: Maintenir un seuil minimum de fiabilité	90
5.3	Ordonnancement d'un atelier flow shop hybride.....	91
5.4	Méthodes de résolution du problème d'ordonnancement Flow shop.....	92
5.4.1	Algorithme de Johnson	92
5.4.2	Algorithme NEH_H	94
5.4.3	Algorithme de PCDT et PLDT.....	94

5.5	Conclusion.....	95
Chapitre 6 : Approches développées pour l'intégration de politiques de maintenance systématique dans l'ordonnancement		
6.1	Introduction.....	98
6.2	Approche FSHMAGA proposée	98
6.2.1	Organigramme de l'approche FSHMAGA proposée	98
6.2.2	Algorithme de l'approche FSHMAGA proposée	100
6.2.3	Schéma de codage et Opérateurs de l'approche FSHMAGA proposée	101
6.2.3.1	Clé aléatoire.....	101
6.2.3.2	Stratégie d'élite.....	102
6.2.3.3	Mécanisme de sélection.....	102
6.2.3.4	Croisement uniforme paramétré.....	103
6.2.3.5	Mutation ponctuelle unique.....	104
6.2.4	Conclusion.....	104
6.3	Approche FSHMAIA proposée	105
6.3.1	Organigramme de l'approche FSHMAIA proposée	105
6.3.2	Algorithme de l'approche FSHMAIA proposée	107
6.3.3	Schéma de codage et Opérateurs de notre approche FSHMAIA.....	108
6.3.3.1	Initialisation.....	108
6.3.3.2	Calcul de la fonction d'objective.....	108
6.3.3.3	L'affinité.....	109
6.3.3.4	Algorithme de Calcul d'affinité.....	109
6.3.3.5	Exemple d'application de l'algorithme de Calcul d'affinité.....	111
6.3.4	Conclusion	112
6.4	Résultats et discussion.....	113
6.4.1	Réglage des paramètres.....	113
6.4.2	Génération de données.....	114
6.5	Résultats expérimentaux.....	117
6.6	Conclusion.....	122
	Conclusion générale.....	125
	Références bibliographiques.....	128
	Annexes.....	140

Liste des figures

Chapitre 1

Figure 1.1 Architecture générale du système MAGIC.....	26
Figure 1.2 Coopération/négociation entre agents Machine et Maintenance.....	27
Figure 1.3 Architecture générale du système OASIS.....	28

Chapitre 2

Figure 2.1 Agent dans un environnement.....	38
Figure 2.2 Représentation d'un agent.....	39
Figure 2.3 Agent réactif.....	41
Figure 2.4 Un agent cognitif.....	42
Figure 2.5 Représentation d'un système multi agents.....	45

Chapitre 3

Figure 3.1 Axes de recherches scientifiques de l'émergence.....	55
Figure 3.2 Auto organisation.....	57
Figure 3.3 Emergence et auto-organisation combinées	59
Figure 3.4 Mécanisme de la stigmergie.....	61
Figure 3.5 Principe de fonctionnement des AG.....	63
Figure 3.6 Réseau de neurones à 3 couches.....	63
Figure 3.7 Principe de fonctionnement des ACF	64
Figure 3.8 Optimisation par essaim particulaire (OEP).....	65

Chapitre 4

Figure 4.1 Structure d'un antigène.....	70
Figure 4.2 Types de réponses immunitaires.....	71
Figure 4.3 Méthode de sélection Négative/Positive.....	73
Figure 4.4 Processus de sélection clonale.....	74
Figure 4.5 Développement d'une réponse immunitaire	75
Figure 4.6 Framework pour les SIA.....	77
Figure 4.7 Espace des formes.....	77
Figure 4.8 Niveau d'affinité entre l'antigène et l'anticorps.....	78
Figure 4.9 Complémentarité entre antigène et anticorps.....	78
Figure 4.10 Algorithme de sélection Négative.....	79

Chapitre 5

Figure 5.1 Diagramme de Gantt de la solution pour l'exemple donné	88
Figure 5.2 Diagramme de Gantt de la solution pour l'exemple dans la politique II.....	90
Figure 5.3 Schéma général d'un atelier flow shop hybride.....	92

Chapitre 6

Figure 6.1 Organigramme de l'approche FSHMAGA proposée	99
Figure 6.2 Solution codé en utilisant la représentation CA.....	102
Figure 6.3 Procédure de croisement uniforme paramétré.....	103
Figure 6.4 Procédure de mutation ponctuelle	104
Figure 6.5 Organigramme de l'approche FSHMAIA proposée	105
Figure 6.6 Diagramme de Gantt de l'anticorps candidat au premier étage.....	111
Figure 6.7 Diagramme de Gantt de l'anticorps le plus connu au premier étage.....	111
Figure 6.8 Graphe de meilleures valeurs de PRE entre les différents algorithmes	120
Figure 6.9 Graphe des valeurs de PRE pour le facteur nombre de jobs.....	121
Figure 6.10 Graphe des valeurs de PRE pour le facteur nombre d'étages.....	121
Figure 6.11 Graphe des valeurs de PRE pour le facteur politique de la MP.....	122

Liste des tableaux

Chapitre 3

Tableau 3.1 L'émergence dans le domaine informatique.....	54
---	----

Chapitre 4

Tableau 4.1 Comparaison entre systèmes immunitaires et algorithmes immunitaires.....	83
--	----

Chapitre 5

Tableau 5.1 Temps de traitement pour une seule machine avec $n = 3$	88
Tableau 5.2 Temps de traitement pour une seule machine avec $n = 4$	90

Chapitre 6

Tableau 6.1 Niveaux de paramètres du FSHMAIA.....	114
Tableau 6.2 Niveaux de paramètres du FSHMAGA.....	114
Tableau 6.3 Facteurs et leurs niveaux.....	115
Tableau 6.4 Valeurs de θ correspondant à la variable x_i	116
Tableau 6.5 Valeurs de PRE des algorithmes. pour la politique I.....	117
Tableau 6.6 Valeurs de PRE des algorithmes. pour la politique II.....	118
Tableau 6.7 Valeurs de PRE des algorithmes. pour la politique III.....	119

Liste des abréviations et des sigles

AG	Algorithme Génétique
Alg_{sol}	Valeur de la fonction objective obtenue pour un algorithme donné pour une instance
AR	Affinité de Réglage
APC	Anticorps le Plus Connu
CA	Clé Aléatoire
CUP	Croisement Uniforme Paramétrée
D_{MP}	Durée de la Maintenance Préventive
FSHMAIA	Flow shop Scheduling Hybrid Multi Agent Immune Artificial
FSHMAGA	Flow shop Scheduling Hybrid Multi Agent Genetic Algorithm
IA	Intelligence Artificielle
IAD	Intelligence Artificielle distribuée
Min_{sol}	Bonne solution obtenue pour chaque instance
MC	Maintenance Corrective
MP	Maintenance Préventive
MPS	Maintenance Préventive Systématique
MPU	Mutation ponctuelle unique
NEH	Nawaz-Enscore-Ham
N_r	Meilleurs chromosomes de la population actuelle,
OFSH	Ordonnancement Flow shop hybride
OFSCD	Ordonnancement d'un Flow Shop avec des Contraintes de Disponibilité
P_{j,i}	Temps de traitement de chaque job j à chaque étage i
PCDT	Plus Court Durée de Traitement
PLDT	Plus Longue Durée de Traitement

Pd_i	Poids de facteurs de similarité
P_m	Probabilité de la Mutation
PRE	Pourcentage Relatif de L'Ecart
SA	Seuil d'Affinité
SIA	Système Immunitaire Artificiel
SMA	Système Multi Agents
T_{MPF}	Temps la Maintenance Préventive Fixe
T_r	Nombre d'unités de temps qui prendre la réparation
T_p	Nombre d'unités de temps de la MP
T_{MP}	Intervalle entre deux MP consécutive
T_{MPop}	Intervalle Optimal de la Maintenance Préventive
TP	Taille de la Population
x_i	Variable artificielle

Introduction Générale

Introduction Générale

1.1 Introduction

Les industriels habitués, dans les années passées, à des marges relativement confortables, se trouvent actuellement face à un marché de plus en plus concurrentiel, sur lequel la compétitivité est conditionnée, dans une mesure de plus en plus importante, par l'exploitation optimale des moyens humains et matériels de l'entreprise. Dans ce contexte la maintenance industrielle connaît actuellement un essor spectaculaire, dû à la possibilité de puiser dans les ressources existantes de l'entreprise afin d'améliorer ses performances.

A partir des années 80, des problèmes plus compliqués, des contraintes plus élevées apparaissent, et les approches utilisées ne répondaient plus aux besoins. Les problèmes appropriés aux humains, incluent d'autres règles telles que la collaboration, l'apprentissage et la distribution des connaissances.

Ces connaissances peuvent être partagées sur un groupe de personnes. Par exemple, pour la conception de machines évoluées, les personnes se coordonnent de façon à faire appel à plusieurs compétences (mathématiciens, informaticiens, designers, ...etc.). Cette approche de concevoir des systèmes a donné naissance à un nouveau domaine de l'Intelligence Artificielle Distribuée (IAD) : Les *Systèmes Multi-Agents* (SMA), qui sont des systèmes utilisant des entités intelligentes (agents) autonomes qui travaillent en collaboration pour compléter un objectif proposé.

L'utilisation de l'émergence et de l'auto-organisation concerne d'ailleurs un certain nombre de méthodes métaheuristiques, comme notamment, le calcul évolutionnaire, les réseaux neuronaux, les algorithmes de colonie de fourmis et l'optimisation par essaim particuliers.

L'exploration du corps humain a régulièrement provoqué une très haute réflexion pour sa grande complexité. Nous présenterons dans le cadre de notre recherche une méthode d'émergence très spécifique qu'est le système immunitaire humain. Pendant une longue durée, ce dernier est resté un secret difficile à dénouer pour les chercheurs.

Ce n'est qu'avec les progrès de la microbiologie et la génétique moléculaire que la conception de ce système est devenue réalisable [87].

La majorité des travaux touchant les relations Production/Maintenance utilisent des approches probabilistes dont l'objectif est de définir le meilleur instant pour organiser une action de maintenance en fonction d'un arrangement entre le coût de maintenance et le danger d'altération de disponibilité des machines.

L'objectif principal de notre thèse est de proposer des approches hybride basée sur les SMA d'un côté, et l'exploitation de méthodes d'émergences tels que les Algorithmes Génétiques (AG) et les algorithmes du Système Immunitaire Artificiel (SIA), pour mieux intégrer des politiques de maintenance préventive systématique dans les ateliers d'ordonnancement. L'objectif est de minimiser les temps d'exécutions pendant le déroulement de l'ordonnancement.

1.2 Contexte de travail

L'utilisation des techniques de l'IA a fait l'objet de nombreuses recherches. La possibilité de maintenir sans modéliser le système, dans un environnement ou celui-ci est soumis à des adaptations et des reconfigurations permanentes, présente un intérêt évident pour les industriels.

Une approche traditionnelle en IA est de proposer des méthodes de résolution de problèmes en s'inspirant des stratégies qu'ont développées les systèmes naturels. Ces méthodes reposent le plus souvent sur une formalisation à priori du domaine du problème. Dans les domaines dynamiques et incertains comme la maintenance dans les systèmes industriels, cette formalisation devient à priori difficile et requiert une capacité adaptative accrue.

Les tendances actuelles ainsi que les nécessités croissantes et les contraintes progressives ont conduit à l'émergence d'un nouvel axe de l'IA qui est vite devenu l'un des axes les plus importants de l'informatique.

Les systèmes multi-agents ont apporté une nouvelle image et une nouvelle manière de concevoir des solutions à des problèmes qui auparavant étaient irréalisables ou très difficiles à résoudre. Les systèmes multi agents permettent d'appréhender la complexité de la résolution de ce genre de problèmes.

Le choix de l'utilisation des SMA à base de méthodes d'émergence pour l'intégration des politiques de maintenance préventive systématique dans

l'ordonnancement, est justifié par certaines caractéristiques très intéressantes retrouvées parmi les nombreuses propriétés des SMA et de ces méthodes d'émergence et qui répondent aux nécessités et attentes de notre domaine de recherche.

1.3 Organisation de la thèse

Notre thèse est organisée de la manière suivante :

Après une introduction générale décrivant le contexte de notre travail, le chapitre 1, ***Contexte et problématique***, présente une recherche bibliographique sur les SMA et les méthodes d'émergence dans le domaine industriel. Celle-ci nous a permis de positionner notre problématique de recherche.

Le chapitre 2, ***Systèmes Multi-Agents***, vise à étudier les notions fondamentales d'agents, les différentes définitions, caractéristiques, différentes architectures et les principaux types d'agents. Ensuite nous verrons les SMA, leur émergence de l'IAD et leurs propriétés. Nous aborderons également les concepts d'agents au sein des sociétés, les nouvelles contraintes et fonctionnalités issues d'une coopération entre les différents agents constituant la société.

Le chapitre 3, ***Emergences et Métaheuristiques***, présente dans la première section les notions d'émergence, d'auto-organisation et les définitions et fondements adéquats. Les méthodes métaheuristiques qui utilisent l'émergence dans son fonctionnement, comme par exemple, le calcul évolutionnaire, les réseaux neuronaux, les algorithmes de colonie de fourmis et l'optimisation par essaim particulaire sont alors présentées.

Le chapitre 4, ***Algorithmes du Système Immunitaire Artificiel***, est consacré à des notions indispensables des systèmes immunitaires naturels et les fondements théoriques sur lesquelles est basée la conduite de ces derniers. Nous présenterons les systèmes immunitaires artificiels qui sont l'application des systèmes immunitaires naturels pour la résolution de problèmes complexes. Les mécanismes efficaces du système immunitaire : la sélection clonale, la capacité d'apprentissage, la mémorisation, la robustesse et la flexibilité, rendre les systèmes immunitaires artificiels utile pour les problèmes d'ordonnancement.

Le chapitre 5 : ***Méthodes de résolution du problème d'ordonnancement de type flow shop***, est réservé à la présentation des notions de la maintenance préventive systématique, leurs caractéristiques et leurs politiques utilisées. Nous donnons quelques

méthodes de résolution du problème de l'intégration des politiques de maintenance systématique dans l'ordonnancement, pour faire des comparaisons avec nos solutions proposées.

Le chapitre 6 : *Approches développées pour l'intégration de politiques de maintenance systématique dans l'ordonnancement*, présente notre travail qu'est l'intégration des politiques de maintenance préventive systématique dans des ateliers de type flow shop hybrides afin de minimiser le makespan. Nous proposons deux approches Multi agent basée sur deux méthodes d'émergence, les algorithmes génétiques et les systèmes immunitaires artificiels.

Enfin, nous concluons ce travail de recherche en présentant les conclusions auxquelles nous avons abouties, ainsi que les perspectives de notre travail.

1.4 Contributions

De nombreuses hypothèses réalistes ont été incorporées dans les problèmes d'ordonnancement. Par exemple, en raison de l'interaction entre les activités de production et de maintenance, de nombreux chercheurs ont étudiés comment planifier simultanément les deux activités [134-140].

Plusieurs heuristiques ont été proposées pour résoudre le problème d'ordonnancement de type flow shop [134-140], y compris *PCDT*, *PLDT*, *règle Johnson ($m/2, m/2$)* et celle de Nawaz, Ensore et Ham (NEH).

Chacune de ces heuristiques présentent de nombreux inconvénient. Par exemple, les méthodes proposées sont souvent complexes et nécessitent des tâches difficiles de codage pour les mettre en œuvre. En plus, elles ne sont praticables que pour les petites instances (jusqu'à 10-15 Jobs) et même dans ce cas, le temps de calcul est élevé.

Les principales contributions de notre thèse sont regroupées en trois parties. Une première partie relative à une recherche bibliographique sur les SMA et les méthodes d'émergences, notre domaine de recherche.

La deuxième partie de notre travail, consiste à proposer deux approches Multi agent basées sur deux méthodes d'émergence, les AG et les SIA, pour l'intégration des politiques de maintenance préventive systématique dans les ateliers flow shop hybrides afin de minimiser le makespan. Les deux approches proposées sont appliquées sur trois

politiques de maintenance préventives. Ces politiques sont destinées à maximiser la disponibilité ou à maintenir un niveau minimum de fiabilité au cours de la production.

La troisième partie de notre contribution consiste à proposer plusieurs variantes d'heuristiques, y compris *PCDT*, *PLDT*, *règle Johnson ($m / 2, m / 2$)* et celle de Nawaz, Enscore et Ham (NEH). L'objectif est de faire des comparaisons entre ses heuristiques et nos deux approches proposées qui sont basées sur les SMA et les méthodes d'émergences. En plus, nous avons extraire plusieurs résultats basées sur ces comparaisons.

L'originalité de cette thèse réside dans l'utilisation des SMA avec les AG et les SIA dans l'intégration des politiques de maintenance préventive systématique dans un ordonnancement de type flow shop hybride.

Chapitre1: Contexte et Problématique

Chapitre1: Contexte et Problématique

Résumé

Nous présentons dans ce chapitre un état de l'art sur notre domaine de recherche, à savoir les SMA et les méthodes d'émergence.

Nous commençons d'abord par la présentation de quelques études réalisées avec les SMA dans le domaine industriel.

Nous citons ensuite des travaux de recherche réalisée avec les méthodes d'émergences dans le domaine industriel.

Nous nous intéressons particulièrement aux travaux réalisés dans le domaine de la maintenance systématique, et plus particulièrement, dans l'intégration de leurs politiques dans un ordonnancement de type flow shop hybride.

1.1 Introduction

Dans le domaine de la production industrielle, les tendances actuelles indiquent que les systèmes manufacturiers performants doivent s'adapter rapidement aux fluctuations du marché (demandes aléatoires) et aux perturbations internes (pannes des machines). Les machines doivent pouvoir fabriquer plusieurs types de produits simultanément et de manière efficace [62].

Dans un tel contexte, rester toujours performant, passe obligatoirement par le maintien en état de fonctionnement de l'outil de production, qui reste toujours la préoccupation majeure de tous les gestionnaires dans un monde industriel où les notions de réactivité, de coûts et de qualité ont de plus en plus d'importance, et où il est vital de pouvoir s'appuyer sur un système de production performant à tout instant [60].

Dans ce chapitre nous présentons une recherche bibliographique sur ce domaine, afin de préciser notre problématique à travers les travaux réalisés dans le même contexte.

1.2 Les SMA dans le domaine industriel

Touaf [40] a proposé une architecture basée sur le paradigme des SMA cette architecture a été conçue dans le cadre d'un projet européen intitulé MAGIC (Multi-Agent based Diagnostic Data Acquisition and Management in Complex Systems). Le système MAGIC offre une aide au diagnostic lorsqu'un défaut s'est produit. Il permet aussi de détecter des défauts en voie de développement dans les systèmes complexes.

La figure 1.1 présente l'architecture générale du système MAGIC.

Coudert et al [46] proposent la modélisation de la fonction d'ordonnancement de la production et celle d'ordonnancement de la maintenance par deux SMA coopérants. Ils ont utilisé la logique floue pour modéliser les degrés de liberté temporels de la négociation.

Le processus de coopération entre un agent Machine et un agent Maintenance dans la modélisation de Couderet et al [46] est défini comme suit: avant de formuler une enchère sur une opération de fabrication, l'agent machine demande à son agent Maintenance si le positionnement envisagé provoque un conflit avec une activité de maintenance. S'il n'y a pas de conflit, l'agent Maintenance envoie son accord à l'agent Machine qui formule son enchère.

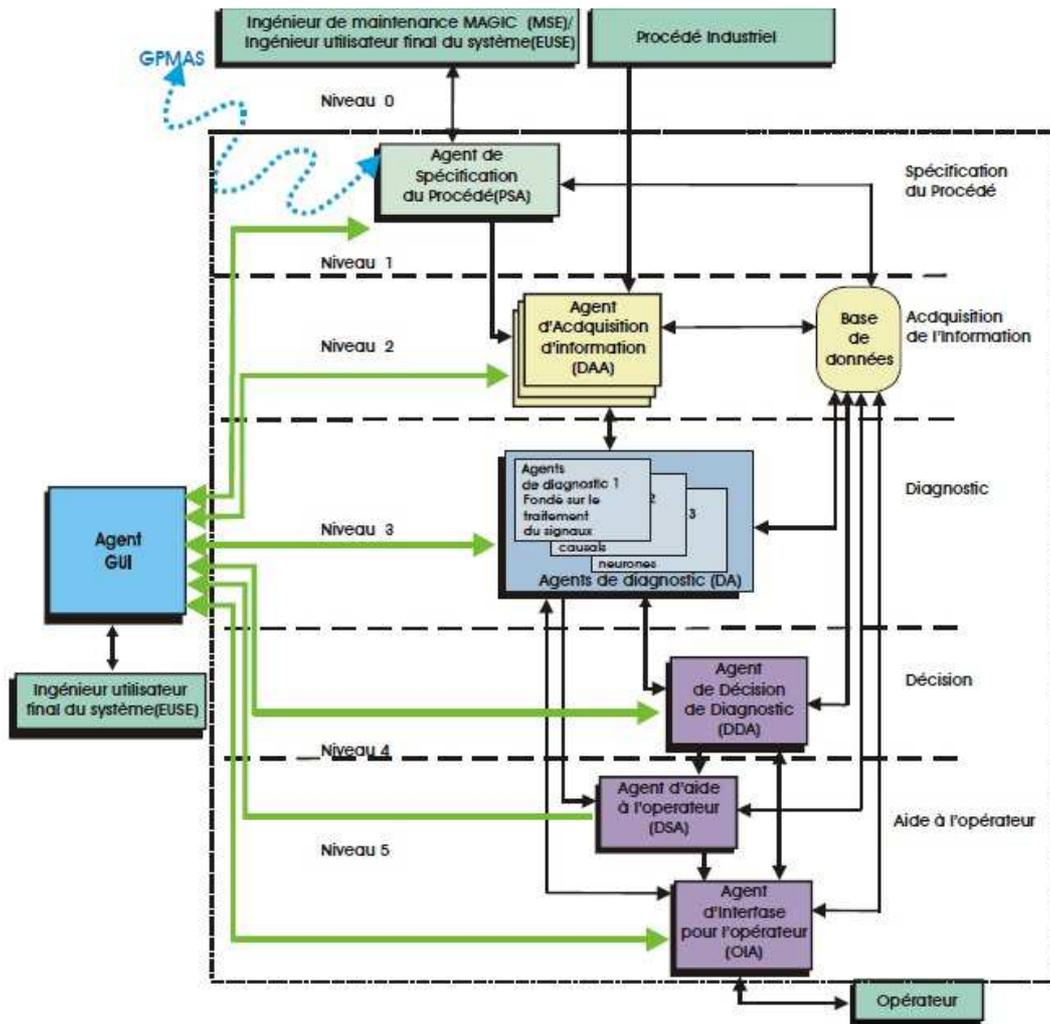


Figure 1.1 Architecture générale du système Magic

Si un conflit est détecté, l'agent Maintenance vérifie s'il est possible de déplacer le début de l'activité de maintenance à l'intérieur de sa fenêtre temporelle. Si le début de l'activité reste dans la fenêtre, l'agent Maintenance accepte ce changement et valide le positionnement de l'opération de fabrication. Dans le cas contraire, l'agent Maintenance demande à l'agent Machine s'il est possible de déplacer le début de l'opération de fabrication à l'intérieur de sa propre fenêtre temporelle, et suggère une date de fin compatible avec la fenêtre temporelle de l'activité de maintenance en conflit. Si cela est possible, l'enchère est modifiée ; sinon, il est nécessaire de trouver un compromis en relâchant la contrainte sur l'activité de maintenance ou sur l'opération de fabrication (Figure 1.2).

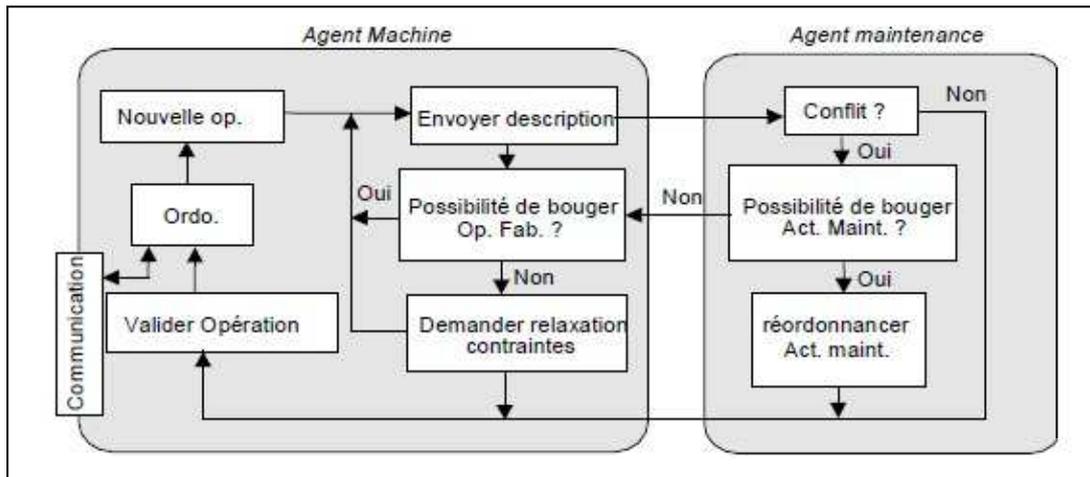


Figure 1.2 Coopération/négociation entre agents Machine et Maintenance

Deguet [41] envisage les liens entre l'émergence, SMA et résolution de problèmes à travers une triple restriction :

- Restriction de la résolution de problèmes à la recherche heuristique.
- Restriction de l'émergence à la caractérisation de vrais composites par les synergies.
- Restriction des SMA à un modèle multi-agent fondé sur des traitements fonctionnels et une organisation arborescente.

Gauthier [42] développe la méthode ADELFE pour la conception de systèmes ouverts, complexes et distribués, qui est basée sur la théorie des AMAS (Adaptive Multi-Agent Systems) et le concept d'émergence. Cette théorie donne un critère de conception local aux agents de sorte que l'émergence d'une organisation au sein du système et ainsi d'une nouvelle fonction soit rendue possible, rendant le passage à l'échelle possible en l'absence de contrôle global du système.

Gaber et al [43] proposent une auto-adaptative et auto-organisationnelle, basée sur les agents mobiles et inspirée du système immunitaire, pour la découverte de services dans un réseau dont l'évolution est aléatoire. Cette approche est basée sur l'organisation des serveurs en communautés par la création de réseaux d'affinités pour représenter les services dans le réseau. L'établissement des liens d'affinités entre les serveurs permet de résoudre, par collaboration, les requêtes des utilisateurs.

F. Ingrand [44] développe un système d'IA temps réel pour supporter les directeurs de flux. Il a été développé en utilisant le système procédural de raisonnement

(SPR) [44].OASIS a été conçu en subdivisant la tâche de gestion de trafic aérien en ses parties majeurs et en concevant des agents séparés pour résoudre chacun de ces sous problèmes. Chaque agent résout sa partie de la tâche indépendamment et en coopérant avec les autres pour pouvoir produire le comportement général du système.

Il y a 5 agents globaux (voir la figure 1.3):

- COORDINATOR : sert comme un gestionnaire de tâche, il coordonne les activités des autres agents globaux et les agents avion.
- SEQUENCER : utilise les techniques de recherche pour arranger l'avion dans la plus petite séquence délais/coût.
- TRAJECTORY CHECKER : vérifie que les instructions proposées par le système ne causent pas une violation des règles de sécurité de séparation.
- WIND MODEL : utilise les observations individuelles présent par les agents avion pour prévoir le champ de vent que l'avion peut rencontrer.
- USER INTERFACE : sert comme le seul point de communication avec le directeur de flux et gère tous les interactions utilisateurs.

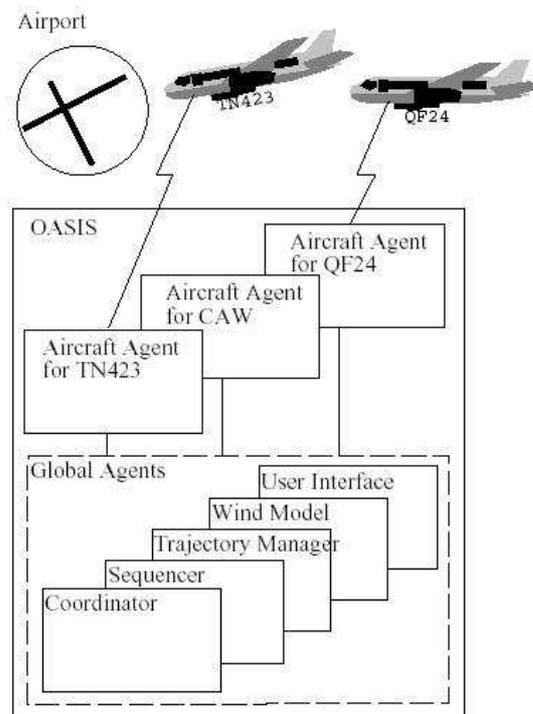


Figure 1.3 Architecture générale du système OASIS

Le système OASIS est implémenté dans un environnement de simulation de trafic aérien sur une station UNIX. Le système temps réel SPR est implémenté sur LISP mais il sera remplacé par un système basé sur C++ (MARS).

Le développement de OASIS a duré 2 ans et demi, il peut gérer maintenant des exemples real de trafic d'heure de pointe de l'aéroport de Sydney, incluant 65 arrivés pendant une période de 3 heures et demi.

1.3 Quelques études faites sur les méthodes d'émergence dans domaine industriel

Benbouzid [60] propose l'adaptation d'un certain nombre d'heuristiques des approches constructive, itérative et évolutive pour le cas de l'ordonnancement conjoint production/maintenance dans un atelier de type flow shop de permutation. D'autre part, elle étudié la contribution de la maintenance à la robustesse de ces ordonnancements conjoints. Le modèle proposé a pour objectif d'inscrire la génération d'ordonnancements conjoints comme une démarche proactive, et de démontrer que l'intégration de la maintenance lors de la génération des ordonnancements de production est à mettre au profit de la robustesse de ces derniers.

Harrath [47] s'intéresse à l'ordonnancement d'un atelier de type job shop. L'objectif de son travail concerne l'élaboration d'une méthode de résolution aussi bien dans le cas classique d'un ordonnancement relatif à la production que dans le cas beaucoup moins étudié touchant l'ordonnancement conjoint de la production et de la maintenance. Sa contribution comporte deux volets. Le premier volet prend appui sur les solutions générées par un algorithme génétique qui sont étudiées par des méthodes d'apprentissage. Dans un souci de validation et de comparaison par rapport aux travaux faits dans la communauté, la démarche proposée a été élaborée sur un problème classique de type $J // C_{max}$ et sur des benchmarks connus. Le deuxième volet propose un algorithme génétique Pareto optimal résolvant le problème d'ordonnancement conjoint de la production et de la maintenance au sein du job shop.

Le contexte de travail de *Kaabi* [62] concerne l'ordonnancement des activités de maintenance préventive systématique dans les systèmes de production.

La contribution de Kaabi comporte 3 volets :

- Le premier volet prend appui sur une règle de priorité proposée par Chu dans le cas classique d'ordonnancement sur une machine.

- Le deuxième volet propose un algorithme par séparation et évaluation permettant de générer des ordonnancements de permutation du problème conjoint de la production et de la maintenance au sein du flow shop a deux machines;
- Le troisième volet étend l'étude au cas du flow shop a plusieurs machines. Elle a proposé dans ce cas un algorithme génétique avec un codage approprié des solutions génétiques.

1.4 Les applications des systèmes immunitaires artificiels

Les applications fondées sur les *SIA* sont d'une grande variété de domaines. Cela marche de l'apprentissage, jusqu'à la robotique en passant par la détection d'anomalie et la résolution de problèmes [106].

Dans l'article [106] les auteurs mentionnent certains exemples d'applications servant les *SIA* :

1998	Un prototype de réseau immunitaire artificiel disposant pour résultat de rassembler des données et de clarifier les données répétées [102].
2002	La sélection négative est utilisée pour créer un groupe de détecteurs pour découvrir exclusivement ce qui est nécessaire. Cette manière aperçoit dans la reconnaissance de formes tout comme dans la segmentation d'images aériennes.
2002	L'algorithme de la sélection clonale est encore appliqué pour repérer des changements dans des données de série chronologique [88]. L'identification d'un ensemble de caractères binaires, en servant l'algorithme de la sélection clonale [104].
2003	On recouvre aussi la façon de la sélection négative dans des applications de surveillance de pannes. Ceci, dans l'objectif de créer des systèmes matériels donnant un maximum degré de fonctionnement, même en cas d'erreur [78].
2004	Dans les applications de la détection d'anomalie, se sont les Algorithmes de la sélection négative et de la sélection clonale qui sont beaucoup plus réels. Dans ce domaine, plusieurs études de recherches sont menées vers la sécurité des ordinateurs et des réseaux informatiques. L'algorithme de la sélection négative est employé pour détecter un germe (non soi) ou un client non inscrit. Celui de la sélection clonale progresse cette détection (en faveur à sa tâche d'hyper

	mutation somatique) et accepte une initiation des interventions (cellules mémoire) [107].
2005	L'algorithme du réseau immunitaire est employé pour constituer un système de reconnaissance et de classification de formes [85].

1.5 Etat de l'art sur le problème d'ordonnancement Flow shop

De nombreuses hypothèses réalistes ont été incorporées dans les problèmes d'ordonnancement. Par exemple, en raison de l'interaction entre les activités de production et de maintenance, de nombreux chercheurs ont étudié conjointement à planifier les deux activités.

Adiri [138] prouve que le problème de Flow Shop avec une période d'indisponibilité sur une seule machine est NP-difficile.

Kubiak [134] explore l'ordonnancement d'un Flow Shop avec des contraintes de disponibilité (*OFSCD*). Il considère deux variantes de l'*OFSCD* non-préemptif. Dans la première variante, les dates de démarrage des activités de maintenance sont fixes tandis que dans le second les dates de démarrage des activités de maintenance sont censées faire preuve de souplesse. Un algorithme basé sur l'algorithme génétique et la recherche tabou a été appliqué pour résoudre le problème.

Cheng et Wang [135] explore la non-préemptif de deux étages d'ordonnancement Flow Shop avec une seule machine sur le premier étage et m machines sur le deuxième étage sous la minimisation de la durée d'exécution. Ils supposent que chaque machine a eu plus une période d'indisponibilité et ces périodes sont connues à l'avance. Ils enquêtent également sur la performance du pire cas de trois autres heuristiques.

Reeves [139] envisage d'intégrer l'ordonnancement d'une seule machine et la planification de la maintenance préventive. Ils développent un algorithme génétique pour résoudre le modèle intégré mis en place par Kutanoglu [140].

Blazewicz et al. [141] enquête sur la Flow Shop de deux machines par un nombre quelconque de périodes d'indisponibilité sur une seule machine et prouve que le problème de minimisation du makespan est fortement NP-difficile.

Breit [142] traite le problème de l'ordonnancement de n job préemptif dans un Flow Shop de deux machines sur laquelle la première machine n'est pas disponible pour le traitement pendant un intervalle de temps donné.

Allahverdi [144] étudie le problème des machines stochastiques avec les pannes et le temps de Réglages séparés.

Yang et al. [146] envisage une Flow Shop de deux machines où les activités de la maintenance doivent être faites après avoir obtenu un nombre fixe de job. Les durées de ces activités de maintenance sont constantes.

Schmidt [147] enquête sur les méthodes existantes pour résoudre les problèmes d'ordonnancement sous contraintes de disponibilité ainsi que les résultats de la complexité.

1.6 Critiques et limites des travaux existants

La majorité des travaux touchant les relations Production/Maintenance utilisent des approches probabilistes dans l'objectif de définir le meilleur instant pour organiser une action de maintenance en fonction d'un arrangement entre le coût de maintenance et le danger d'altération de disponibilité des machines (voir par exemple [48], [49]). Dans le domaine industriel, les intervalles entre les fonctions de maintenance sont proposés par les fournisseurs de machines, ou déterminés à partir de contraintes techniques reliées au processus. Dans les logiciels d'ordonnancement industriels, on indique généralement ces fonctions comme des intervalles de fermeture : les niveaux de liberté pouvant existée sur leur positionnement exact sont donc oubliés.

Certains travaux de recherches ont été faites sur les systèmes immunitaires dans l'objectif de comprendre les mécanismes essentiels de ces derniers et essayer de fournir des processus et des procédés pour le développement de nouvelles résultats dans divers domaines.

La minimisation du C_{max} dans le cas du flow-shop général est un problème NP-difficile au sens fort (voir [134][138][141][144]). Plusieurs heuristiques ont été proposées pour le résoudre, y compris *PCDT*, *PLDT*, règle *Johnson* ($m / 2$, $m / 2$) et celle de Nawaz, Ensore et Ham (NEH). Chacune de ces heuristique donne de bons résultats, mais aucune ne garantie la solution optimale.

Bien qu'un grand nombre de méthodes, y compris la programmation mathématique, des critères différents et des méthodes heuristiques ont été présentés à intégrer l'ordonnancement de la production et de la maintenance, ils présentent de nombreux inconvénients.

Par exemple, les méthodes proposées sont souvent tout à fait complexes et nécessitent des tâches ardues de codage pour mettre en œuvre.

Les Méthodes exactes ne sont pas praticables que pour les petites instances (jusqu'à 10-15 Jobs) et même dans ce cas, le temps de calcul a tendance à être très élevé.

1.7 Synthèse

Les méthodes par Intelligence Collective montrent des exemples types pour la résolution de problèmes par émergence. Elles ont pourtant des spécificités distinctives qui nous semblent limitatives. En effet, chacune est très proche de son aspect biologique et en saisit donc les aspects distinctifs, typiques à chacun des prototypes. Il est alors généralement difficile de les reprendre pour des problèmes montrant des topologies ou contraintes divers.

En ce qui touche l'auto-organisation dans les *SMA*, il suffit en fait de formuler les outils dirigeant les conduites des agents en accord avec les spécificités distinctives du principe d'auto-organisation.

Réunir la résolution de problèmes par des *SMA* fondés sur des dispositifs d'auto-organisation indique alors parcourir la variété de mécanismes faisables dans un cadre commun les soutenant.

Il existe de très nombreuses méthodes de résolution de problèmes que l'on recouvre généralement sous l'appellation de méthodes d'optimisation. La technique qu'on a choisi est les algorithmes des systèmes immunitaires artificiels, qu'on va expliquer en détail dans le chapitre 4.

Le choix de cette technique est basé sur le fait que peu de travaux réalisés avec cette méthode dans le domaine industriel, et plus particulièrement dans l'intégration des politiques de maintenance systématique dans le l'ordonnancement.

Les mécanismes efficaces du système immunitaire : la sélection clonale, la capacité d'apprentissage, la mémorisation, la robustesse et la flexibilité, rendre les systèmes immunitaires artificiels utile pour les problèmes d'ordonnancement.

En plus nous pensons que l'hybridation de cette technique avec les SMA peut donner des résultats très satisfaisants.

La présentation des SMA fera l'objet du chapitre 2.

Chapitre2: Systèmes Multi-Agents

Chapitre2: Systèmes Multi-Agents

Résumé

Les systèmes multi-agents forment un paradigme pour la conception des systèmes complexes et proposent des outils pour les analyser, les concevoir et les implanter. L'approche agent considère les systèmes comme des sociétés composées d'entités autonomes et indépendantes, appelées agents, qui interagissent en vue de résoudre un problème ou de réaliser collectivement une tâche.

Cette approche semble intéressante dans le cadre des métaheuristiques, tout d'abord pour aborder la distribution et l'adaptation, mais aussi pour apporter des outils conceptuels et méthodologiques permettant de les analyser et de les modéliser.

Nous introduisons dans ce chapitre les définitions principales d'agents, caractéristiques, différentes architectures et les principaux types d'agents. Nous traitons aussi, les SMA, leur émergence de l'IAD et leurs propriétés.

2.1 Introduction

Nous présentons dans ce chapitre les notions fondamentales d'agents, les différentes définitions, caractéristiques, différentes architectures et les principaux types d'agents. Nous verrons ensuite les *SMA*, leur émergence de l'IAD et leurs domaines d'applications. Nous aborderons également les concepts d'agents au sein des sociétés, les nouvelles contraintes et les nouvelles fonctionnalités issues d'une coopération entre les différents agents constituant la société.

2.2 Agents

Les agents sont les éléments principaux des *SMA*. Chacun d'entre eux est une entité autonome chargée de réaliser un travail bien précis.

2.2.1 Définitions d'un agent

Ils existent plusieurs définitions d'agents, qui diffèrent selon le domaine d'utilisation et les spécificités des systèmes.

La notion d'agent est utilisée d'une façon assez floue, toutefois quelques définitions claires existent dont voici quelques-unes :

- Un agent est une entité physique ou virtuelle [24].
 - capable d'agir dans un environnement,
 - pouvant communiquer directement avec d'autres agents,
 - régie par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
 - possédant des ressources propres,
 - capable de percevoir (mais de manière limitée) son environnement
 - ne disposant que d'une représentation partielle de cet environnement (et éventuellement aucune)
 - possédant des compétences et offrant des services
 - pouvant éventuellement se reproduire
 - dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont il dispose, et en fonction de sa perception, de ses représentations.

- Une entité devient un agent aussitôt qu'elle est capable d'exercer un contrôle local sur ses processus de perception, de communication, d'acquisition de connaissances, de raisonnement, de prise de décision ou d'exécution [22].
- Un agent est un système informatique se trouve dans un environnement qu'il peut percevoir et sur lequel il peut agir de manière autonome.

Après cette comparaison, nous pouvons conclure que la dernière définition est la plus convenable pour notre domaine de recherche.

La figure 2.1 présente la position d'un agent par rapport à son environnement.

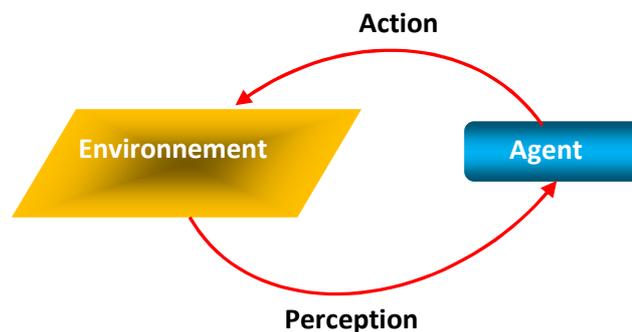


Figure 2.1 : *Agent dans un environnement.*

2.2.2 Architecture d'un agent

La structure d'un agent est parfois complexe. C'est cette structure qui lui permet de réaliser les objectifs pour lesquels il a été conçu, de réagir en fonction des événements et d'interagir avec les composants de son système.

Dans une architecture d'agents, on perçoit les paramètres ordinaires tels que : le type d'approche, structures de subordination...etc. Plusieurs architectures existent telles que, les architectures connexionnistes de tâches compétitives et l'architecture à base de modules horizontaux [24]. L'Architecture modulaire horizontale est sans doute la plus répandue, que ce soit pour des travaux théoriques ou des applications pratiques.

L'architecture principale pour développer les agents cognitifs est conçue sur la notion d'ensembles de modules horizontaux attachés entre eux. Les architectures modulaires sont conçues comme un assemblage de modules accomplissant chacun une fonction horizontale particulière. Les modules les plus courants sont :

- ✓ Les activités perceptives.
- ✓ L'envoi et l'explication des communications.

- ✓ La base de croyances interprétant la spécification de l'environnement et des autres agents.
- ✓ La gestion des contrats...etc.

Nous apercevons nettement sur la figure 2.2 les connexions existantes entre les différents modules de l'architecture de cet agent.

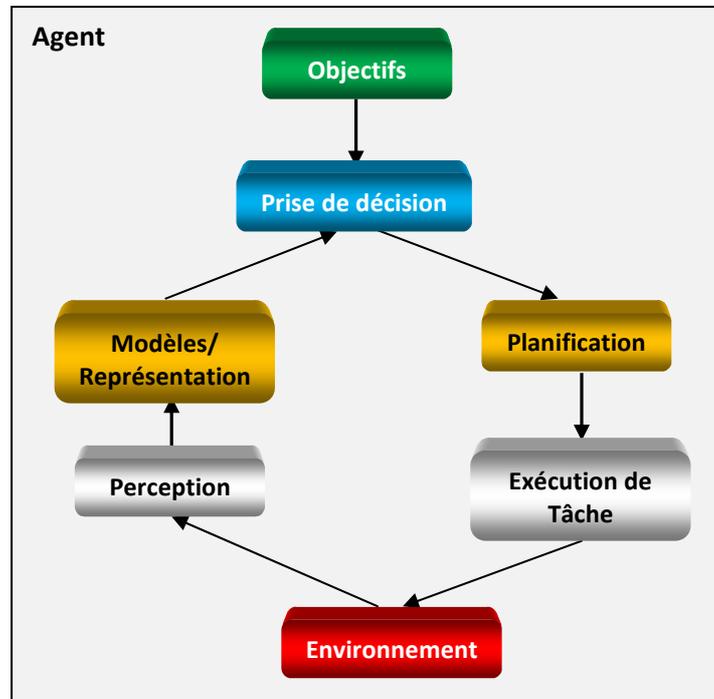


Figure 2.2 : Représentation d'un agent.

2.2.3 Propriétés des agents

Nous percevons parmi les propriétés des agents, quelques caractéristiques de base telles que la communication ainsi que d'autres caractéristiques bien particulières aux agents, telles que la sociabilité, la réactivité ...etc.

Autonomie	L'agent doit être autonome, il doit agir seul et uniquement en fonction des signaux qu'il acquiert des autres agents ou de l'environnement. Il dirige son état interne en fonction de ces données [22].
Apprentissage	Les agents ont l'aptitude d'apprendre tout au long de leur vie. Cet apprentissage leur permet de progresser et même de modifier leurs comportements [25].
Communication	La communication est une propriété essentielle des agents. Elle est identiquement la base de la collaboration entre agents. Un agent

	peut requérir un service auprès d'un autre agent en lui envoyant des messages [26]. Cette propriété est la base du comportement social des agents. Néanmoins, des obstacles majeurs existent en notions de standards et langage de communication ...etc.
Pro-activité	Un agent peut identiquement répondre en fonction de ce qu'il doit accomplir, ce type d'agents est composé d'une certaine intelligence et connu sous le nom d'agents cognitifs et sera précis par la suite.
Sociabilité	Fréquemment les agents sont structurés en groupes. Un groupe exige des relations entre ses membres. De ce fait, les agents sont conduits à négocier, collaborer pour la résolution de problèmes. La sociabilité est fondée sur des interactions entre agents, qui se font grâce à des standards de communication qui seront précis par la suite [27].
Activité	L'agent travaille dans un processus à part est régulièrement actif, et il n'a pas besoin d'une assistance humaine pour l'exécuter [28].
Réactivité	Cette propriété se résume au fait que l'agent répond en fonction de ce qui se passe dans son environnement. L'agent dispose des capteurs lui permettant de recueillir des informations de l'environnement ou des autres agents, et en fonction de ces informations il peut agir via des actionneurs.

2.2.4 Classes d'agents

Selon de ce que perçoit l'agent comme situations, il peut répondre. S'il s'agit d'un cas familier pour l'agent, pour laquelle il connaît 'parfaitement' l'action à effectuer il s'agit d'un agent réactif. En revanche, si ce cas est familier pour l'agent mais il lui faut un certain 'raisonnement' pour le résoudre, il s'agit dans ce cas d'un agent cognitif [29]. Il existe deux catégories d'agent :

2.2.4.1 Agents réactifs

Cette classe d'agents se décrit par le fait qu'ils n'ont pas de description de leur environnement, ni du monde auquel ils dépendent. Ces agents sont les plus simples à mettre en œuvre du fait qu'ils s'interprètent selon le stimulus. L'agent sera accommodé sous forme de couples « Stimulus/Réponse ».

Souvent les agents réactifs sont vus comme non ou peu intelligents, et nous observons que l'intelligence révélée de la collaboration des différents agents.

Ces derniers sont de plus bas degrés et n'ont qu'une norme et un langage de correspondance réduite.

La figure 2.3 donne une présentation générale d'un agent réactif.

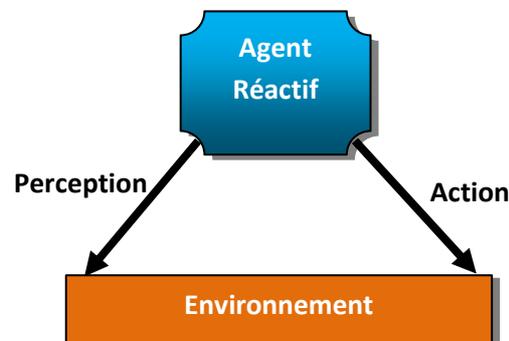


Figure 2.3 : *Agent réactif*

Selon les tuteurs de cette approche, il n'est pas obligatoire que les agents soient intelligents pour que le système ait un comportement total intelligent.

Les exemples préférables de ce genre de systèmes composés d'entité de faible intelligence mais dont la collaboration conduit à des systèmes très intelligents, sont les colonies de Fourmis [28].

2.2.4.2 Agents cognitifs

Au contraire des agents réactifs, les agents cognitifs sont habituellement dotés d'une certaine intelligence et une bonne représentation de l'environnement et du monde auquel ils appartiennent.

Les groupes d'agents cognitifs sont couramment composés d'un petit nombre d'agents. Ces agents sont analogues à des systèmes experts et sont des agents communicants [30].

Ces agents ont un diagramme très clair de leur environnement. Chaque agent a des perceptions comprenant des informations et du savoir-faire. Les agents cognitifs sont dit conscients car ils ont des buts à réaliser, avec une véritable IA et une aptitude d'apprentissage et d'adaptation.

La figure 2.4 présente la structure générale d'un agent cognitif.

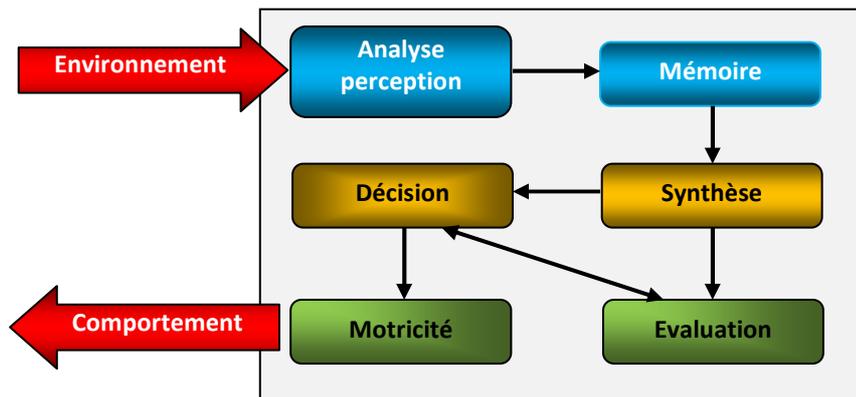


Figure 2.4 : Agent cognitif

Les agents cognitifs sont intéressants indépendamment et simultanément pour leur intelligence. D'autre part le rapport des agents réactifs est exclusivement l'interaction avec les autres agents, et l'accomplissement de tâches claires et simples [22].

2.2.5 Normes de communication entre les agents

Vu le nombre élevé d'interdépendances entre les agents, et vue la divergence fonctionnelle des agents, une norme de communication s'exige afin de normaliser les messages et les signaux entre agents. Il ya eu certaines propositions pour les langages de communication [21]:

<p>Knowledge Interchange Format(KIF)</p>	<p>Basé principalement sur la logique du premier ordre, ce standard autorise de formuler les caractéristiques d'un domaine spécifique. KIFa une notation analogue au langage de programmation LISP avec certains opérateurs et des quantifieurs.</p>
<p>Knowledge Query Meta Language (KQML)</p>	<p>Plus général, KQML admet de développer le format général d'un message. En termes de programmation, un</p>

	message KQML en un objet recueilli de la classe message et a des attributs.
Foundation for Intelligent Physical Agents (FIPA)	Extension de KQML, un standard de langage de communication ACL (Agent Communication Language), a été créée par la FIPA en 1995. ACL est composé de plusieurs unités de performatifs, et plus d'attributs

2.2.6 Agents et Objets

Les deux grands axes de l'informatique actuel sont sans éventualité la programmation orientée objets (POO) et la programmation orientée agents (POA). Les différences essentielles entre les deux accès [21]:

- ✓ Un agent est un objet qui est inductible, rationnel, autonome, capable de communiquer et d'agir.
- ✓ Un agent logiciel est de préférence conçu sous forme d'un objet ayant les propriétés d'un processus.
- ✓ Un objet O1 peut appeler une méthode existante d'un objet O2 qui doit le réaliser.
- ✓ Un agent A1 peut demander l'exécution d'une méthode à l'agent A2 qui est libre de l'exécuter ou pas.
- ✓ Un agent travaille en fonction de son objectif, de ses contraintes et de ses aptitudes.

Ces points effectuent la différence entre agents et objets, et c'est sur ces bases que se fait le choix de l'approche à utiliser selon la problématique.

2.3 Systèmes multi-agents

2.3.1 Historique des systèmes multi agents

Les SMA sont la réunion de divers domaines de recherches :

- L'IA pour l'aspect prise de décision de l'agent.
- Les systèmes partagés et l'IA partagée pour les interactions et la remise de l'exécution.
- Le génie logiciel pour le changement vers des éléments logiciels de plus en plus libres et proactifs.

L'une des grandes sources d'inspiration pour les *SMA* est le comportement des colonies animales et de quelques groupes d'insectes tels que les fourmis, les abeilles et les termites. L'inspiration est identiquement originaire d'autres domaines connexes surtout la sociologie, la psychologie sociale, les sciences cognitives et bien d'autres [31].

Les recherches dans la discipline des *SMA* poursuivent sur deux buts importants. Le premier intéresse l'analyse théorique et empirique des dispositifs qui ont lieu lorsque certaines entités autonomes interagissent. Le second s'intéresse à l'accomplissement de programmes partagés apte de réaliser des fonctions difficiles via la collaboration et l'interaction. Leur situation est donc double : d'un côté, ils se mettent au sein des sciences cognitives, sociales et naturelles pour à la fois spécifier, développer et simuler des événements naturels, et provoquer des prototypes d'auto-organisation; de l'autre côté, ils se montrent comme une pratique et une méthode qui vise l'accomplissement de systèmes informatiques compliqués à partir des notions d'agent, de message, de collaboration et d'organisation de faits [33].

On peut dire que les *SMA* ont aperçu le jour avec l'apparition de l'IA. À ses origines, l'IAD ne tenait qu'à collaboration entre régisseurs de problèmes afin de participer à résoudre un but conjoint. Pour y arriver, on partageait habituellement un problème en sous-problèmes, et on fournissait ces sous-problèmes à divers solveurs qui sont désignés à modifier des données pour organiser des solutions partielles. Celles-ci sont finalement assemblées en une solution entière au problème de démarrage.

Aussi, l'IAD au démarrage avait une image « agent-centrée » qui favorisait le « problème à résoudre » tout en mettant l'accent sur la détermination d'un tel problème par de diverses entités intelligentes. En revanche, les *SMA*s maintenant ont plus une image « interaction-centrée » où les agents sont autonomes, habituellement hétérogènes, etc. Dans ce cas, l'accent est plutôt mis sur le fait de comprendre comment les agents vont s'arranger pour échanger leurs idées, objectifs et schémas pour accomplir et dénouer des problèmes.

2.3.2 Définition d'un système multi-agents

L'utilisation d'un agent indépendamment ne répond généralement pas aux espérances. En conséquence, des problèmes plus compliqués réclament l'utilisation de divers agents, chacun ayant un but, et ce n'est qu'en les faisant collaborer qu'on aboutit à des solutions plus efficaces. Une autre contrainte est le fait que des problèmes sont de

manière inévitable répartis (ex : gestion décentralisée d'un réseau électrique), et donc seule une solution distribuée peut résoudre ce type de problèmes [33].

Un SMA est comparable à un système réparti composé d'agents. On appelle SMA, un système constitue des éléments suivants [24]:

- Un environnement **E**, c'est-à-dire un intervalle accommodant fréquemment d'une métrique.
- Un groupe d'objets **O**. situés, c'est-à-dire que, pour tout objet, il est faisable, à un moment donné, d'assembler une position dans E. Ces objets sont inactifs, qui peuvent être vus, créés, détruits et changés par les agents.
- Un ensemble **A** d'agents, qui sont des objets spécifiques ($A \subseteq O$), lesquels indiquent les entités actives du système.
- Un ensemble de relations **R** qui réunissent des objets entre eux.
- Un ensemble d'opérations **Op** admettant aux agents de A de percevoir, constituer, consommer, modifier et traiter des objets de O.
- Des opérateurs chargés de symboliser l'application de ces opérations et la réplique de l'ensemble des constituants du SMA à cette tentative de modification, que l'on appellera les lois de l'univers multi agents.

La représentation d'un SMA est défini dans la figure 2.5.



Figure 2.5 : Représentation d'un système multi agents.

Cette définition des SMA a été proposée par Ferber en 1995, elle n'a pas été remise en cause à ce jour [24].

2.3.3 Types de systèmes multi agents

Il existe plusieurs types de SMA. Nous pouvons faire la différence entre les types de SMA selon les types d'agents les constituants ou bien selon les interactions entre leurs agents ...etc. Si nous considérons les SMA selon les interactions entre leurs agents, nous trouvons deux types de SMA :

Ouverts	Dans ce type de SMA, aucune barrière n'est exigée aux agents (relatif à leur déplacement). En effet, les agents peuvent entrer, ressortir, modifier d'environnement.
Fermés	Aucun échange n'est permis, les agents ne doivent interférer qu'au sein de leur environnement et doivent se rassurer de ce qui est présent dans ce dernier.

Si en revanche, nous examinons les SMA selon le type d'agents les composants.

Homogènes	Le SMA est constitué de plusieurs agents du même type.
Hétérogènes	Divers types d'agents peuvent se remarquer au sein du même SMA.

2.3.4 Propriétés des systèmes multi-agents

L'utilisation des SMA est parfois demandée. Quand on s'aperçoit face à des problèmes qui obligent un tel choix. Néanmoins, nous pouvons favoriser d'utiliser des SMA pour les raisons suivantes [36]:

<i>Les SMA reflètent la réalité</i>	la plupart des problèmes sont partagés qui s'assemble simplement aux SMA.
<i>Diversité,</i>	les SMA peuvent avoir parmi les agents les composants une grande différence, ce qui donne l'éventualité aux concepteurs d'incorporer divers agents (réactifs, cognitifs ...etc.)
<i>Coopération,</i>	Les systèmes entre eux peuvent collaborer pour la résolution de problèmes plus complexes.
<i>Modularité,</i>	Le grand nombre d'agents admet de diviser les problèmes en sous problèmes simples, cette approche est l'extension du découpage modulaire de la POO à la POA.

2.3.5 Plates-formes multi-agents

Certaines plates-formes permettant de développer et d'exécuter des SMA suivant aux normes de communications entre agents auparavant évoquées.

Les plates-formes multi-agents les plus connues sont :

Java Agent DEvelopment (Jade)	Jade permet le développement et l'exécution de SMA conforme aux normes FIPA. Cette plate-forme propose, un service de nommage, un service de pages jaunes, des dispositifs de transport de messages, un service d'analyse et une bibliothèque des protocoles d'interactions de FIPA.
Jack	Jack est exposé comme étant un environnement pour créer, réaliser et adopter des systèmes multi-agents commerciaux, écrits en Java et utilisant une approche orientée composants.
AgentBuilder	AgentBuilder est un ensemble d'outils incorporés permettant de

	créer des agents intelligents. Développée par <i>Reticular Systems Inc.</i> . Cet outil est important il assemble à la fois un logiciel de haute qualité et un modèle sous-jacent qui a fait ses preuves au niveau académique. [35].
Zeus	Zeus est un environnement incorporé pour la création rapide d'applications à base d'agents coopératifs. Il est développé par l' <i>Agent Research Program</i> du <i>British Telecom Intelligent System Research Laboratory</i> [34].
MultiAgent Development Kit (MadKit)	MadKit est une plate-forme multi-agents écrite en Java appuyée sur un modèle organisationnel. Elle est développée par <i>Olivier Gutknecht</i> et <i>Jacques Ferber</i> [34].

2.4 Conclusion

Les agents, une nouvelle représentation de programmation se déterminent par leur autonomie, activité, réactivité, pro-activité ...etc. Nous avons vu qu'il existe deux types d'agents : les agents cognitifs qui essaient de réaliser un objectif donné et les agents réactifs qui sont un ensemble de stimulus/réactions (agissent en fonction des événements). Quand les agents sont disposés dans des sociétés, ils sont dirigés à collaborer, communiquer et doivent faire face à des contraintes telles que la répartition des ressources.

Le choix de l'utilisation des systèmes multi-agents avec les méthodes d'émergences pour l'intégration des politiques de maintenance systématique dans l'ordonnancement des ateliers de production ; est justifié par certaines caractéristiques très intéressantes retrouvés parmi les nombreuses propriétés des SMA et les méthodes d'émergence et qui répondent aux nécessités et attentes de notre domaine de recherche.

Le troisième chapitre est consacré au domaine des méthodes d'émergence et leurs relations avec les métaheuristiques, nous présentons leurs notions de bases qui sont inspirés essentiellement des ensembles d'insectes tels que les colonies de fourmis, l'essaim particulaire et des systèmes nerveux humaines comme les réseaux de neurones.

Chapitre 3 :

Emergences et Métaheuristiques

Chapitre 3 : Emergences et Métaheuristiques

Résumé

L'émergence est un concept qui intervient lorsque des systèmes simples interagissent en nombre suffisant pour faire apparaître un certain niveau de complexité qu'il était difficile de prévoir par l'analyse de ces systèmes pris séparément [64].

L'objectif de ce chapitre est de présenter les notions d'émergence et d'auto-organisation. L'utilisation de l'émergence et de l'auto-organisation concerne un certain nombre de méthodes métaheuristiques, comme notamment, le calcul évolutionnaire, les réseaux neuronaux, les colonies de fourmis et l'optimisation par essaim particulaire.

3.1 Introduction

L'utilisation de l'émergence et de l'auto-organisation concerne un certain nombre de méthodes métaheuristiques, comme notamment, le calcul évolutionnaire, les réseaux neuronaux, les algorithmes de fourmis et l'optimisation par essaim particuliers. Ces techniques sont présentées dans la suite de ce chapitre.

Nous présentons dans la première section de ce chapitre les notions d'émergence et d'auto-organisation. Nous citons leurs définitions et fondements adéquats en mettant l'accent sur ce qui ressemble, mais surtout ce qui distingue les deux notions.

3.2 Emergence et Auto-Organisation

Généralement, dans plusieurs SMA et en général dans les systèmes adaptatifs compliqués, les notions d'émergence et d'auto-organisation sont utilisées simultanément [61].

L'émergence et l'auto-organisation, accentuent des spécificités distinctes du comportement d'un système. Les deux concepts peuvent trouver indépendamment, comme ils peuvent coexister dans un système dynamique.

Plusieurs travaux de recherche présentent l'émergence et l'auto-organisation, faussement, comme des synonymes. Cette confusion fausse le concept qu'on peut avoir quant à leurs sens respectifs.

L'objectif de cette partie est de traiter les spécificités importantes des deux notions. Une vision d'ensemble sur l'historique de l'emploi de chaque notion est donnée. Des définitions pratiques qui sont utilisées dans tout le document sont identiquement retenues.

3.3 Emergence

3.3.1 Historique

L'origine de l'émergence pourrait bien être l'axiome datant de la Grèce antique : « *le tout est plus que la somme de ses parties* ». On voit des marques de la notion d'émergence et d'auto-organisation dans des notes de Thalès (620 av. J.-C. – vers 540 av. J.-C.) et Anaximandre (615 av. J.-C. – vers 540 av. J.-C.). Dans le même ordre de notions, Aristote (380 av. J.-C., 320 av. J.-C.) discute du « *tout avant les parties* ».

Il se base sur ce concept pour montrer quelques idées qu'on n'arrive pas à modifier. On voit bien plus tard la même notion dans les rédigés de J.W. Von Goethe (1748 – 1831). Ce scientifique, ordonne la théorie de la Gestalt qui est une source à une configuration ou un prototype d'éléments si assemblés dans l'ensemble qu'on ne peut pas les énoncer comme une combinaison fondamentale de ses portions. Toutefois, le « tout avant les parties » identique que la Gestalt présument qu'une entité homogène est « pré-donnée» alors que l'émergence est une composition dynamique semblant au cours du temps [57].

L'émergence est par conséquence une notion pluri-définie. Elle est fréquemment montrer comme un événement où le contenu intégral d'un système dérive des dépendances entre les portions locales de ce même système. Dans ce qui suit, une explication plus approfondit de la notion d'émergence est exposée [52].

3.3.2 Définition d'émergence

La définition choisie de l'émergence est la suivante [50]: « Un système exhibe de l'émergence quand il y a des émergents cohérents au niveau macro qui apparaissent dynamiquement des interactions entre les parties au niveau micro. De tels émergents sont nouveaux par rapport aux parties du système prises individuellement ».

3.3.3 Caractéristiques de l'émergence

<p>relation micro-macro</p>	<p>La relation micro-macro [50][51][52][53] est la spécifique de l'émergence la plus sérieuse et la plus mentionnée dans les écrits. Il fait origine à des caractéristiques, des contenues, des compositions, ou des prototypes (des émergents) qui sont examinés au niveau macro et qui découlent des (dépendances au niveau micro du système).</p>
<p>Cohésion</p>	<p>La cohésion [52][53][54][55] fait source à un rapport logique entre les portions. Les « émergents » apparaissent comme les tous incorporés qui ont aptitude à soutenir une vraie perception d'identité dans le temps (c'est-à-dire un modèle soutenu). La cohésion perçoit et joint les éléments distincts du niveau bas en une unité au niveau haut. Les relations entre les éléments sont indispensables pour aboutir un tout homogène.</p>

Dépendance des parties	Sans les dépendances [52][53][54][56], les comportements macroscopiques attachants ne présentent jamais. En conséquence, les émergents dérivent des dépendances entre les parties.
Originalité	Le comportement intégral du système est inédit [55][56] par rapport aux comportements individuels au niveau microscopique. Il en découle que les unités au niveau microscopique n'ont pas une image exprimée du contenu entier. Dans les textes, il aperçoit certains énoncés pour cette idée : « le tout est plus grand que la somme de ses parties » [56], « ne peut pas être réduit à. . . » [57].
Contrôle décentralisé	Le contrôle décentralisé utilise uniquement des dispositifs locaux pour entreprendre le contenu total. Il n'y a aucun contrôle capital, c'est-à-dire aucune partie du système n'autorité le contenu macroscopique. Les faits des portions sont vérifiables alors que le tout n'est pas instantanément vérifiable.
Puissance et Souplesse	Les émergents sont partiellement distants aux troubles ou aux méprises. L'accroissement des troubles réduira l'exploit, mais l'abaissement ne sera pas coléreux : la valeur du résultat réduira constamment, sans altération rapide de fonction. Le revers ou le changement d'une entité ne parleront pas d'échec entier des émergents. Cette souplesse [56][57] ne fait que les distinctes entités peuvent être changées, cependant la composition émergente attendant.
Rapport bidirectionnel	Dans les systèmes émergents il y a un rapport bidirectionnel [53][58]. Du degré microscopique au degré macroscopique, les portions font présenter des émergents (voir relation micro-macro ci-dessus). Dans l'autre côté, les émergents agissant les portions.

3.3.4 L'émergence dans le domaine d'informatique

Pendant près de vingt ans, tout un domaine d'étude s'est peu à peu conçu autour de l'émergence pour profiter de ses spécificités dans les systèmes informatiques [53]

[54]. "*Emergent Computation*" [55] [56] est le mot utilisé habituellement pour indiquer ce champ dans lequel se placent nos recherches

Ce mot rassemble en fait des accès très différents car l'origine de l'émergence est certainement conceptuel et commun, et semble donc approprié quelle que soit la méthode utilisée, à partir de l'instant où celle-ci admet de s'installer dans le cadre de l'explication de l'émergence. Le mot d'*Emergent computation* n'a en conséquence décidément été que peu exploité, les groupes scientifiques favorisant se rassemblent par existentiel employée à une méthode claire et ses formes.

Malgré la concentration d'étude menée, l'émergence reste un fait peu habituel et on pourrait même dire un peu secret. Nous allons donc donner une explication "pratique" avec une coloration strictement informatique. Nous la corrompons en trois parties : ce que l'on veut faire émerger, à quelle modalité il y a émergence et comment nous nous en utilisons [59] [61]. Le tableau 3.1 donne une explication de l'émergence dans le domaine informatique.

Tableau 3.1 *L'émergence dans le domaine informatique.*

Objet	Un système informatique a pour but d'accomplir une activité appropriée à ce que l'on espère du système. C'est cette activité, pouvant changer au cours du temps, que nous voulons faire émerger.
Condition	Cette activité est émergente si la codification du système n'appartient aucunement de la compréhension de cette activité. Cette codification doit tenir des dispositifs admettant l'ajustement du système au cours de ses permutations avec l'entourage afin de déployer à tout moment vers l'activité appropriée.
Méthode	Pour modifier l'activité il faut modifier l'ordre des éléments du système. Ces mécanismes sont désignés par des règles dirigeant l'auto-organisation entre les éléments et ne dépendant pas de la compréhension de l'activité commune.

3.3.5 Axes de recherches scientifiques de l'émergence

Le phénomène d'émergence, tel qu'il est vu, calculé et utilisé de nos jours, tient ses racines dans plusieurs domaines de recherches (Figure 3.1). En commençant l'étude de systèmes simples, linéaires et en équilibre (Théorie générale des systèmes, la cybernétique, théorie de l'information) et en stimulant de phénomènes purs en physique et en biologie (Biologie Evolutive, l'état solide / Physique de la matière condensée) et de leurs théories collaboratrices (Théorie des Catastrophes, systèmes dynamiques, géométrie fractale).

La recherche s'est doucement tenue sur les usages de non linéarité, d'auto-organisation, de non stabilité, d'attracteurs (systèmes adaptatifs complexes, systèmes dynamiques non linéaires, théorie du chaos, thermodynamique d'équilibre, ...).

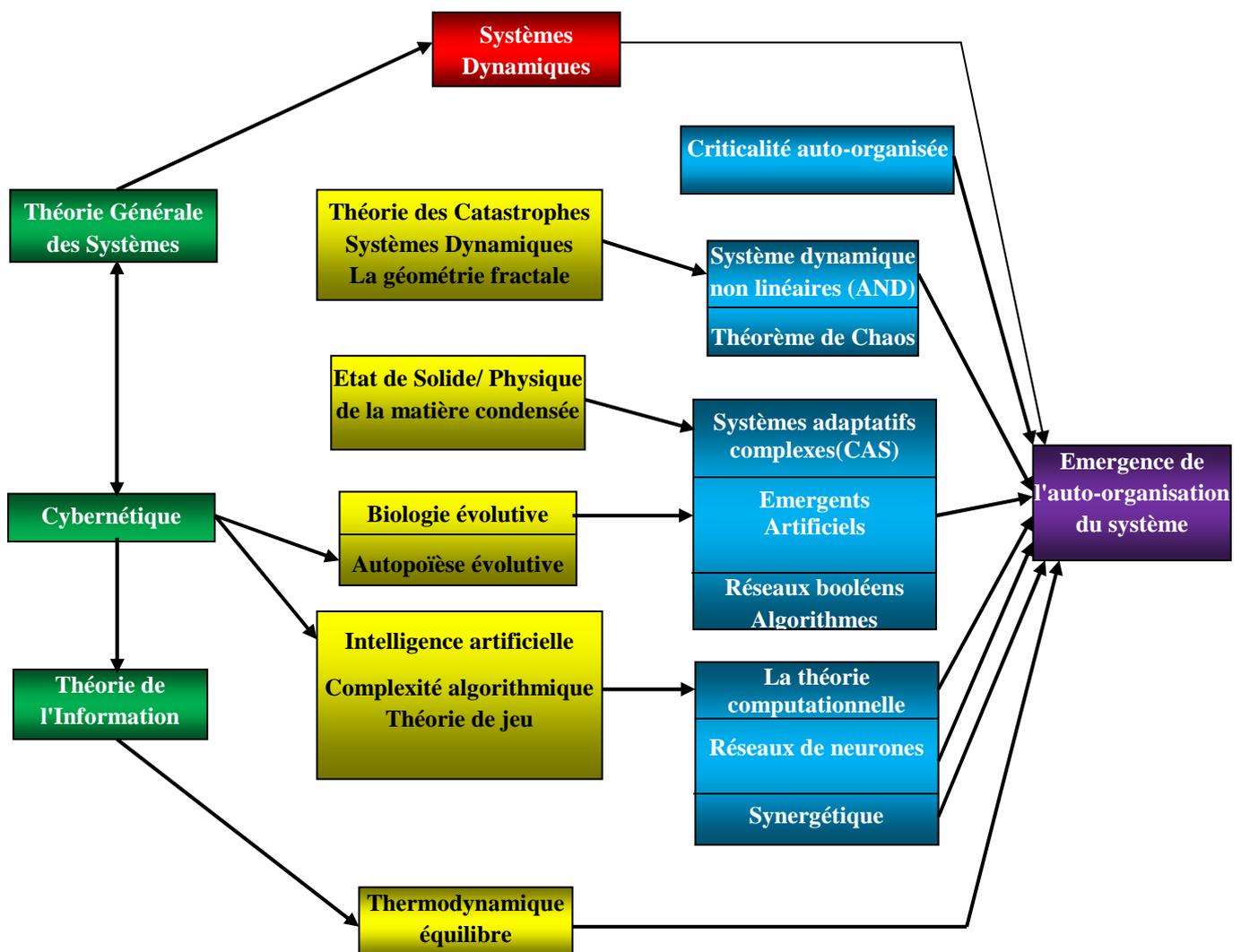


Figure 3.1 Axes de recherches scientifiques de l'émergence

L'émergence est en conséquence une solution pour montrer le besoin de progresser vers la macro niveau et ses mouvements propres, ses lois, ses caractéristiques, dans l'objectif d'éclairer de façon plus appropriée ce qui est examiné.

Les recherches sur l'émergence admettent de créer les dispositifs indispensables, les moyens et les constructions pour mieux expliquer le phénomène d'émergence et les systèmes dans lesquels ils s'aboutissent.

3.3.6 Exemples de phénomènes émergents

Pour bien expliquer l'émergence, on peut donner certains modèles de phénomènes très habituels dans plusieurs domaines et qui sont dits émergents au vu des caractéristiques exposées ci-dessus:

- En Physique et en Chimie, les dépendances entre des éléments simples donnent délivrance à des compositions compliquées affectées de domaines complètement nouvelles [58].
- En Biologie, plusieurs systèmes naturels construits d'unités libres montrent des dispositions à accomplir des fonctions capables de composés sans vérification entier. De plus, ils peuvent s'arranger à leur espace soit pour y demeurer, soit pour progresser le mouvement du commun. C'est bien nécessairement le cas des colonies d'insectes sociaux tels que les termites et les fourmis [63].
- Le système immunitaire est lui aussi typique du mouvement d'un système compliqué construit d'un groupe d'agents. En conséquence, le système immunitaire est une composition compliqué et réuni de cellules, de molécules et d'organes qui admettent de distinguer le soi du non-soi. Des dépendances au sein du système immunitaire ainsi que de celles avec les autres systèmes et organes, émerge le contrôle de l'organisme, assurant sa marche stable [63].

3.4 Auto-organisation

Pour Edgar Morin, l'auto-organisation c'est aisément et poétiquement:« le désordre organisateur » (figure 3.2).

C'est aussi ; une caractéristique d'un système remettant en compte son aptitude à changer et se modifier, créer et se créer, assembler et s'assembler, maintenir et se maintenir [61].

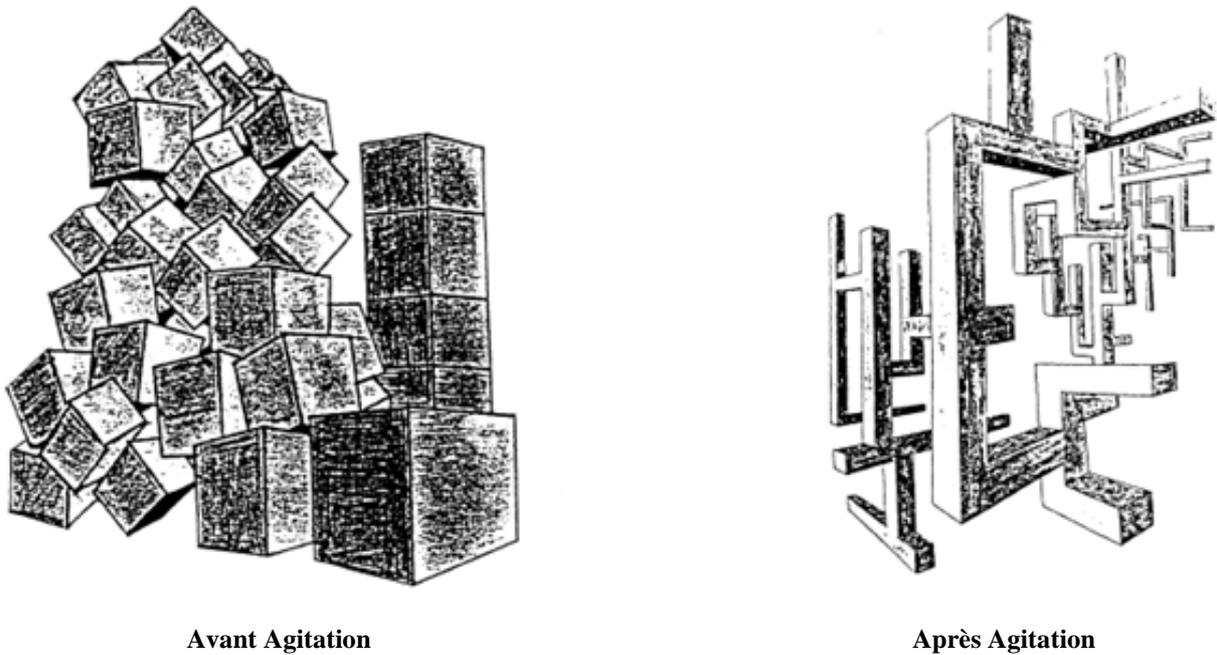


Figure 3.2 *Auto-organisation*

3.4.1 Historique et Définition

Si l'on retourne aux sources sorties de la physique et plus clairement de la Thermodynamique de l'auto-organisation, une explication peut-être attribuée : l'approche de compositions ou modèles (prototypes) sauf qu'un agent d'extérieur ne l'exige [59].

La cristallisation et les cellules de Bénard [63] sont parmi les modèles les plus courants. Mais l'idée est identiquement utilisée pendant longtemps par la Biologie [64] pour développer la divergence structurée des modèles dans le vivant comme par exemple les raies et autres prototypes sur les animaux ou les compositions géométriques de coquillages. [66] en fait surtout une exposition grande notable.

Une autre forme cruelle que l'on doit aux divers recherches de Francisco Varela, sur la Biologie et les cellules d'une part, mais surtout sur la conscience, est celui d'*autopoïèse* [Autopoietica]. Cette idée peut se synthétiser à l'aptitude d'un système, à travers la dépendance, le changement, la dévastation et la renaissance de ses principes, à conserver ses éléments et leur actif singulière, ainsi que l'unité clair du système.

3.4.2 Caractéristiques de l'auto-organisation

Sans rentrer dans les détails techniques, on peut invoquer quelques spécificités et notions inséparables ou communs comme :

<p>Augmentation de l'ordre</p>	<p>Le concept d'« organisation » [61][63][64] est l'une des spécifiés les plus importantes de la notion d'auto-organisation. L'organisation peut-être vu comme un accroissement de l'ordre dans la conduite d'un système qui accepte à ce dernier de fournir une composition universel, matérielle ou commode. Néanmoins, il faut annoncer que tout système présentant une mise en ordre graduelle ne peut être dit auto-organisé que s'il est autonome.</p>
<p>Indépendance</p>	<p>La deuxième spécifié essentielle de l'auto-organisation est l'absence de la vérification extérieur [63][64][65], d'où le nom « auto ». Un système doit s'arranger sans l'intervention de l'extérieur. Souvent, les entrées sont faisables tant qu'elles n'évoquent pas des instructions de l'extérieur du système.</p>
<p>Ajustement et Puissance face aux modifications</p>	<p>Dans les systèmes auto-organisés, le terme puissance [61][63][64] est utilisé dans l'effet de l'ajustement aux troubles et aux modifications. Un système auto-organisé doit faire face à ces modifications et garder son organisation de manière libre. Dans [65] les concepteurs signalent cette aptitude d'ajustement par le fait qu'une modification dans l'espace peut conduire le même système à créer des conséquences distinctes, sans aucune attraction sur les conduites des entités qui le composent.</p>
<p>Actif</p>	<p>Face à la nécessité de l'ajustement aux modifications de contexte, un système auto-organisé à l'exigence d'être actif [61][63][64]. Les modifications ou les troubles simulent la composition structurée. Il s'agit donc d'un mouvement fixe pour faire face à ces modifications.</p>

3.4.3 Combinaison entre l'émergence et l'auto-organisation

Les deux notions d'émergence et d'auto-organisation supportent des caractéristiques distinctes du contenu d'un système. Pour synthétiser, l'essence de l'émergence est l'existence d'un comportement entier récent par rapport aux éléments du système. Celle de l'auto-organisation est un comportement inductible qui procure et entretient d'une façon libre un ordre augmentant. Ces deux notions peuvent être indépendamment.

La figure 3.3 (a) présente un système auto-organisé. Il n'y a aucune vérification procédant de l'extérieur et aucune conséquence micro-macro. Au contraire, la figure 3.3 (b) montre un système totalement émergent. Toutefois, dans la majorité des systèmes ces deux notions voisinent (figure 3.3 (c)). Dans ce cas, ces systèmes sont nommés « systèmes auto-organisés émergents ». Dans les problèmes décentralisés et actifs, la réunion de l'émergence et de l'auto-organisation est même conseillée [55].

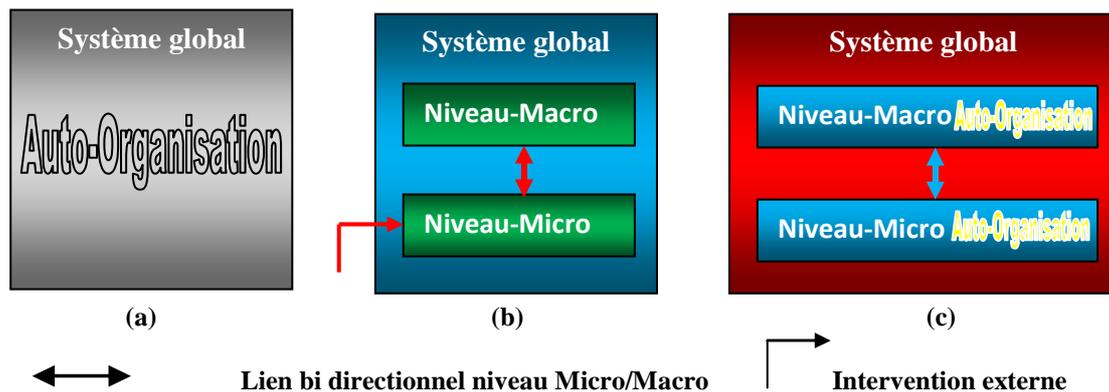


Figure 3.3 (a) auto-organisation sans émergence; (b) émergence sans auto-organisation; (c) émergence et auto-organisation combinées

La robustesse de détermination des solutions auto-organisées et émergentes se trouve plus dans les interdépendances et la planification entre les entités ou les agents, que dans le développement interne ou la difficulté inséparable des entités ou des agents pris séparément. L'assemblage de l'émergence et de l'auto-organisation forme un accès prometteur pour accomplir des achèvements pour des problèmes partagés et strictement actifs.

L'un des dispositifs très important, équipé évidemment par le paradigme de SMA réactifs, est la notion de stigmergie.

3.5 Stigmergie

Dans les *SMA* établis, l'environnement forme un élément fondamental au système. L'attraction biologique (éthologique en singulier), réaffirme cette impact. Dans ce contexte, les agents établis utilisent l'environnement comme façon d'interdépendance et aussi comme support d'écrit des conséquences, représentant des résultats et des marques de leurs faits.

Le mot stigmergie [57] est proposé par Paul Grassé en 1959, origine au comportement des termites. En faveur aux changements de la situation de l'environnement (gradients de phéromones) les termites approchent, sans normalisation centrale, à révéler et à arranger leur fait.

3.5.1 Définition

La stigmergie est l'organisation des fonctions et la normalisation des constructions ne concernent pas aussitôt des travailleurs, mais des constructions elles-mêmes. L'ouvrier ne conduit pas ses tâches, il est éclairé par lui. C'est cette incitation d'un caractère singulier que nous l'appelons de stigmergie.

Inspirés par la nature, les systèmes auto-organisés émergents transmettent par « stigmergie ». Le mot procède des termes grecs stigma « marque, signe » et ergon « travail, action », formulant le concept que les faits d'un agent déposent des signes dans l'environnement, signes vus par lui-même et par les autres agents et qui conduisent leurs prochaines opérations [57].

La stigmergie est un dispositif, qui admet à un environnement des 'auto-organiser à travers les faits des agents sur l'environnement. Cette organisation de l'environnement a une conséquence sur la conduite des agents qui font sur la structure de l'environnement, et ainsi de suite [55]. On participe alors à un évènement d'assemblage entre le système et son environnement, engendrant l'auto-organisation, à un degré macro, du couplage (système + environnement).

La figure 3.4 illustre le mécanisme de la stigmergie.

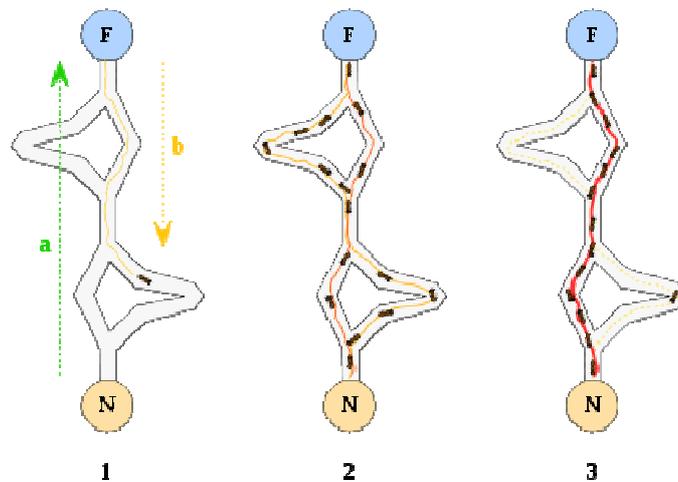


Figure 3.4 Mécanisme de la stigmergie

Ce couplage admet au système de s'auto-constituer comme un collectif auto-organisé. En d'autres termes, la stigmergie offre une méthode de planification entre les agents, provenant du système lui-même. C'est donc un moyen de correspondance indirecte dans un environnement émergent auto-organisé, où les unités transmettent entre eux en changeant leur environnement.

L'idée de stigmergie a connu un départ spécialement pour un type de problèmes d'optimisation. Cette origine a conduit certains concepteurs à produire le terme d'« optimisation stigmergique » [57]. Notamment, plusieurs études de recherches touchent l'exploitation de l'environnement pour la résolution et l'optimisation de problèmes à travers des dispositifs stigmergiques : des phéromones, des champs de forces, etc.

Parmi les approches d'optimisation qui utilise la stigmergie nous mentionnons les algorithmes de colonie de fourmis artificielles qui simulent l'aptitude des fourmis réelles à trouver des solutions aux problèmes compliqués et à s'auto-organiser par stigmergie.

3.6 Auto-organisation et SMA

Pour qu'un système présente bien la bonne conduite en fonction de l'environnement, il doit saisir cet environnement et donc être conduit par lui.

On recouvre alors le confus de la définition de l'auto-organisation en ce qui concerne l'influence extérieure dans la large palette de marches possibles au sein des

SMA acheminant de la vérification avec contrôle central jusqu'à des systèmes dont les agents ne se préoccupent presque pas de l'environnement.

Les *SMA* offrent bien tous les mécanismes essentiels pour créer des systèmes fondés sur la notion d'auto-organisation, et ces mécanismes sont robustes, efficaces à utiliser et communs.

3.7 Quelques techniques métaheuristiques

L'utilisation de l'émergence et de l'auto-organisation concerne un certain nombre de méthodes métaheuristiques, comme notamment, les Algorithmes Génétiques, les Réseaux Neuronaux, les Algorithmes de Colonies de Fourmis et l'Optimisation par Essaim Particulaires. Nous présentons dans cette section les techniques les plus utilisées.

3.7.1 Algorithmes génétiques

Le calcul évolutionnaire se base en premier sur les théories Darwiniennes de transformation des groupes par des dispositifs d'héritage génétique et de choix naturelle. Il se base sur un ensemble de moyens de détermination et d'amélioration stochastique qui montrent un prototype réduit des fondements biologiques réunis et peut alors être observé comme du "Darwinisme Artificiel".

Le fondement essentiel est l'utilisation de populations d'individus qui indiquent des aspects de l'espace d'étude pour un problème présenté.

Il s'agit alors d'exécuter stochastiquement un nombre d'actions sur chacun des unités tout en les regroupant par *générations* du mouvement simulé de changement. Ces actions sont de deux modèles différents : la *sélection*, fondée sur le cas de l'exploit d'une unité au vue du problème réfléchi (fonction de *fitness*, i.e. de *coût*, *performance*, *évaluation*), et les *dispositifs génétiques*, souvent les *croisements* (*crossover*) et les *mutations*, qui créent de nouveaux individus [67].

Le processus est alors fondé à une boucle indispensable au sein de laquelle chaque répétition indique une génération et ses individus sont nommés les *parents* de la prochaine génération.

Une récurrence inclut entour (de nombreuses évolutions sont faisables), et dans l'ordre, les étapes suivantes : Le choix, la propagation, l'estimation, le emplacement et le Cas d'arrêt.

La figure 3.5 donne le principe de fonctionnement des algorithmes génétiques.

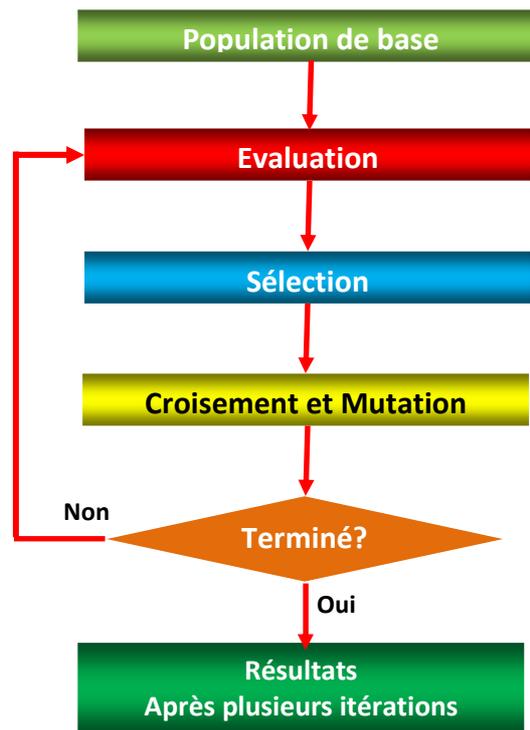


Figure 3.5 Principe de fonctionnement des AG

3.7.2 Réseaux neuronaux dynamiques

Un RN est un modèle formel réduit le dynamisme des neurones biologiques, d'autant plus réduit que l'étude sur le cerveau n'a arrêté d'améliorer et que nous nous rendons compte que les dispositifs de base ne sont en fait qu'une petite partie d'une composition énormément plus compliqué (rôle cruel de l'activité chimique, difficulté de la morphogénèse du cerveau, flexibilité permanente sur le cerveau adulte, ...).

Un RN est un réseau construit souvent d'un grand nombre d'unités de calcul très élémentaires (les neurones formels) avec chacun une mémoire étroite. La figure 3.6 représente un réseau de neurones à 3 couches.

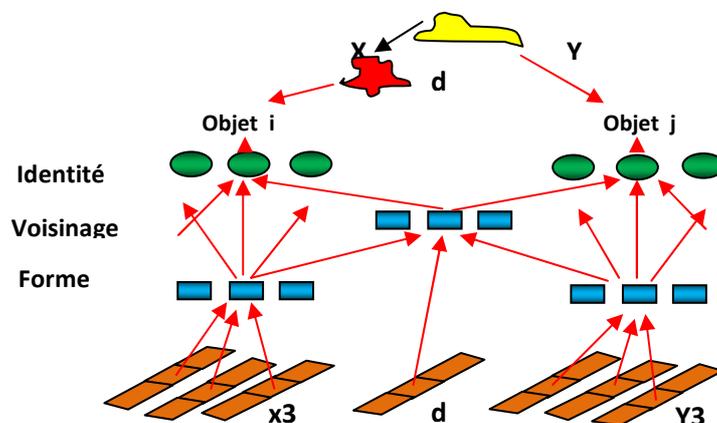


Figure 3.6 Réseau de Neurones à 3 couches.

3.7.3 Algorithmes de colonies de fourmis

Les algorithmes de colonies de fourmis (*ACF*) est une méthode beaucoup plus récente s'inspire de conduites étudiés dans la nature. Cette méthode est plus claire et particulière que les algorithmes génétiques ou les réseaux neuronaux artificiels, dans l'effet où elles s'inspirent de conduites très visible. Pour les *ACF*, les outils simulent ceux exhibés par les éthologues quant aux comportements de fourragement des fourmis [70].

Le principe de fonctionnement des *ACF* est schématisé dans la figure 3.7

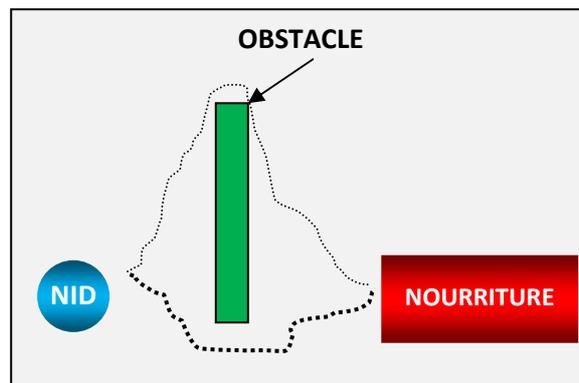


Figure 3.7 Principe de fonctionnement des *ACF*

Cette approche remarque son source dans la reproduction du fourragement des fourmis, et en singulier de leur aptitude à découvrir le plus petit chemin du nid vers la nourriture, en termes de dispositifs simples basé notamment sur le dépôt de phéromone et la direction par rapport des stigmates de phéromones présentes dans l'entourage. Les premières études sont dues à Dorigo et Maniezzo [68] et présentent une explication algorithmique qui indique des dispositifs communs de résolution de problèmes par stigmergie. La stigmergie est le fondement exposé initialement selon lequel des correspondances détournées au travers de changement de l'entourage engendrent des conduites caractéristiques sur le commun.

3.7.4 Optimisation par essais particulaire (OEP)

La similitude est ici réalisée avec les déplacements communs examinés chez certains animaux et fondée exclusivement sur l'examen local de la conduite des voisins (figure 3.8). L'accès par OEP a été notamment précisé par Kennedy et Eberhart [76] et se base sur la survenue de mouvements communs distinctifs ne dépendant que des

conduites locaux des unités par rapport aux variables tels que l'endroit et la rapidité des voisins adjacents.

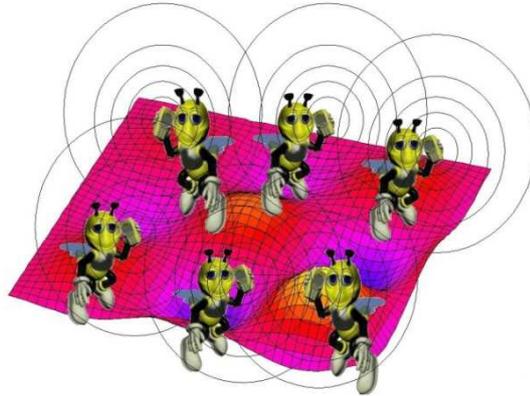


Figure 3.8 *Optimisation par Essaim Particulaire (OEP)*

L'exemple courant [71] est la conduite de déviation d'un banc de poissons séparé à un nuisible. On présume que les poissons n'ont que des conduites déterminés sous forme de «rester proche des voisins», "ne pas entrer en collision", "se bouger à la même rapidité", additionné à la déviation individuelle d'un nuisible ou d'un obstacle quand celui-ci est raisonnablement proche pour être repéré.

Il est alors facile d'envisager l'éreintement de l'approche d'un nuisible pour les tournées du banc complet : les poissons distinguant le nuisible réalisent des itinéraires pour l'empêcher et les autres poissons se rassurent de reproduire leurs actions de proche en proche. Les groupes du banc de poisson réalise alors une tournée de déviation alors qu'il n'a été adapté que par quelques poissons.

La majorité des poissons ont pu d'ailleurs ne pas se rendre compte de l'existence du nuisible, si ce n'est par l'accroissement des déplacements. L'algorithme formel se base sur l'interprétation de particules exploité un vecteur de déplacement ainsi que des "voisins". Ce voisinage peut prendre des aspects variés et n'est pas nécessairement interstellaire. Le mouvement du vecteur de déplacement est alors compté en par rapport aux déplacements antérieurs couplés avec les déplacements des particules voisines.

Si l'on augmente une estimation de l'emplacement des unités, et que cette estimation agit pour la prise en compte de l'effet des autres voisins d'une particule donnée, le fondement de l'OEP admet alors d'explorer le champ d'étude pour aboutir à de meilleures emplacements, pour une unité donné ou pour le commun. Les notes antérieures découvrent totalement le domaine et une réunion peut se découvrir dans [77].

3.8 Conclusion

Nous avons présenté dans ce chapitre les notions d'émergence et d'auto-organisation et la relation entre ces deux notions.

Parfois la notion d'auto-organisation est associée à la notion d'émergence. On trouve fréquemment la définition suivante dans la littérature : un phénomène est dit émergent lorsqu'on ne pouvait pas prédire son observation à partir de la seule connaissance du système au sein duquel il apparaît.

Une méthode d'émergence est un processus de résolution de problème qui s'appuie sur un ensemble de règles simples pour faire émerger un comportement global plus complexe sans que ce dernier n'ait été explicitement détaillé.

Nous avons présenté quelques méthodes métaheuristiques qui utilisent l'émergence dans son fonctionnement, comme par exemple, le calcul évolutionnaire, les réseaux neuronaux, les algorithmes de fourmis et l'optimisation par essaim particulaire.

Nous allons présenter dans le chapitre quatre une méthode d'émergence, c'est les systèmes immunitaires artificiels.

Le choix de cette technique est basé sur le fait que peu de travaux réalisés avec cette méthode dans le domaine industriel, et plus particulièrement dans l'intégration des politiques de maintenance systématique dans le l'ordonnancement.

En plus nous pensons que l'hybridation de cette technique avec les SMA peut donner des résultats très satisfaisants.

Chapitre 4 :

Algorithmes du Système Immunitaire Artificiel

Chapitre 4 : Algorithmes du Système Immunitaire Artificiel

Résumé

Le système immunitaire d'un organisme est un système biologique constitué d'un ensemble coordonné d'éléments de reconnaissance et de défense qui discrimine le « soi » du « non-soi ».

Un système immunitaire artificiel (SIA) est une catégorie d'algorithme inspirée par les principes et le fonctionnement du système immunitaire naturel des vertébrés. Ces algorithmes exploitent typiquement les caractéristiques du système immunitaire pour ce qui est de l'apprentissage et de la mémorisation comme moyens de résolution de problèmes.

L'objectif de ce chapitre est de présenter les notions essentielles des systèmes immunitaires naturels et les bases théoriques sur lesquelles est réalisée la conduite de ces derniers.

Nous présentons ensuite les SIA qui sont l'application des SIN pour la résolution de problèmes complexes.

L'hybridation des SIA avec les SMA sera l'une de nos approches pour résoudre le problème d'intégration des politiques de maintenance systématique dans l'ordonnancement.

4.1 Introduction

L'être humain est composé par un ensemble de systèmes compliqués et très divers, chacun d'eux réalise une fonction bien claire. Ces divers systèmes sont dirigés par le cerveau du corps, assurant ainsi une bonne correspondance et gestion des événements.

Dans cette partie nous découvrons les notions indispensables des systèmes immunitaires naturels et les fondements théoriques sur lesquelles est fondée la conduite de ces derniers.

Ensuite nous présentons les systèmes immunitaires artificiels qui sont l'application des systèmes immunitaires naturels pour la résolution de problèmes complexes.

4.2 Systèmes immunitaires biologiques

4.2.1 Historique et définitions

Les années soixante sont habituellement observées comme la naissance de l'immunologie moderne. Rodney Porter et Gerald Edelman arrivèrent à expliquer la composition des anticorps entre 1959 et 1961, et furent conquérants du prix Nobel de médecine en 1972.

Environ 1960, les groupes scientifiques apercevaient, grâce aux études de Jacques Miller, différents spécificités principales des cellules immunitaires. En 1989, Charles Janeway présente un prototype selon lequel ce serait l'immunité naturelle qui serait la véritable surveillante des clefs de la naissance d'une réponse immunitaire.

4.2.2 Notions sur le système immunitaire

Le système immunitaire présent est très compliqué et peut être aperçu sous plusieurs coins. Nous pouvons l'observer en tant qu'une composition d'organes, molécules ou cellules.

a. Unités immunitaires

Si nous nous présentons les parties du système immunitaire, ce dernier est composé notamment de:

- **Moelle osseuse** : Endroit de maturation des lymphocytes B.
- **Thymus** : Dans le bas du cou, forme le site de maturation des lymphocytes T.

- **Vaisseaux lymphatiques** : Portent la lymphe, les vaisseaux lymphatiques sont établis dans tout le corps.

b. Cellules immunitaires

Si par contre nous examinons les cellules du système immunitaire, celles incluses dans la défense sont les globules blancs nommées *leucocytes*.

Il aperçoit divers types de globules blancs (leucocytes), les plus essentiels sont les *lymphocytes*. Les lymphocytes sanguin [79] se compose de :

- **Lymphocytes B** : Les lymphocytes B vient leur nom de la moelle osseuse chez les mammifères.
- **Lymphocytes T** : Il distingue divers formes des lymphocytes T. Leur responsabilité est d'attaquer les cellules infectées (réponse cellulaire).
- **Anticorps** : On décrit un anticorps également d'une protéine compliquée rassemblée par les cellules du système immunitaire en réponse à la lucidité d'un corps étranger [80].

c. Antigène

Les antigènes sont souvent des protéines possédées dans des cellules ou des objets étrangers (bactéries, virus, globules rouges transfusés,...), ou visibles dans le milieu (pollens). L'effet anticorps-antigène est le fondement de l'immunité. Elle garantit notre défense contre les infections [81].

La figure 4.1 donne la structure d'un antigène.

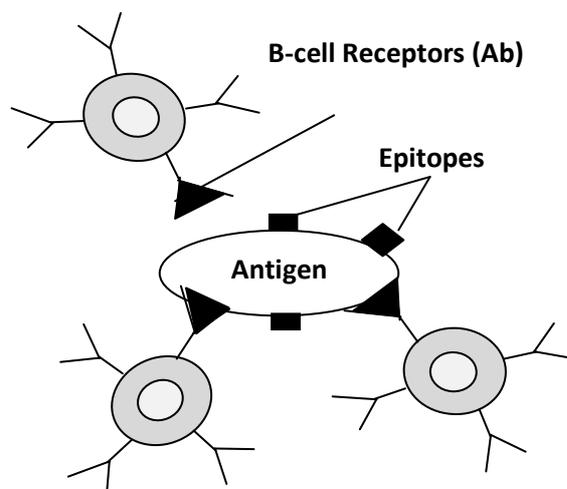


Figure 4.1 Structure d'un antigène

4.2.3 Immunité innée et adaptative

La protection de l'organisme auprès l'espace externe comprend un système immunitaire dit inné ou naturel, soutenu en absence de tout contact avec un antigène, et un système immunitaire dit adaptatif ou obtenu, apparaissant après contact de l'organisme avec des objets étrangers qui sont des antigènes [82].

a. Immunité innée

L'immunité innée est formée de la ligne physique (la peau et le système respiratoire), des situations psychosomatiques telles que le ph, la température et les cellules tueuses naturelles (NK) [82].

b. Immunité adaptative

Le système immunitaire adaptatif dérive du contact du système immunitaire avec les antigènes en faveur à la spécifique. Les lymphocytes T, les lymphocytes B et les immunoglobulines forment les acteurs primordiaux de l'immunité adaptative [84].

4.2.4 Réponses immunitaires

Selon le caractère d'infection, la réponse se décompose en :

a. Immunité cellulaire

Réclame le portage de cellules. L'immunité cellulaire est communiquée par des immunoglobulines rassemblées par des cellules [83].

La figure 4.2 présente les différents types de réponses immunitaires [83] :

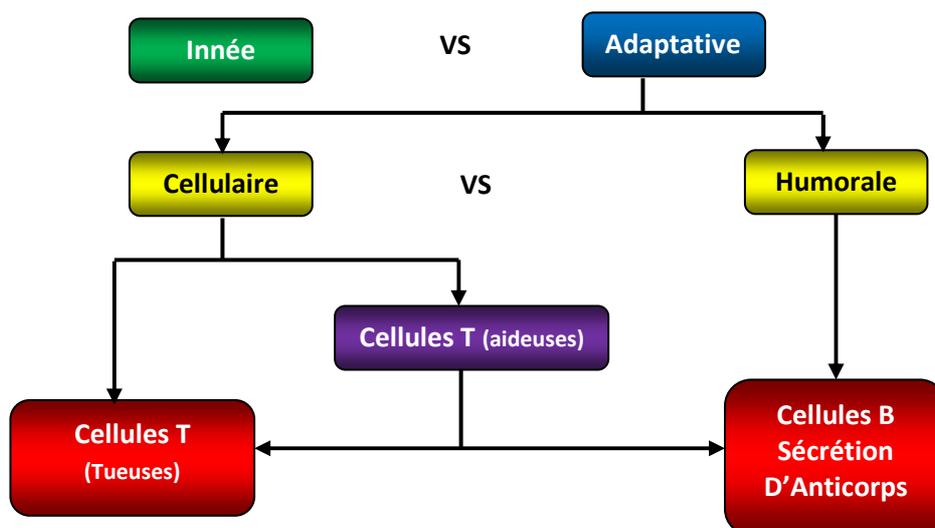


Figure 4.2 Types de réponses immunitaires

b. Immunité humorale

Adroit de contrôler les bactéries et les toxines, elle est assurée par l'enlèvement d'humeurs (vaccin ou sérum) d'un patient à un autre (ou d'un animal expérimental à un autre). Souvent l'immunité humorale est utilisée pour les maladies infectieuses [85].

4.2.5 Caractéristiques du système immunitaire

Le système immunitaire est une source d'inspiration pour les nouveaux axes de recherche. Avec des caractéristiques très intéressantes, il transforme une source utile. Nous nous limitons à citer quelques-unes de ces caractéristiques :

Distinction entre soi et non soi	C'est la caractéristique la plus importante qui est le fondement des réponses immunitaires et la capacité du système immunitaire à différencier entre les cellules du soi et les cellules du non soi (étrangères) ainsi que la potentialité de reconnaître le type vrai de chaque cellule étrangère.
Auto-organisation	L'auto-organisation précise une évolution au cours dont une composition émerge au niveau global exclusivement à partir d'un grand nombre d'interdépendances entre ses éléments de niveau local du système [67].
Distributivité	Le système immunitaire avec toute sa complication, n'inclut pas de point capital de vérification, chaque cellule est excitée et réplique à l'antigène agressant.
Apprentissage et mémorisation	Comme nous avons pu l'apercevoir auparavant, le système immunitaire est évolutif. A chaque contact d'un nouveau type d'antigènes, le système immunitaire catégorise ce dernier et le conserve en mémoire, ceci en faveur au dispositif de division cellulaire assidu d'un processus de choix afin de filtrer et de progresser la réponse du système immunitaire au contact suivant avec le même antigène [88].
Correspondance et propagation	Les distincts acteurs du système immunitaire ont nécessité d'échanger des messages sous forme de signaux.

4.2.6 Méthodes immunitaires

La conduite et les réponses du système immunitaire sont notamment gérées par des méthodes immunitaires.

4.2.6.1 Méthode de la sélection Négative

Cette méthode dirige le processus de création des lymphocytes. Plus régulièrement, cette méthode gère le processus de création au niveau de la distinction entre soi et non soi.

Les lymphocytes ont sur leurs champs des récepteurs (*Paratopes*). Les lymphocytes originaires de la moelle osseuse partent vers le thymus. A ce stade ils sont nommés *cellules T naïves ou immatures*. Leurs Paratopes supportent un processus de réorganisation génétique pseudo aléatoire, ensuite une expérience primordiale est mise en place [89].

L'expérience en question comporte à examiner si les nouveaux récepteurs s'attaquent aux cellules du soi, dans ce cas des lymphocytes sont abattus et purifiés de la multitude des nouveaux lymphocytes, on parle de sélection négative. Le reste de la population est qualifié à laisser le thymus pour circuler dans le sang et accomplir leurs fonctions de protection. Ce processus est illustré par la figure 4.3 :

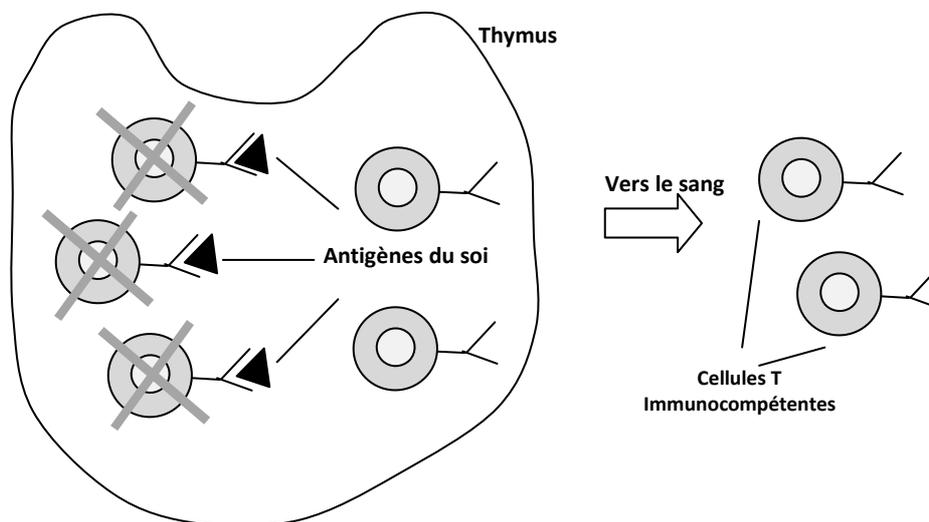


Figure 4.3 Méthode de sélection Négative

4.2.6.2 Méthode de la sélection clonale

Cette méthode a été énoncée en 1959 par Burnet [96]. Elle applique en avant la réponse du système immunitaire à un stimulus antigénique.

La méthode de la sélection clonale développe les deux processus de reproduction et de maturation d'affinité. Le principe de cette méthode est la suivante :

Dès la reconnaissance d'un antigène par les lymphocytes B, ces derniers fabriquent des anticorps particuliers (chaque cellule condense un seul type d'anticorps). L'anticorps se réunit à l'antigène à la participation des récepteurs (Paratopes-Epitopes) puis à l'aide des cellules telles que les T aideuses, les cellules B sont excitées et un processus de reproduction accepte aux cellules B de se reproduire en générant des clones d'elles-mêmes [91]. Un deuxième processus acceptera de trier parmi les nouvelles cellules celles montrant une grande affinité pour d'en faire des cellules mémoires [84].

Les cellules mémoires circulent à travers le sang, la lymphe et les tissus, et à l'existence d'un antigène auparavant reconnu, une réponse rapide et efficace est instantanément lancée [93].

Cette méthode est notamment utilisée dans les domaines tels que l'optimisation, la reconnaissance de formes et l'apprentissage-machine (IA). La figure 4.4 présente la méthode de sélection clonale.

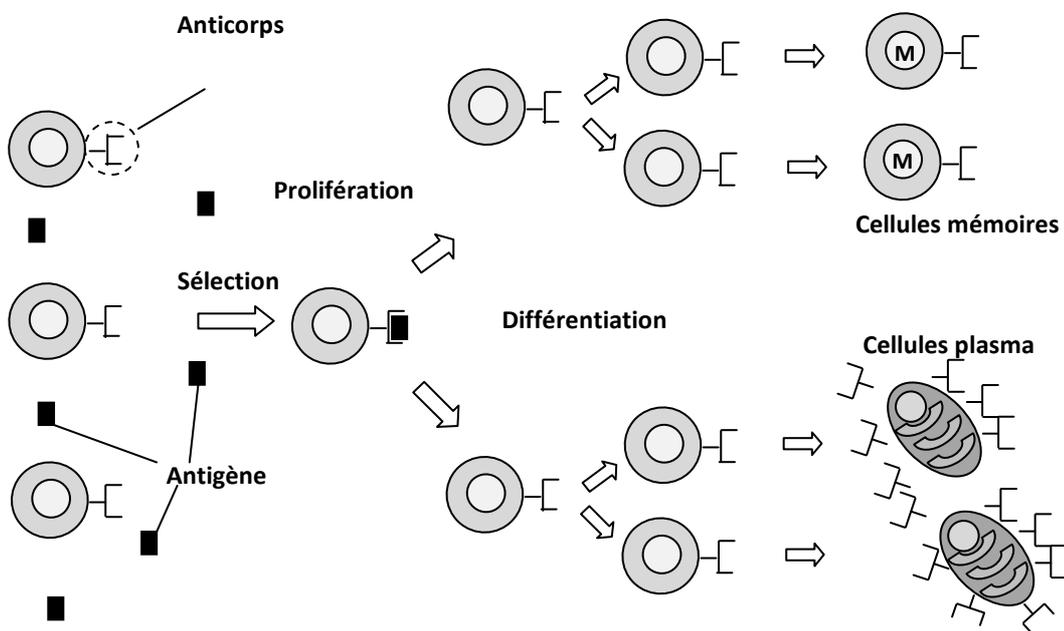


Figure 4.4 Méthode de sélection clonale

4.2.6.3 Méthode du danger

Présentée premièrement par Polly Metzinger, cette méthode est une nouvelle image qui recule de l'approche traditionnelle. La méthode du danger dirige la conduite du système et sa réponse selon les nouvelles exigences suivantes :

- Le système immunitaire ne doit pas répondre contre le soi, sauf si ce dernier est risqué.
- Le système immunitaire défend contre le non soi, sauf si ce dernier n'est pas dangereux.

4.2.7 Développement d'une réponse immunitaire

Dès la pénétration d'un corps étranger, elle se fait neutraliser par un *macrophage* qui lui extrait des portions (*peptides antigéniques*) et les exprime à sa surface sur son épiderme avec le soutien des molécules de démonstration, les molécules HLA du système d'histocompatibilité.

Cette démonstration sur l'épiderme accepte la reconnaissance du corps par les lymphocytes T (figure 4.5). Chaque lymphocyte T ne reconnaît qu'un seul modèle d'antigène ce qui admet une réponse particulière (immunité innée) qui est lente à se créer mais très influente [94].

Les lymphocytes T secondaires sont alertés et condensent des *cytokines* qui s'interprètent comme des agents de renforcement acceptant la reproduction des lymphocytes T et B. Des agents de distinction acceptant de fabriquer des anticorps (réponse humorale) [92].

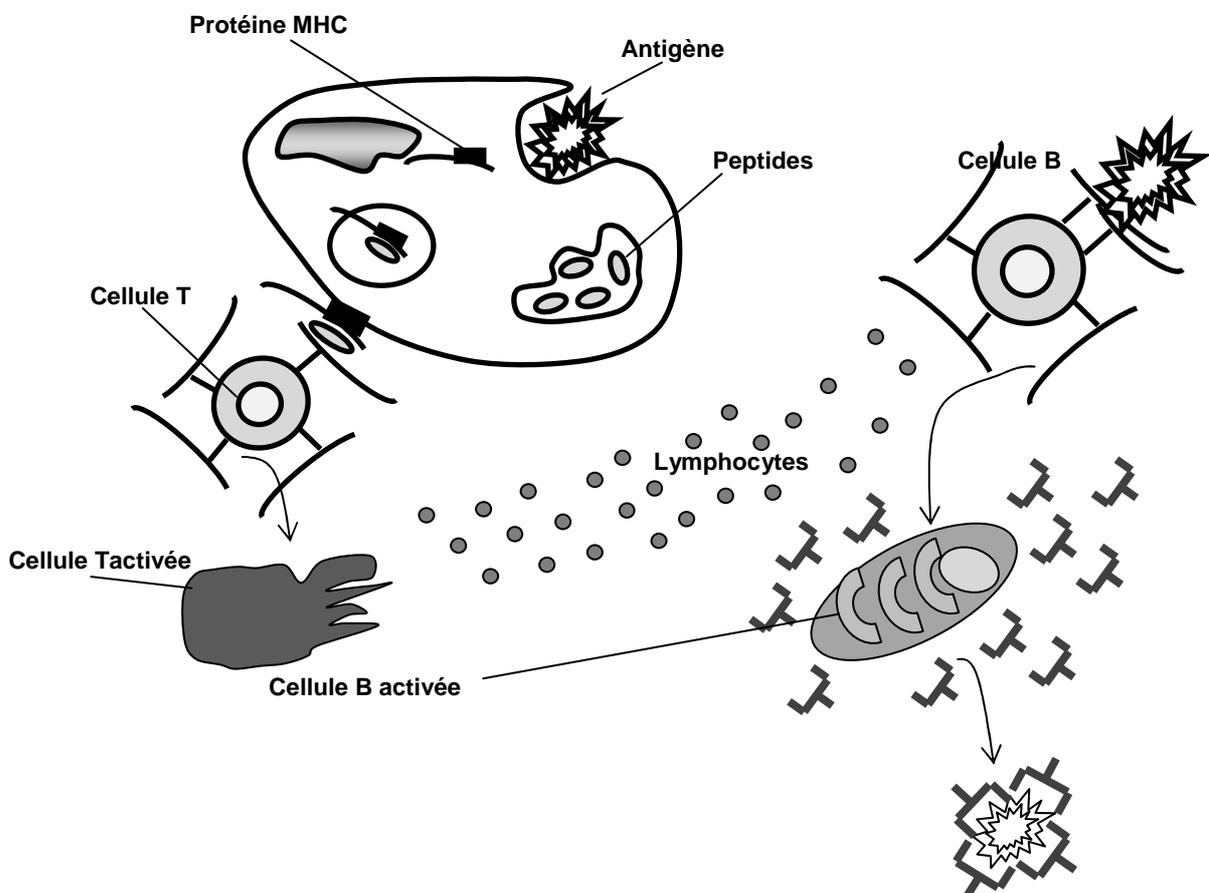


Figure 4.5 Développement d'une réponse immunitaire

4.3 Systèmes immunitaires artificiels (SIA)

4.3.1 Définition du SIA

Les *SIA* sont un ajustement mathématique ou informatique de la progression du SIN. Cette implémentation retire les plus grandes lignes de son fonctionnement.

Néanmoins, il reste environ irréalisable de spécifier le comportement entier des systèmes biologiques. Voici une définition des *SIA* :

« Les *SIA* sont des méthodes affectées ‘d’intelligence’ emportées des SIN pour la résolution des problèmes réels » [100].

Pour la modélisation des *SIA*, les chercheurs essaient au mieux de symboliser ce fonctionnement avec les façons techniques vacantes (outils de modélisation, langages de programmation, méthodes mathématiques ...etc.).

4.3.2 Spécification des SIA

Les recherches dans ce domaine sont confrontées à quelques ennuis pour les causes suivantes :

- Le nombre de personnes qui travaille dans ce domaine reste très limité [90].
- Les applications des *SIA* sont très diverses, donc il est difficile de trouver une solution complète ou des algorithmes génériques.

Ce n'est que 1999 qu'un prototype commun a été mis en place pour la représentation de ces systèmes. Ce modèle a simplifié considérablement les activités aux chercheurs [99].

La figure 4.6 montre le prototype conceptuel pour concevoir un Framework pour les *SIA*, mettant en œuvre les parties et molécules du SIN. Un ensemble d'activités d'affinité et des activités quantitatives des interdépendances entre les différents éléments sont encore à prédire.

Le modèle conjoint connu sous le nom du *Framework* des *SIA*, décrit les règles que doit respecter un *SIA* ainsi que les processus à suivre pour la réalisation de nouveaux accès. Les exigences essentielles sont :

- Le dessin des éléments du système.
- Un ensemble de dispositifs pour l'estimation de l'interdépendance entre les unités et leurs entourages.
- Des procédures d'adaptation pour vérifier la modification du système.

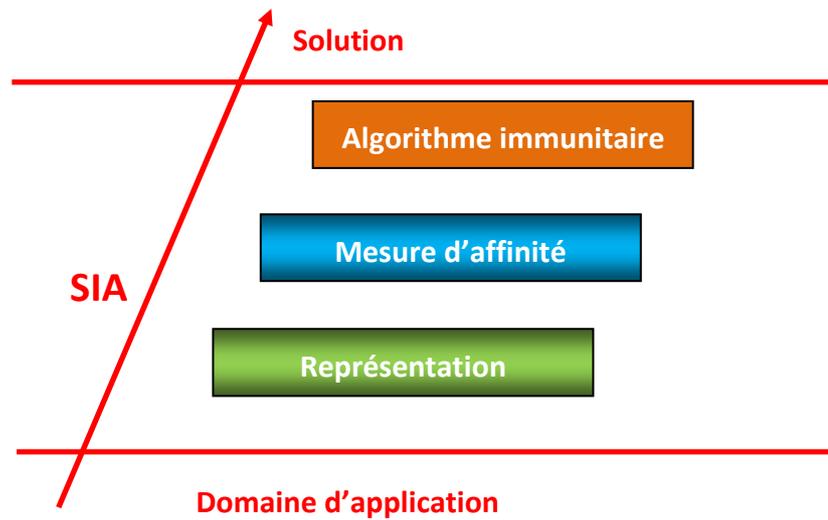


Figure 4.6 Framework pour les SIA [99]

Les trois exigences mentionnées ci-dessus sont impérieuses pour la conception d'un Framework pour développer un SIA.

Comme nous l'avons exposé auparavant, les cellules primordiales sont les lymphocytes B et T. Ces dernières sont connues pour avoir sur leurs champs des récepteurs (Paratopes) apte de reconnaître des antigènes par la caractéristique de complétude avec leurs Epitopes [93].

On peut distinguer la forme d'un anticorps comme un ensemble de L grandeurs. Ces grandeurs peuvent être indiquées par un point dans un espace de L dimensions (figure 4.7). Une première opération est que dans ce modèle, les anticorps qui se conviennent sont contigus les uns des autres.

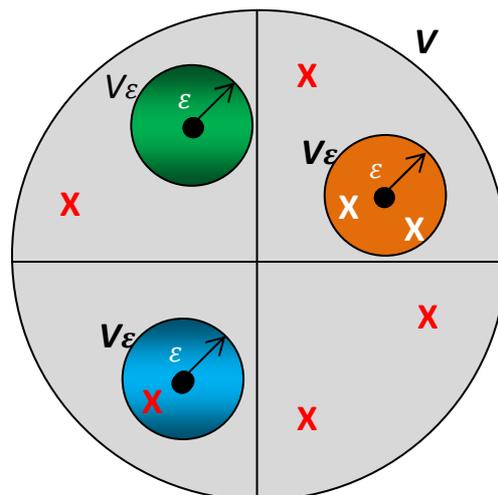


Figure 4.7 Espace des formes

Un ensemble ou une liste de N unités (récepteurs) est spécifié comme un espace de formes d'une capacité finie V contenant N points.

Ensuite un antigène est remplacé par un point $Ag = \langle Ag_1, Ag_2, \dots, Ag_l \rangle$, un anticorps est à son tour remplacé par un point $Ab = \langle Ab_1, Ab_2, \dots, Ab_l \rangle$ (Figure 4.7).

Pour estimer le niveau d'affinité entre l'antigène et l'anticorps, certaines manières peuvent être utilisées. Le plus habituel on recourt à l'utilisation des distances. Divers écarts existent dont voici les plus utilisées [104] :

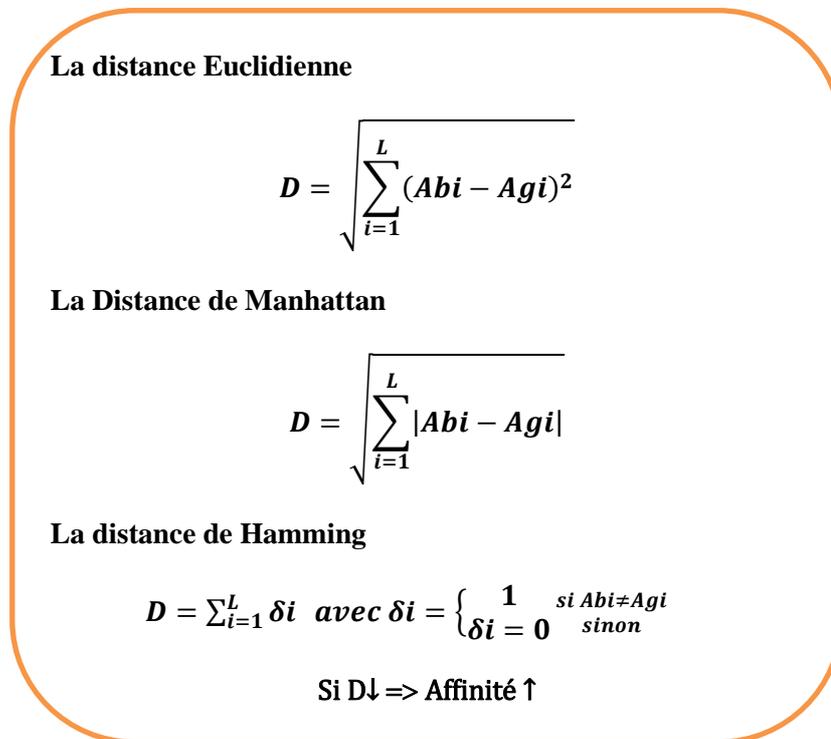


Figure 4.8 Niveau d'affinité entre l'antigène et l'anticorps

Donc on remarque bien que l'affinité anticorps-antigène est proportionnelle à l'écart dans cet espace entre ces derniers. En conséquence, plus l'écart antigène-anticorps est petit, plus l'affinité entre ces derniers est plus grande.

La figure 4.9 présente le complémentarité entre l'antigène et l'anticorps.

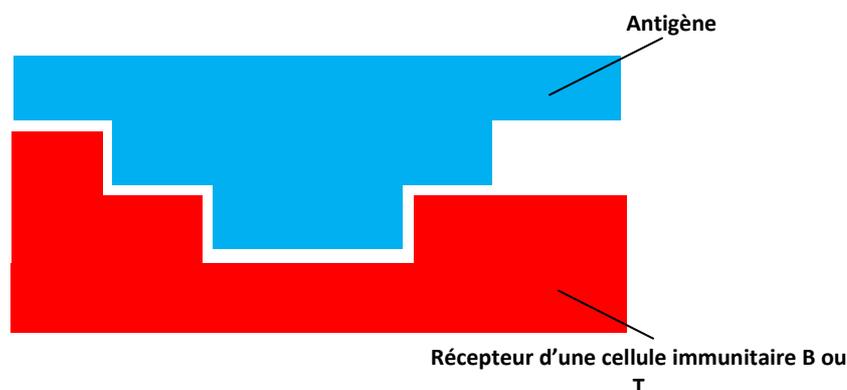


Figure 4.9 Complémentarité entre antigène et anticorps

4.4 Algorithmes du SIA

Il existe différentes utilisations des méthodes immunitaires suivant le contexte et le problème à résoudre.

4.4.1 Algorithme de la sélection négative

Cet algorithme est utilisé premièrement dans le domaine de la sécurité informatique. L'idée essentielle sur laquelle se fonde cette algorithme est que seules les cellules T qui ne s'attaquent pas aux cellules du soi sont capables à quitter le thymus et posséderont pour fonction de reconnaître les cellules du non soi.

Cette idée est très intéressante, notamment pour les applications de surveillance des systèmes et la détection d'utilisations anormales ou inhabituelles [91].

Forrest et al (1994) [110] comparent le problème de protection des systèmes informatiques au problème d'apprentissage de la diversification entre soi et non soi. Plus nettement, ils comparent le problème de détection des modifications au sein des systèmes au processus de sélection négative ayant emplacement au niveau du thymus [64].

Développée en 1994, une méthode de détection d'anomalies fondée sur la sélection négative des cellules T dans le thymus, cette méthode ainsi que toutes les autres méthodes utilisant l'algorithme de sélection négative se développent en deux cycles :

Cycle 1 : Reproduction d'un ensemble de détecteurs.

Cycle 2 : Mise en place des détecteurs, afin d'observer les données (Figure 4.10).

Pour la comparaison, on compte le nombre de bits en communs entre une donnée et un détecteur, s'il est supérieur à un seuil r on dit qu'il y'a communication entre les deux. Voici un résumé de l'algorithme de sélection négative [107].

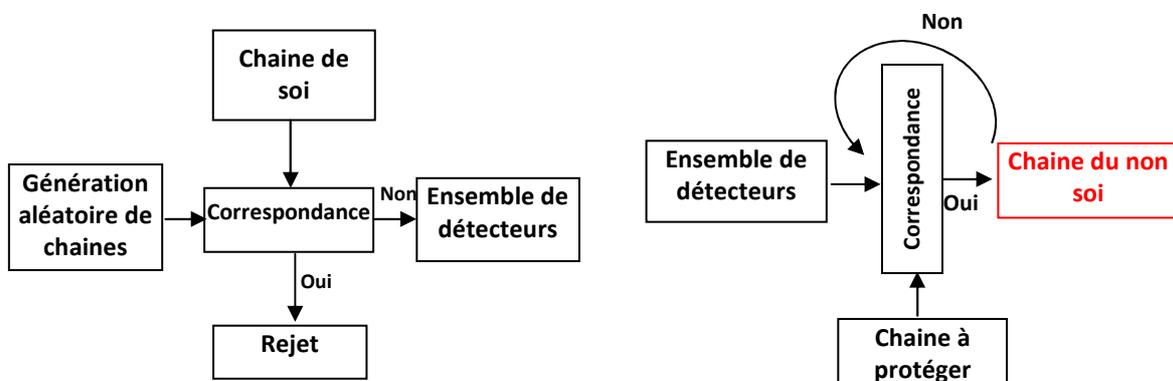


Figure 4.10 Algorithme de sélection Négative

Afin de résumer l'algorithme, De Castro [82] a proposé le pseudo code suivant :

Extrait de l'algorithme de DeCastro
<p>Pour chaque individu de P Faire</p> <p>déterminer son affinité avec tous les éléments de l'ensemble du soi S.</p> <p>Si l'affinité est supérieure à un seuil r, alors</p> <p>l'élément de P reconnaît un élément du soi et doit être supprimé de P,</p> <p>Sinon</p> <p>il est considéré comme un détecteur du non soi et sera mis dans N.</p> <p>Fin Si</p> <p>Fin Pour</p>

4.4.2 Algorithme de la sélection clonale

Comme nous avons pu l'apercevoir dans les parties préalables, cette théorie se base sur le fondement que seules les cellules ayant reconnu l'antigène prolifèrent et mûrissent et deviennent des cellules mémoires.

Il existe diverses utilisations de la théorie de la sélection clonale selon les contextes, que ce soit dans la reconnaissance de formes, l'apprentissage-machine, l'optimisation multi modale ou la détection d'anomalies [104], les traitements de cette théorie retirent les étapes importantes du développement de cette dernière. Les deux algorithmes les plus connus sont CLONALG et CLONCLASS,

CLONALG est appelé CSA (Clonal Selection Algorithm) est fondé sur les principes suivants :

- Conserve d'un ensemble de cellules mémoires.
- Choix et clonage des anticorps les plus stimulés.
- Re-sélection des clones relativement à l'affinité avec l'antigène.
- Suppression des anticorps non stimulés.
- Maturation de l'affinité de ces derniers [105].

C'est l'algorithme que nous allons utiliser dans notre approche avec quelque adaptation. Le choix de clonalg pour l'intégration des politiques de maintenance systématique dans l'ordonnement, est justifié par son capacité d'apprentissage, son

mémorisation, sa robustesse et sa flexibilité, qui rendent CLONALG utile pour les problèmes d'ordonnement.

L'algorithme de CLONALG est présenté de la manière suivante :

Algorithme de CLONALG

- Créer un ensemble de taille N d'anticorps

- Un sous ensemble de taille m contiendra les anticorps mémoires qui représenteront la solution du problème.

Tant que la condition d'arrêt n'est pas vérifiée **Faire**

 Sélectionner un antigène a de l'ensemble des Ags .

Pour chaque anticorps c de l'ensemble des anticorps Abs , **Faire**

 calculer son affinité avec a .

 Sélectionner les n meilleurs anticorps selon leurs affinités.

 Cloner les n meilleurs anticorps.

 Appliquer le processus de maturation de l'affinité sur les clones pour augmenter leurs degrés de correspondance avec l'antigène a .

 Exposer les clones de nouveau à l'antigène a et recalculer leurs affinités.

 Les meilleurs clones seront placés dans m .

 Les r anticorps de N les plus faibles seront remplacés par d'autres générés aléatoirement.

Fin pour

Fin Tant que

- L'ensemble m est considéré comme solution de l'algorithme.

- Selon le problème la solution globale peut être le meilleur individu ou une collection d'individus de m .

4.4.3 Algorithme de réseau immunitaire

Le réseau immunitaire décrit la manière dont les cellules répondent entre elles dans le système immunitaire. Cette méthode du réseau immunitaire, a été présentée par Jerne[101] en 1976.

Elle est très difficile à démontrer expérimentalement, c'est pourquoi elle exista généralement le sujet de reproches de la part de quelques immunologistes.

La façon est considérablement la même que celui de la sélection clonale, sauf qu'il aperçoit un dispositif de suppression qui abattu les cellules qui ont un certain seuil d'affinité entre elles.

L'algorithme aiNet [105] a pour objectif de borner la répétition dans l'ensemble de la population donnée en entrée de l'algorithme. Voici sa version réduite telle que la donne Timmis :

Extrait de l'algorithme de Timmis
<ul style="list-style-type: none">- Entrée: Un ensemble G de formes à reconnaître- Entrée: Un ensemble N de détecteurs aléatoire- Entrée: n le nombre de meilleurs récepteurs- Sortie: M un ensemble de détecteurs générés capable de reconnaître les formes entrantes.- Initialisation aléatoire d'une population d'individus B <p>Pour Chaque forme à reconnaître faire</p> <p> Calculer l'affinité pour chaque membre de B avec chaque détecteur de N.</p> <p> Cloner et muter chaque n selon le degré d'affinité.</p> <p> Sélectionner les n membres de B ayant la meilleure correspondance avec les formes.</p> <p> Placer l'élément n ayant la plus forte affinité et l'ajouter à l'ensemble M.</p> <p> Parcourir l'ensemble Met retirer les membres les plus faibles (la plus faible affinité).</p> <p> Générer des éléments b et les ajouter à l'ensemble B.</p> <p>Fin pour</p>

4.5 Systèmes immunitaires et algorithmes immunitaires

Une comparaison entre les éléments des systèmes immunitaires et leurs équivalents dans les algorithmes immunitaires présentés dans le tableau 4.1.

Tableau 4.1 *Comparaison entre les systèmes immunitaires et les algorithmes immunitaires*

Systèmes immunitaires	Algorithmes immunitaires
Antigènes	Problème à résoudre
Anticorps	Vecteur des meilleures solutions
Reconnaissance d'antigènes	Identification du problème
Production d'anticorps à partir des cellules mémoires	Chargement des meilleures solutions préalablement trouvées.
Suppression des cellules T	Elimination du surplus des solutions potentielles.
Prolifération d'anticorps	Utilisation d'un processus pour la création de copies exactes de la solution

Selon le tableau 4.1, on peut faire des comparaisons entre un algorithme immunitaire et son application réelle dans notre approche.

Par exemple, l'antigène qui correspond au problème à résoudre, dans notre approche, représente la valeur du Makespan.

L'anticorps qui correspond au vecteur de meilleures solutions, dans notre approche, représente à une instance d'un benchmark.

4.6 Conclusion

Dans ce chapitre nous avons abordé les notions principales du SIN à savoir les parties le composant (thymus, moelle osseuse) ou ses cellules (lymphocytes, macrophages).

Nous avons vu la fonction de chacun de ces éléments, ainsi que l'interdépendance qui se trouve entre eux afin d'établir les deux grandes barrières de protection, connues sous le nom d'immunité innée et adaptative.

Le fonctionnement de ce système est fondé sur des théories telles que la sélection clonale et la sélection négative, qui après certaines études, ont donné naissance aux systèmes immunitaires artificiels.

En conséquence les *SIA* sont une adéquation du système immunitaire biologique, essayant de spécifier son fonctionnement afin de découvrir des solutions à des problèmes tout aussi variés que compliqués.

Les caractéristiques des *SIA* sont très intéressantes, tel que la capacité d'apprentissage, la mémorisation, la robustesse et la flexibilité.

Toutes ces caractéristique rendre cette méthode applicable notamment, pour l'intégration des politiques de maintenance systématique dans l'ordonnancement.

Nous allons proposer une approche hybride entre les *SIA* et les *SMA* pour la maintenance systématique et plus particulièrement, l'intégration de leurs politiques dans l'ordonnancement.

Le chapitre suivant donne une présentation détaillé des méthodes de résolution du problème d'ordonnancement de type flow shop.

Chapitre 5 :

Méthodes de résolution du problème d'ordonnancement de type flow shop

Chapitre 5 : Méthodes de résolution du problème d'ordonnancement de type flow shop

Résumé

Parmi les problèmes de la maintenance industrielle, nous pouvons citer l'intégration des politiques de maintenance préventive systématique dans l'ordonnancement.

L'objectif de ce chapitre est de présenter la maintenance systématique et leurs politiques utilisées dans l'intégration dans l'ordonnancement.

Enfin nous donnons quelques méthodes de résolution du problème d'ordonnancement flow shop pour faire des comparaisons avec nos solutions proposées.

5.1 Introduction

Aujourd'hui, il est précisément reconnu dans le contexte du secteur industriel que la fonction de maintenance, dont le but est d'assurer le bon fonctionnement des outils de production, est désormais, une des fonctions stratégiques dans les entreprises.

Cependant, les clients ont eux aussi changé dans le sens où, ils exigent des produits de haute qualité, des livraisons rapides, tout cela avec des coûts raisonnables.

Sous cette charge, les industries sont contraintes de progresser de façon régulière sur plusieurs niveaux, notamment le coût de revient, la qualité du produit et les délais de livraison.

Nous présentons dans ce chapitre les notions de la maintenance systématique et leurs politiques utilisés dans l'intégration dans l'ordonnancement. Nous allons présenter aussi, un atelier d'ordonnancement flow shop hybride. Enfin, nous donnons quelques méthodes de résolution du problème d'ordonnancement flow shop.

5.2 Maintenance préventive systématique

Les entreprises ont besoin de différents types de machines à produire des biens. Chaque machine n'est pas fiable dans le sens où il se dégrade avec l'âge et l'utilisation, et échoue finalement [148]. Les opérations de maintenance peuvent être classées en deux grands groupes: la maintenance corrective (*MC*) et la maintenance préventive (*MP*). La *MC* correspond à des actions menées lors de l'échec a déjà eu lieu. La *MP* est la mesure prise sur un système alors qu'il est encore en activité. La *MP* est effectuée afin de maintenir le système au niveau de fonctionnement souhaité. Plusieurs politiques de la *MP* peuvent être définies [140], [149], [150] et [151], dans le but de déterminer le moment où il est nécessaire de mener des opérations de la *MP* sur les machines en fonction de différents critères. Ce qui suit sont trois politiques classiques [153].

5.2.1 Politique I: Maintenance préventive à des intervalles de temps fixes et prédéfinis

Les opérations de la *MP* sont prévus à l'avance, à des intervalles de temps prédéfinis (T_{MPF}) sans tenir compte des modèles probabilistes pour le temps de l'échec et faire le meilleur usage des arrêts programmés après une semaine, un mois ou même des périodes de production annuelle cycliques. Dans cette politique, les intervalles de temps fixes sont déterminés et des opérations de la *MP* sont réalisées exactement à ces intervalles de temps. On suppose que les Jobs sont non-préemptif (le processus d'un Job

ne peut pas être interrompu), à chaque fois qu'il ya un chevauchement entre le processus d'un Job et les opérations de la MP, le processus du job est reportée et les opérations de la MP sont effectuées en premier. La procédure est illustrée par un exemple. Supposons qu'un atelier planifie ses activités de la MP en se basant sur la politique I, décide de procéder les opérations de la MP chaque 120 unités de temps. La durée des opérations MP (D_{MP}) est de 40 unités de temps. Cet atelier dispose de trois Job ($n = 3$) à traiter. Les temps de traitement sont présentés dans le tableau 5.1

Tableau 5.1 Temps de traitement pour une seule machine avec $n = 3$.

Activité	Temps de traitement
Job1	40
Job2	50
Job3	60
MP	40

Supposons que la séquence sélectionnée des Jobs est $\{2, 3, 1\}$. Les processus de Job 2 et 3 sont retardés de 110 unités de temps. Maintenant, il n'est pas possible de traiter le troisième job de la séquence (Job 1), parce qu'il y a le temps de traitement de la MP de 40 unités de temps, ce qui mettrait fin au moment de 150 tandis que $T_{MPF} = 120$. Donc, la machine serait laissée au repos pendant 10 unités de temps, puis l'opération de la MP commence au moment de la 120ème unité, et dure pendant 40 unités de temps. Après avoir effectué la MP, le Job 1 est traité. La figure 5.1 montre le diagramme de Gantt de l'exemple compte tenu de la politique I.

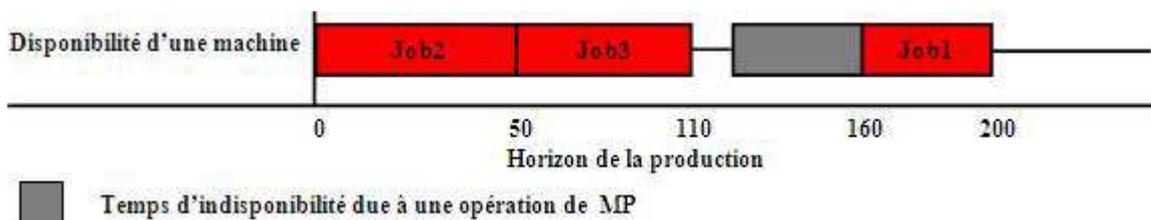


Figure 5.1 Diagramme de Gantt de la solution pour l'exemple donné.

5.2.2 Politique II: Maintenance préventive selon un modèle d'une période optimale pour maximiser la disponibilité des machines

Classiquement, la période optimale entre deux activités de maintenances préventives séquentielles est déterminée par la maximisation de la disponibilité des machines. Dans cette politique, la *MP* est effectuée en fonction de la période de maintenance optimale. L'objectif de cette politique consiste à maximiser la disponibilité du système.

Selon Kutanoglu [140], l'intervalle optimal de la maintenance T_{MPop} peut être calculé par la formule suivante:

$$T_{MPop} = \theta \cdot \left[\frac{T_p}{T_r(\beta-1)} \right]^{1/\beta}$$

Le temps de l'échec est supposé suivre une distribution de probabilité de Weibull, $T \approx W[\theta, \beta]$, avec $\beta > 1$. T_r est le nombre d'unités de temps qui prend l'intervention et T_p le nombre d'unités de temps de la *MP*. T_{MPop} est l'intervalle entre deux *MP* consécutives.

Exemple

Nous pouvons affirmer que la politique II consiste à effectuer une *MP* chaque fois qu'une machine fonctionne T_{MPop} unités de temps.

Par exemple, supposons que le temps à l'échec suit un modèle de Weibull, $T \approx W[\theta, \beta]$, avec $\theta = 1349\text{h}$ et $\beta = 2$, et avec $t_r = 6\text{ h}$ et $t_p = 1\text{ h}$.

Dans ce cas, l'intervalle *MP* optimale, T_{MPop} , est 550,73 h, et lorsqu'on applique la fonction de fiabilité on trouve environ une fiabilité de 85%.

Afin de mieux comprendre la procédure de l'intégration de la *MP* et l'ordonnancement de la production dans cette politique, considérons l'exemple suivant. Supposons qu'un atelier planifie ses activités de la *MP* en se basant sur la politique II, décide de procéder les opérations de la *MP* chaque 60 unités de temps ($T_{MPop} = 60$). La durée de ces activités *MP* sur la machine est de 25 unités de temps ($D_{MP} = 25$). Cet atelier dispose de quatre Jobs ($n = 4$) à traiter. Le tableau 5.2 indique les temps de traitement.

Tableau 5.2 Temps de traitement pour une seule machine avec $n = 4$.

Activité	Temps de traitement
Job1	25
Job2	30
Job3	20
Job4	30
MP	25

La séquence des Jobs est {4, 1, 3, 2}. Après le traitement des Jobs 4 et 1, le temps de traitement total cumulé est de $30 + 25 = 55$. Maintenant, il est impossible de traiter le Job 3, car il a le temps de traitement de 20 unités de temps, ce qui aboutirait à un temps de traitement total cumulé de 75 unités. Ainsi, le processus de Job3 est reporté et la *MP* est effectuée en premier. Après la réalisation du *MP* d'abord avec une durée de 25 unités de temps qui dure de 55 à 80 unités, le temps de traitement total cumulé redémarre à zéro, et le processus de Job 3 commence. Le même raisonnement est appliqué à tous les Jobs. La figure 5.2 montre le diagramme de Gantt de l'exemple compte tenu des politiques II.

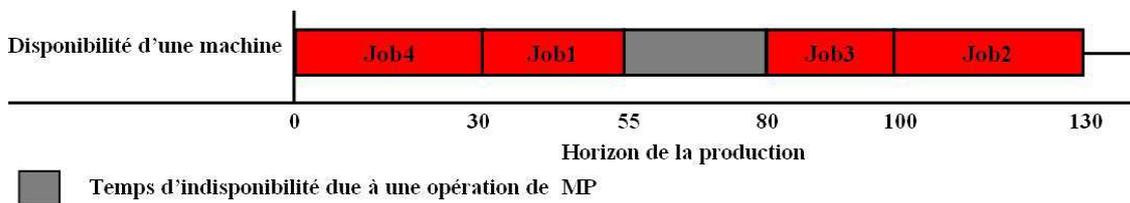


Figure 5.2 Diagramme de Gantt de la solution pour l'exemple dans la politique II

5.2.3 Politique III: Maintenance préventive selon un seuil minimum de fiabilité pour une période de production donnée t

Dans certains systèmes, le vieillissement et l'usure affectent le taux d'échec, c'est-à-dire qu'il peut être augmenté sur le temps. Cette politique consiste à effectuer une *MP* systématique après un temps T_{MP} de manière à garantir un minimum de fiabilité du système ($R_0(t)$) à partir de l'instant $t = 0$. Il est supposé que les *MP* restaurent les machines à l'état bon comme neuf. Dans ce cas, la *MP* sera réalisé à des intervalles réguliers $0, T_{MP}, 2T_{MP}, 3T_{MP}, \dots, NT_{MP}$ qui sont considérés comme des points de rénovation.

Lorsque le temps à l'échec suit un modèle de Weibull, $T \approx W [\theta, \beta]$, avec $\beta > 1$ (le taux d'échec croît avec le temps), le temps entre les MP dans cette politique peut être obtenu comme suit:

$$T_{MP} = \left[-\theta\beta \cdot \frac{\text{Ln}R_0(t)}{t} \right]^{1/\beta-1}$$

L'Intégration de la MP et l'ordonnancement de la production dans la politique III est fait de la même manière comme nous le faisons dans la politique II.

Dans les Politiques II et III, la MP dépend de l'ampleur de temps que la machine est en marche, contrairement à la politique I dans laquelle les activités de la MP sont effectuées en fonction du temps (le temps de fonctionnement des machines n'est pas important). Un autre problème est la durée des activités de la MP , ce qui est appelé D_{MP} .

5.3 Ordonnancement flow shop hybride

Le problème d'ordonnancement flow shop hybride (*OFSH*) peut être énoncé comme suit : considérons un ensemble de n jobs doivent être traitées dans m étages. Chaque étage i peut avoir plusieurs machines identiques en parallèle, notée par m_i . En *OFSH*, nous avons besoin de tous les jobs de passer à travers les étages d'un même ordre de départ de l'étage 1 à l'étage m . Chaque job peut être exploité par n'importe quelles machines tout en un seul étage, cependant, quand il est affecté à une machine, son processus ne peut pas être interrompu. Chaque machine peut fonctionner sur un seul job à la fois. Il n'ya pas de contrainte de précédence entre les jobs, c'est-à-dire qu'ils peuvent être traités dans n'importe quel ordre. Le temps de traitement de chaque job j à l'étage i (désignée par $P_{j,i}$) est fixe et connu à l'avance. Étant donné que les machines sont identiques, le temps de traitement d'un job à un étage est constant entre les machines à cet étage. En *OFSH*, il ya deux dimensions de décision:

- Séquence de job
- Affectation du job aux machines à chaque étage.

La figure 5.3 présente un schéma général d'un atelier flow shop hybride.

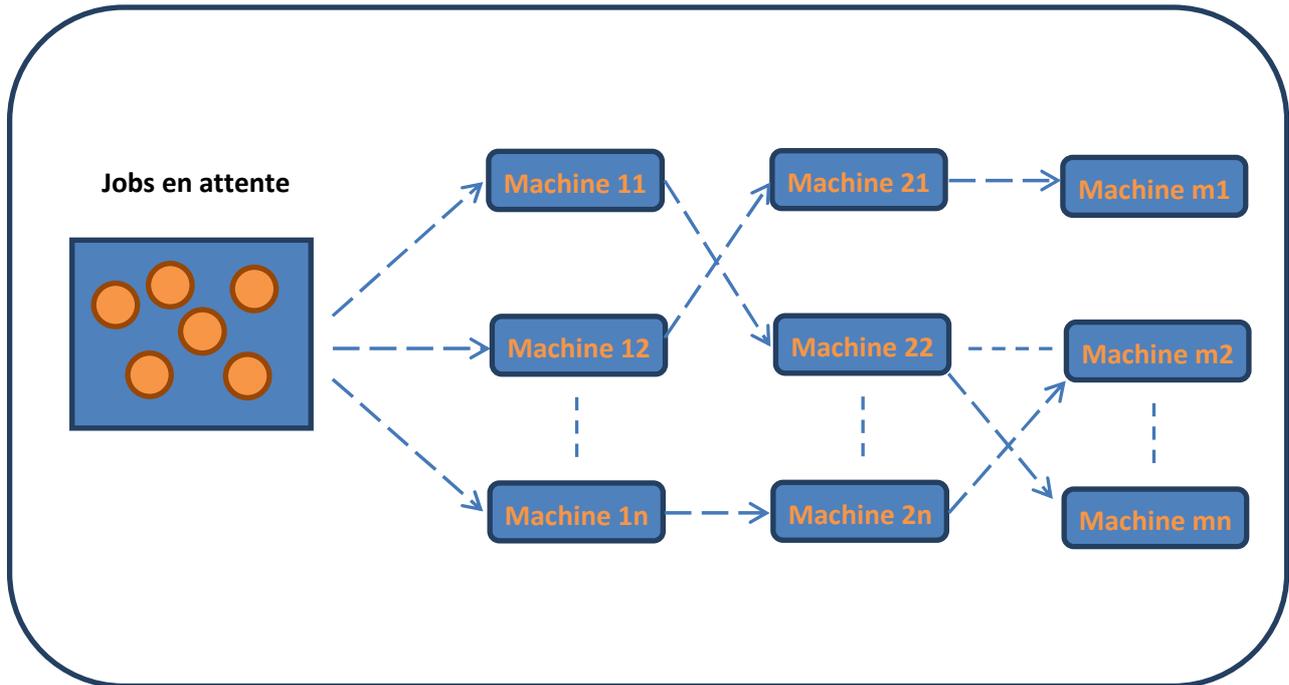


Figure 5.3 Schéma général d'un atelier flow shop hybride

5.4 Méthodes de résolution du problème d'ordonnancement flow shop

Nous allons présenter dans cette section quelque méthode de résolution du problème d'ordonnancement flow shop, existants dans la littérature. Cette présentation sera utile dans le prochain chapitre pour faire des comparaisons avec nos approches de résolution de ce problème.

5.4.1 Algorithme de Johnson

L'algorithme de Johnson est un paradoxe de l'ordonnancement. La référence de Johnson 54 est sans aucun doute la référence la plus citée du monde de l'ordonnancement. Pratiquement personne ne l'a lu, elle n'a pratiquement aucun intérêt industriel, mais l'originalité de son approche et sa simplicité font un "objet culte" [134].

L'algorithme de Johnson s'applique à un problème de flow Shop à deux machines, et le critère à optimiser est le Cmax (makespan).

Les hypothèses sont donc :

- Tous les jobs ont deux opérations, la première sur la machine 1, la seconde sur la machine 2,

- les machines ne travaillent que sur un job à la fois,
- En début d'ordonnancement, les deux machines sont libres et l'atelier est vide,
- Les deux tâches d'un job ne peuvent pas se chevaucher (on doit attendre la fin de la première pour pouvoir commencer la seconde)
- Il n'y a aucun temps de réglage, ou bien ceux-ci sont inclus dans les temps de traitements, ce qui revient au même.

Notation :

i Indice des jobs

A_i : temps de traitement du job i sur la première machine.

B_i : temps de traitement du job i sur la seconde machine.

Johnson a prouvé que pour trouver l'ordonnancement qui minimise le C_{max} (date de fin de dernier job sur la dernière machine), il suffit de classer i avant j si :

$$\min(A_i, B_j) < \min(A_j, B_i)$$

À partir de la propriété précédente, Johnson a développé l'algorithme suivant:

Algorithme de Johnson
<ul style="list-style-type: none">- Classer les jobs en deux groupes:- Dans le premier groupe, G_1, mettre tous les jobs tels que $A_i < B_i$- Dans le second groupe, G_2, mettre tous les jobs tels que $B_i \leq A_i$- Classer G_1 par plus court durée de traitement (ou PCDT) sur la machine M_1 (par A_i croissants)- Classer G_2 par plus longue durée de traitement (ou PIDT) sur la machine M_2 (par B_i décroissants)- Considérer la séquence obtenue par concaténation de G_1-G_2

Pour s'en convaincre, il suffit de comparer deux éléments i et j, i classés avant j.

- Si i et j sont dans G_1 , alors $A_i < B_i$ et $A_j < B_j$ et $A_i < A_j$ puisque G_1 est classé par PCDT sur M_1 . Donc $A_i < A_j < B_j$ et donc $\min(A_i, B_j) = A_i < \min(A_j, B_i)$.

- si i et j sont dans G_2 , alors $B_i \leq A_i$ et $B_j \leq A_j$ et $B_j \leq B_i$ puisque G_2 est classé par PLDT sur M_2 . Donc $B_j < B_i < A_i$ et donc $\min(A_i, B_j) = B_j < \min(A_j, B_i)$.
- si i est dans G_1 et j dans G_2 , alors $A_i \leq B_i$ et $B_j \leq A_j$. Donc $\min(A_i, B_j) < \min(A_j, B_i)$.

Les deux formulations sont équivalentes

5.4.2 Algorithme de NEH_H

En 1983, Nawaz, Ensore Jr. & Ham [145] ont proposé un algorithme basé sur l'hypothèse qu'un lot ayant un temps total d'exécution élevé est prioritaire (le lot est positionné en priorité dans un ordonnancement partiel) par rapport à une tâche plus faible dans le cas de la minimisation du makespan. Nous avons adapté cette heuristique pour le cas de la minimisation de la somme des retards, en privilégiant les tâches ayant la valeur :

$$Neh_i = \left(\frac{1}{W_i}\right) * (d_i - (r_i + \sum_{j=1}^m P_{ij}))$$

Algorithme de NEH _H
<p>- Ordonner les tâches selon l'ordre croissant de Neh dans T.</p> <p>- Créer une séquence vide S.</p> <p>Tant que (T non vide) faire</p> <p style="padding-left: 20px;">$T = T -$ premier élément j de T ;</p> <p style="padding-left: 20px;">Tester l'élément j à tous les emplacements dans S;</p> <p style="padding-left: 20px;">Insérer j dans S à l'emplacement qui minimise var.</p> <p>Fin Tant que.</p>

5.4.3 Algorithme de PCDT et PLDT

Il existe d'autres méthodes de résolution du problème d'ordonnancement tel que :

- 1) L'application de l'heuristique plus court durée de traitement (ou PCDT) pour OFSH: PCDT organise les Jobs dans l'ordre croissant des temps de traitement des Jobs à l'étage 1 [138].

- 2) L'application de l'heuristique plus longue durée de traitement (ou *PLDT*) pour *OFSH*: *PLDT* organise les Jobs dans l'ordre décroissant des temps de traitement des Jobs à l'étage 1[138].

5.5 Conclusion

Dans ce chapitre, nous avons présenté les principales politiques de maintenance systématique pour les intégrées dans des ateliers d'ordonnancement.

Le chapitre présente aussi un atelier d'ordonnancement flow shop hybride.

Nous avons donné quelques méthodes de résolution du problème d'ordonnancement flow shop, tels que l'algorithme de Johnson, de NEH_H , *PLDT* et *PCDT*. Chacune de ces heuristique donne de bons résultats, mais aucune ne garantie la solution optimale.

En référence aux inconvénients de ces méthodes, le chapitre suivant applique des critères simples à comprendre et facile à mettre en œuvre, mais absolument adaptable à d'autres problèmes d'ordonnancement de la machine.

Il explore l'ordonnancement flow shop hybrides et des activités de la maintenance préventive systématique en même temps, et pour résoudre le problème, nous proposons deux approches multi agent basé sur deux méthodes d'émergence, les algorithmes génétiques et les algorithmes du SIA.

Chapitre 6 :

**Approches développées pour l'intégration
de politiques de maintenance systématique
dans l'ordonnancement**

Chapitre 6 : Approches développées pour l'intégration de politiques de maintenance systématique dans l'ordonnancement

Résumé

Dans ce chapitre nous proposons deux approches multi agent basée sur deux méthodes d'émergence, les AG appelée FSHMAGA et les SIA appelée FSHMAIA.

Ces deux approches permettent de résoudre le problème d'intégration des politiques de la maintenance systématique dans un atelier d'ordonnancement.

Comme cas d'application, nous avons pris l'ordonnancement d'un atelier Flow shop hybride.

Ces deux approches sont développées sous Netlogo, environnement de modélisation programmable permettant de simuler des phénomènes naturels et sociaux.

Nous avons fait des comparaisons entre nos approches et des heuristiques développées sous Matlab, à savoir, l'algorithme de Johnson, algorithme NEH_H , PCDT et PLDT.

A la base de ces comparaisons, nous avons démontré que nos approches donnent des meilleurs résultats par rapport aux autres heuristiques.

6.1 Introduction

L'une des hypothèses de l'ordonnancement les plus étudiées envisage que les machines ne peuvent pas être périodiquement disponibles au cours de l'ordonnancement de la production. Bien que de nombreux chercheurs ont tenté d'intégrer la production et la planification de la maintenance préventive par des différentes méthodes, certaines de ces méthodes sont très complexes.

Ce chapitre propose d'appliquer des méthodes d'intégration simples, pourtant facilement extensibles à d'autres problèmes d'ordonnancement de la machine. Ce chapitre examine l'intégration des politiques de maintenance systématique dans l'ordonnancement d'un atelier flow shop hybrides.

L'objectif est de minimiser les temps d'exécution. Deux approches multi agent basées sur deux méthodes d'émergence, les AG et les SIA, et certaines heuristiques constructives sont développées pour s'attaquer au problème. La première approche s'appelle : *Flow shop Scheduling Hybrid Multi Agent Genetic Algorithm (FSHMAGA)*. La deuxième approche s'appelle : *Flow shop Scheduling Hybrid Multi Agent Immune Artificiel (FSHMAIA)*

6.2 Approche FSHMAGA proposée

Les AG cherchent un espace de problème avec une population de chromosomes dont chacun représente une solution codée. Une valeur de Fitness est assignée à chaque chromosome en fonction de sa performance. Le plus souhaitable est le chromosome qui a la plus petite valeur. La population évolue par un ensemble d'opérateurs jusqu'à ce qu'un critère d'arrêt soit visité.

6.2.1 Organigramme de l'approche FSHMAGA proposée

La figure 6.1 montre un organigramme de notre approche baptisée *FSHMAGA* (*Flow shop Scheduling Hybrid Multi Agent Genetic Algorithm*) pour l'intégration des politiques de maintenance systématique dans l'ordonnancement.

Chaque agent de notre approche exécute le même processus, c'est le calcul de la fonction objective (Fitness), qui correspond dans notre cas à la valeur du makespan.

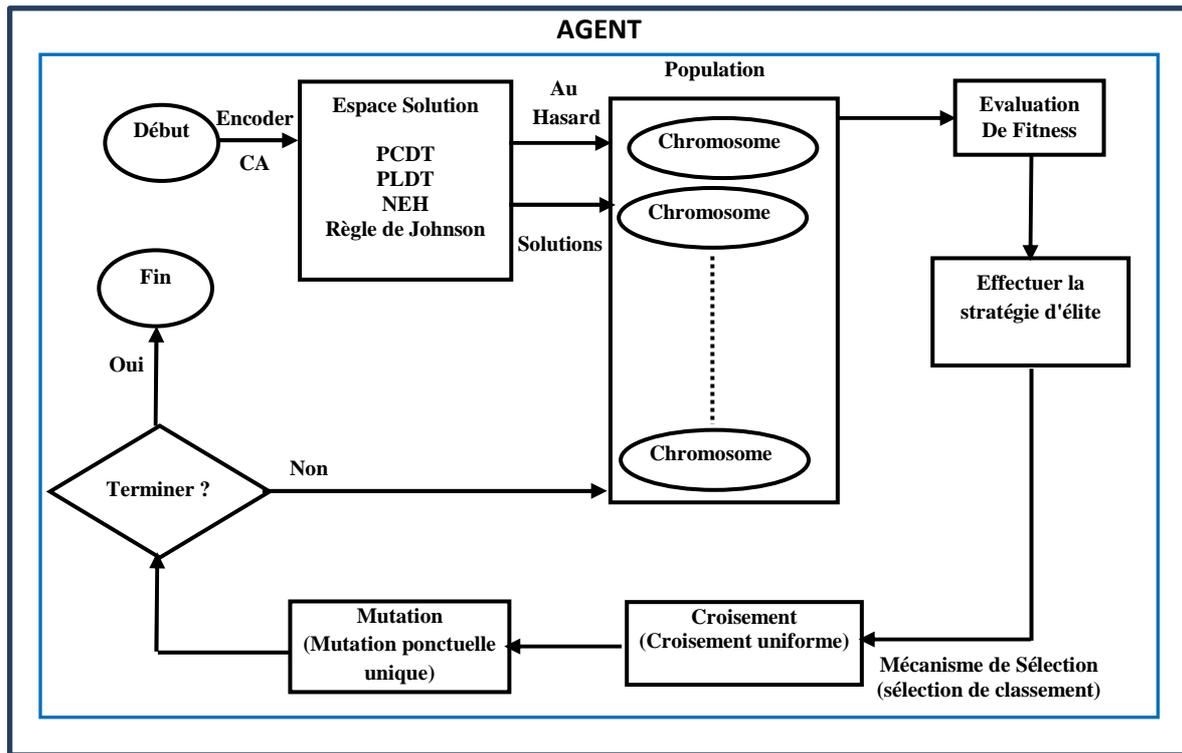


Figure 6.1 Organigramme de l'Approche FSHMAGA proposée

Les améliorations appliquées sur l'AG utilisé sont les suivantes :

- ✓ L'application de la *clé aléatoire* (CA) comme technique de codage de la population initiale. Puisque les solutions initiales peuvent fortement influencer sur les résultats finaux obtenus par les AG. Quatre solutions sont produites par les heuristiques de PCDT, PLDT, règle de Johnson ($m / 2, m / 2$) et NEH_H , et le reste est généré de façon aléatoire.
- ✓ Effectué la **stratégie d'élite** pour copie les meilleurs chromosomes de la population actuelle (Nr individus) directement dans la prochaine génération. L'objectif est de proposer des chromosomes à faible makespan.
- ✓ Comme mécanisme de sélection, nous avons utilisé *la sélection de classement*. Ce mécanisme de sélection choisit les chromosomes de la population actuelle de telle sorte qu'un chromosome avec la valeur de Fitness la plus basse comporte plus de chances d'être sélectionné.
- ✓ Le choix du *croisement uniforme paramétré* pour l'opération de croisement du parent. L'objectif est de créer de meilleures séquences en combinant les parents

- ✓ Le choix de la *mutation ponctuelle unique* pour l'opération de mutation. Le but principal de l'application de la mutation est d'éviter la convergence vers un optimum local et de diversifier la population.
- ✓ Aussi notre apport réside essentiellement dans de l'hybridation de l'AG amélioré avec un SMA.

6.2.2 Algorithme de l'approche FSHMAGA proposée

L'algorithme de l'approche proposée se résume de la manière suivante :

Algorithme FSHMAGA
<p>Pour chaque agent Faire</p> <p>Produire un ensemble de chromosomes (<i>TP</i>) comme une population initiale</p> <p>Tant que un critère d'arrêt n'est pas atteint faire</p> <p style="padding-left: 20px;">Calculer les valeurs de Fitness</p> <p style="padding-left: 20px;">Effectuer la stratégie d'élite // <i>Nr personnes</i> sont copiés dans la prochaine génération</p> <p style="padding-left: 20px;">Pour $I = Nr + 1$ à <i>TP</i> faire</p> <p style="padding-left: 40px;">Sélectionnez deux parents en utilisant un mécanisme de sélection</p> <p style="padding-left: 40px;">Effectuer le croisement uniforme sur deux parents sélectionnés et générer le <i>i</i>-ème descendant</p> <p style="padding-left: 40px;">Si $rand < Pm$ alors // <i>rand</i> est un nombre aléatoire uniformément répartie (0, 1)</p> <p style="padding-left: 60px;">Effectuer une mutation ponctuelle sur le <i>i</i>-ème descendant</p> <p style="padding-left: 40px;">Fin si</p> <p style="padding-left: 20px;">Fin pour</p> <p style="padding-left: 20px;">Fin tant que</p> <p>Fin Pour</p> <p>fin</p>

Chaque agent de notre approche exécute les opérations suivantes :

- ✓ Produire un ensemble de chromosomes (*TP*) comme une population initiale.
- ✓ Calculer les valeurs de Fitness (Makespan).
- ✓ Sélectionnez deux parents en utilisant un mécanisme de sélection.
- ✓ Effectuer le croisement uniforme sur deux parents sélectionnés.

- ✓ Effectuer une mutation de type ponctuelle unique.

Ce travail se répète jusqu'à l'arrivée à l'agent qui a la valeur de fitness la plus faible.

6.2.3 Schéma de codage et Opérateurs de l'approche proposée FSHMAGA

Dans les AG, la représentation des chromosomes est en forme de chaînes binaires composées de 0 et de 1, ce qui n'est pas évidemment approprié pour décrire l'ordonnancement flow shop hybride, car il est assez gênant pour représenter et exploiter pour l'ordonner dans cette forme.

6.2.3.1 Clé aléatoire

Le schéma de codage le plus couramment utilisé est la clé aléatoire (CA), la première représentation proposée par Norman et Bean [154] pour les problèmes de plusieurs machines identiques et utilisée plus tard par [153] et [155]. Les avantages les plus importants de ce type de schéma de codage sont d'être simples à mettre en œuvre et facilement adaptables à tous les opérateurs. L'algorithme de la clé aléatoire pourrait être décrit comme suit:

Algorithme Clé aléatoire
<p>Pour Chaque Job Faire</p> <p>Attribuer un nombre réel dont la partie entière est le numéro de la machine à laquelle le Job est attribué</p> <p>La partie fractionnaire est utilisée pour commander les Jobs affectés à chaque machine.</p> <p>Les nombres aléatoires sont utilisés seulement pour le premier étage. Ils déterminent la séquence de Job et l'affectation uniquement pour l'étage 1.</p> <p>Pour tous les étages successifs i, $\{i = 2, 3, \dots, m\}$ Faire</p> <ul style="list-style-type: none"> - la séquence de Job est déterminée par les temps les plus anciens d'achèvement des Jobs dans l'étage précédent - la règle d'affectation de la machine est la première machine disponible. <p>Fin pour</p> <p>Fin pour</p>

Pour notre problème, nous devons générer quatre nombres aléatoires d'une distribution uniforme entre $(1, 1 + m_I)$ pour le premier étage (Figure 6.2).

2,96	1,58	2,22	1,13
-------------	-------------	-------------	-------------

Figure 6.2 Solution codée en utilisant la représentation CA.

Il est bien connu que les solutions initiales peuvent fortement influencer sur les résultats finaux obtenus par les AG. Nous avons donc généré des solutions initiales comme suit: quatre solutions sont produites par les heuristiques de *PCDT*, *PLDT*, *règle de Johnson* ($m / 2, m / 2$) et *NEH_H*, et le reste est généré de façon aléatoire.

6.2.3.2 Stratégie d'élite

Les chromosomes à faible makespan sont les plus souhaitables et, par conséquent, un certain nombre des chromosomes (Nr) avec les valeurs les plus faibles de makespan sont automatiquement copiés vers la prochaine génération. Ce mécanisme est appelé la *reproduction ou stratégie d'élite*. Le reste des chromosomes $(TP - Nr)\%$ ou à leurs enfants sont produits par le croisement de deux autres séquences ou des parents par un opérateur appelé *opérateur de croisement*. Les opérateurs de croisement doivent éviter de générer des solutions infaisables.

6.2.3.3 Mécanisme de sélection

Pour la sélection des parents à subir de croisement, nous utilisons la sélection de classement qui pourrait être décrite comme suit:

Algorithme sélection de classement
- Les individus de la population actuelle sont d'abord triés en fonction de leurs fonctions objectives.
Pour Chaque individu Faire
Attribuer une probabilité normalisée de telle sorte que les meilleures solutions ont plus de chance d'être sélectionné.
Fin pour
- Les individus sont choisis au hasard en tant que parents de se soumettre à des opérateurs en fonction de leurs probabilités.

6.2.3.4 Croisement uniforme paramétré

Le but est de générer une meilleure génération, c'est-à-dire de créer de meilleures séquences en combinant les parents. Notre croisement est uniformément paramétré (*CUP*), car il a montré son efficacité dans *OFSH* dans des travaux antérieurs dans la littérature [153] et [155]. Il est nécessaire de préciser que les travaux de *CUP* par les clés aléatoires et définit comme suit:

Algorithme CUP	
Pour chaque Job	Faire
	Généré un nombre aléatoire entre (0, 1).
Fin pour	
Si la valeur est inférieure à 0,8	alors
	la CA du Job correspondant au parent 1 est copié à l'enfant,
sinon ,	la CA du parent 2 est sélectionné.
Fin si	
Les Jobs sont classés selon l'ordre ascendant des CA.	

La procédure est illustrée numériquement par son application à un exemple avec $n = 5$ et $m_1 = 2$ représentée sur la figure 6.3.

Parent 1	2,66	1,58	2,92	1,13	2,64
Parent 2	1,42	2,81	2,12	1,92	2,48
Random N°	0,62	0,34	0,97	0,12	0,89
Enfant	2,66	1,58	2,12	1,13	2,48

Figure 6.3 Procédure de croisement uniforme paramétré appliqué sur un exemple avec $n = 5$ et $m_1 = 2$.

6.2.3.5 Mutation ponctuelle unique

Un opérateur de mutation est utilisé pour modifier légèrement la séquence, c'est à dire générer une nouvelle séquence, mais semblable. Le but principal de l'application de la mutation est d'éviter la convergence vers un optimum local et de diversifier la population. L'opérateur de mutation peut aussi être vu comme une simple forme de la recherche locale.

De nombreux chercheurs ont conclu que seule la mutation ponctuelle unique, appelé *MPU*, peut fournir de meilleurs résultats que les autres mutations comme *SWAP* ou l'inversion.

Par conséquent, nous utilisons *MPU* que la mutation génétique [124] et [125]. La procédure de *MPU* peut être énoncée comme suit:

- la *CA* d'un Job choisie au hasard est aléatoirement régénéré. La figure 6.4 montre une solution illustrative qui subit une mutation.

Avant la Mutation	1,62	2,34	1,97	2,12	2,89
Après la Mutation	1,62	1,68	1,97	2,12	2,89

Figure 6.4 Procédure de mutation ponctuelle unique appliquée à un exemple avec $n = 5$ et $m1 = 2$

6.2.4 Conclusion

Nous avons présenté dans cette première partie, l'approche FSHMGA proposée. C'est une approche multi agent basée sur les AG, pour résoudre le problème d'intégration des politiques de maintenance systématique dans l'ordonnancement. Nous avons choisi les opérateurs suivants : la clé aléatoire pour le codage, la stratégie d'élite, la sélection de classement, le croisement uniforme et la mutation ponctuelle unique.

6.3 Approche FSHMAIA proposée

Dans l'approche proposée que nous avons baptisée FSHMAIA (*Flow shop Scheduling Hybrid Multi Agent Immune Artificiel*) pour notre problème, les antigènes se réfèrent à la fonction objective (makespan) qui doit être optimisé. Les anticorps se référant à des solutions candidates (similaire à des chromosomes en AG).

6.3.1 Organigramme de l'approche FSHMAIA proposée

La figure 6.5 montre un organigramme détaillé de notre approche FSHMAIA pour l'intégration des politiques de maintenance systématique dans l'ordonnancement.

Chaque agent de notre approche exécute le même processus, c'est le calcul de la fonction objective (Fitness).

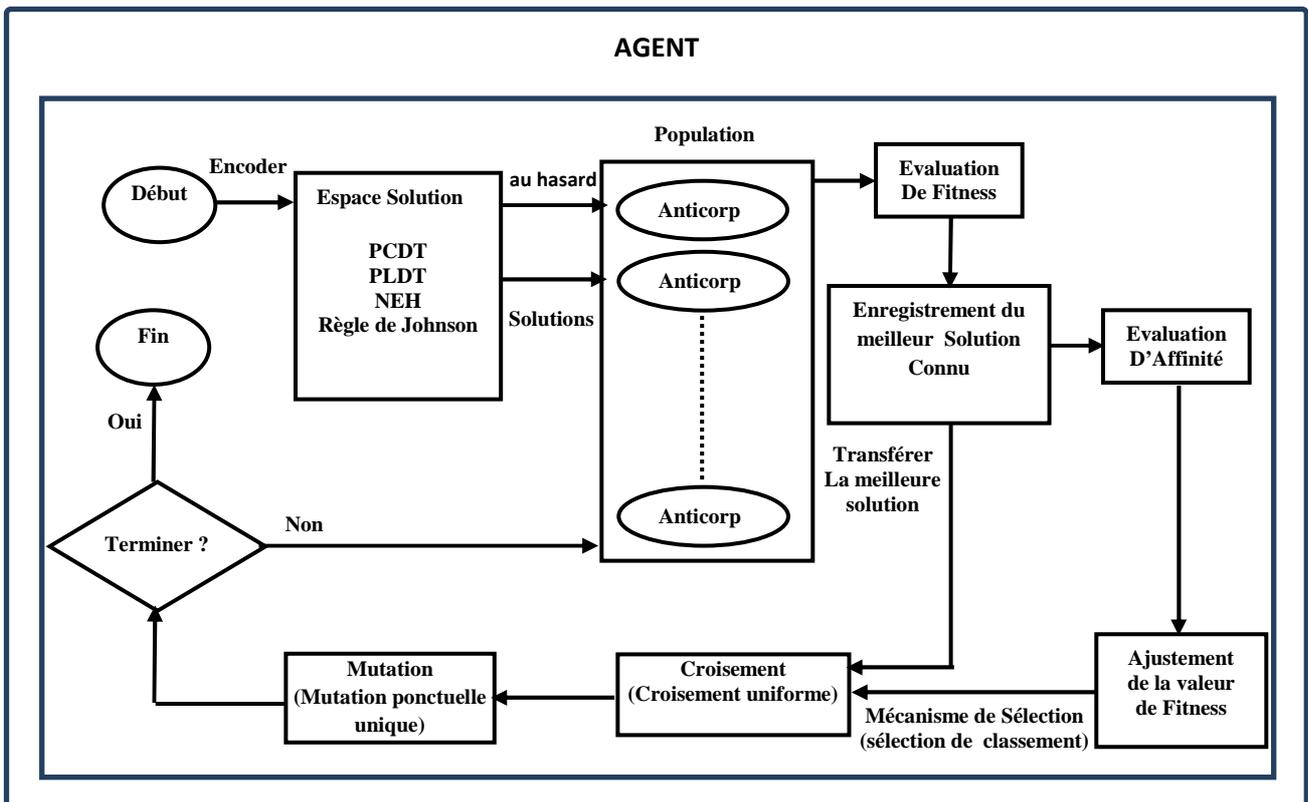


Figure 6.5 Organigramme de l'approche FSHMAIA proposée

Le SIA proposé dans cette approche a été construit à base de la **sélection clonale** et inspiré plus particulièrement de l'algorithme CLONALG proposé par [104]. Notre SIA basé aussi sur le principe de la maturation d'affinité du système immunitaire.

Les améliorations appliquées sur le SIA utilisé sont les suivantes :

- ✓ L'application de la *clé aléatoire* (CA) comme technique de codage de la population initiale. Comme en AG, le SIA commence à partir d'un certain nombre d'anticorps, appelé population. La population est améliorée par un jeu d'opérateurs jusqu'à ce qu'un critère d'arrêt soit atteint.
- ✓ Effectué un *mécanisme d'accélération* (comme la stratégie d'élite pour les AG) pour copie des parents dans la prochaine génération. Par le mécanisme d'accélération, des solutions candidates avec une meilleure aptitude sont transférés à la population suivante.
- ✓ Comme mécanisme de sélection, nous avons utilisé *la sélection de classement*. Pour sélectionner des anticorps à subir des opérateurs, une fonction de sélection qui utilise une valeur de bien de chaque anticorps (similaire à des valeurs de Fitness en AG).
- ✓ Proposer une nouvelle **procédure de calcul d'affinité** pour les anticorps. D'autre part, le calcul des affinités entre les anticorps est de supprimer les anticorps similaires. Il effectue les tâches suivantes: Si une solution candidate à une valeur d'affinité supérieure à une valeur de seuil prescrite (SA), il est assigné une probabilité plus faible en multipliant la probabilité préalablement obtenu à partir de la valeur de bien de l'anticorps avec un facteur inférieur à 1 (AR). Cela permettra de réduire la probabilité d'être sélectionné.
- ✓ Le choix du *croisement uniforme paramétré* pour l'opération de croisement du parent.
- ✓ Le choix de la *mutation ponctuelle unique* pour l'opération de mutation.
- ✓ Aussi notre apport réside essentiellement dans de l'hybridation du SIA amélioré avec un SMA.

6.3.2 Algorithme de l'approche FSHMAIA proposée

L'algorithme de l'approche FSHMAIA proposée se résume de la manière suivante :

Algorithme FSHMAIA
<p>Pour chaque Agent Faire</p> <p>Produire un ensemble d'anticorps (<i>TP</i>) comme une population initiale</p> <p>Tant que un critère d'arrêt n'est pas atteint faire</p> <p> Calculer des anticorps de bonne valeurs</p> <p> Calculer les valeurs d'affinité des anticorps</p> <p> Ajuster les valeurs finales de Fitness des anticorps</p> <p> Effectuer un mécanisme d'accélération // <i>Nr individus</i></p> <p> Pour $I = Nr + 1$ à <i>TP</i> faire</p> <p> Sélectionner deux parents en utilisant un mécanisme de sélection</p> <p> Effectuer le croisement uniforme et générer le <i>i</i>-ème descendant</p> <p> Si $rand < Pm$ alors// <i>rand est un nombre aléatoire uniformément répartie (0, 1)</i></p> <p> Effectuer une mutation ponctuelle unique sur le <i>i</i>-ème descendant</p> <p> Fin si</p> <p> Fin pour</p> <p> Fin tant que</p> <p> Fin pour</p> <p>Fin</p>

Chaque agent de notre approche exécute les opérations suivantes :

- ✓ Produire un ensemble d'anticorps (*TP*) comme une population initiale.
- ✓ Calculer des anticorps de bonnes valeurs.
- ✓ Calculer les valeurs d'affinité des anticorps
- ✓ Effectuer un mécanisme d'accélération
- ✓ Sélectionner deux parents en utilisant un mécanisme de sélection
- ✓ Effectuer le croisement uniforme
- ✓ Effectuer une mutation ponctuelle

Ce travail se répète jusqu'à l'arrivée à l'agent qui a la valeur de fitness la plus faible.

6.3.3 Schéma de codage et Opérateurs de notre approche FSHMAIA

6.3.3.1 Initialisation

- *Réglage de paramètres* : définir la taille de la population (TP), le nombre de solutions directement copié sur la population suivante (N_r), la probabilité de mutation (P_m), le poids des facteurs de similarité (Pd_1, Pd_2, Pd_3, Pd_4), le seuil d'affinité (SA), et l'affinité de réglage (AR).
- *Génération de la population initiale*:
 - ✓ Générer de façon aléatoire une population initiale de ($TP- 4$) d'anticorps.
 - ✓ Générer un anticorps avec $PCDT$.
 - ✓ Générer un anticorps avec $PLDT$.
 - ✓ Générer un anticorps avec *règle de Johnson* ($m/2, m/2$).
 - ✓ Générer un anticorps avec NEH_H .

6.3.3.2 Calcul de la fonction d'objective

En règle générale, une fonction d'objective (Fitness) comprend un ou plusieurs indicateurs de performance qui mesurent l'efficacité d'un anticorps.

Les anticorps candidats sont d'abord transformés dans un ordonnancement en cours de validité. À la fin, ils sont évalués en utilisant une fonction objective afin d'obtenir leurs valeurs d'adaptation. Pour un problème de maximisation, une valeur de fitness élevée est souhaitable et une tentative du SIA de le maximiser. Pour un problème de minimisation, la fonction objective est formulée de telle manière à la transformer en un problème de maximisation

Dans notre cas, le makespan doit être minimisé; une solution candidate avec un makespan haut est assigné a une valeur de Fitness faible.

Pour un anticorps i la fonction de fitness est calculée comme suit:

$$f(i) = \frac{\frac{1}{C_{max}(i)}}{\sum_{i=1}^{TP} \frac{1}{C_{max}(i)}}$$

Où : $f(i)$ est la valeur de fitness pour l'anticorps i , $C_{max}(i)$: est le makespan pour l'anticorps i , et TP : est la taille de population.

6.3.3.3 L'affinité

Selon les probabilités calculées à la fois par la valeur de bon et la valeur d'affinité, la sélection est effectuée en utilisant un mécanisme de sélection de classement.

Afin de mieux évaluer l'efficacité de la fonction de l'affinité du *SIA*, nous avons de nouveau appliqué les paramètres de croisement uniforme et la mutation ponctuelle unique.

La principale différence entre le *SIA* et *AG*, c'est que dans le *SIA*, la probabilité d'être sélectionné pour chaque anticorps dépend à la fois de ses performances et sa valeur d'affinité. Par contre au *AG*, cette probabilité est calculée uniquement par ses performances.

L'évaluation d'affinité du *SIA* augmente la diversité des anticorps dans une population, et, par conséquent, fournit l'occasion de visiter plus d'espace de recherche au détriment de son temps de calcul nécessaire.

La principale question est de savoir si dans tel problème complexe, le calcul d'affinité vaut son coût, c'est-à-dire la proportion du temps qu'il consomme de temps de calcul total de l'algorithme. Dans le paragraphe suivant, nous décrivons en détail la procédure de calcul d'affinité.

6.3.3.4 Algorithme de Calcul d'affinité

Pour calculer l'affinité, les anticorps sont comparés avec l'anticorps le plus connu (*APC*) obtenu jusqu'à présent. L'affinité exprime simplement la similarité entre un anticorps et l'*APC*.

La valeur d'affinité pour chaque anticorps dans notre problème est calculée en appliquant l'algorithme suivant:

Algorithme Calcul d'affinité
<p>Pour chaque Job <i>j</i> Faire</p> <p style="padding-left: 40px;">Calculer un rapport de similarité</p> <p>Fin pour</p>

- La moyenne des ratios de tous les jobs dans un anticorps est définie comme la valeur d'affinité globale de cet anticorps.
- Le rapport de similarité du job j est calculé en comparant la position de ce job en correspondant l'anticorps et APC .
- La séquence du Job et l'affectation pour les étages ultérieurs sont obtenues par les règles suivantes :
 - Le temps le plus anciens d'achèvement des Jobs à l'étage précédent et la première machine disponible, respectivement.
 - Le rapport de similarité est obtenu à partir de la position du Job j à l'étage 1.

Fin

Nous définissons les quatre critères suivants pour calculer le ratio de similarité du job j :

Algorithme ratio de similarité

(1)**Si** le Job j est dans les deux anticorps (soit anticorps candidat et APC) **Alors**
Le job j traitée par la même machine.

Fin si

(2)**Si** le facteur 1 est oui, **Alors**
le Job j est dans la même position dans la machine correspondante.

Fin si

(3)**Si** le précédent du Job j est dans les deux anticorps **Alors**
est le même Job.

Fin si

(4)**Si** le successeur du Job j est dans les deux anticorps **Alors**
est le même Job.

Fin si

Chaque critère a un poids qui montre son importance relative (notée par Pd_i , $i = \{1, 2, 3, 4\}$). Si un critère est satisfait, le Job j reçoit son poids, et la valeur totale des

pois reçus est le rapport de similarité du Job j . Après le calcul de similarité de chaque Job, les similarités moyennes des Jobs sont utilisées comme l'affinité de l'anticorps candidat.

6.3.3.5 Exemple d'application de l'algorithme de Calcul d'affinité

Nous allons illustrer l'algorithme en l'appliquant sur un exemple avec $n = 5$ et $m_1 = 2$. Supposons que des solutions décodées obtenues par l'anticorps candidat et APC sont présentés dans la figure 6.6 et la figure 6.7. Encore une fois considérer que $Pd_1 = 0,3$, $Pd_2 = 0,1$, $Pd_3=0,3$ et $Pd_4 = 0,3$. Dans ce cas, le Job 1 satisfait aux critères 1 et 4, par conséquent, son ratio de similarité résume à 0,6. Le Job 2 remplit les critères 1, 2 et 3, de même, son ratio de similarité devient 0,7. De même, le Job3, 4 et 5 ont des ratios de similarité de 0,6, 0 et 0, respectivement. Enfin, la valeur d'affinité de cette solution candidate est de 0,38.

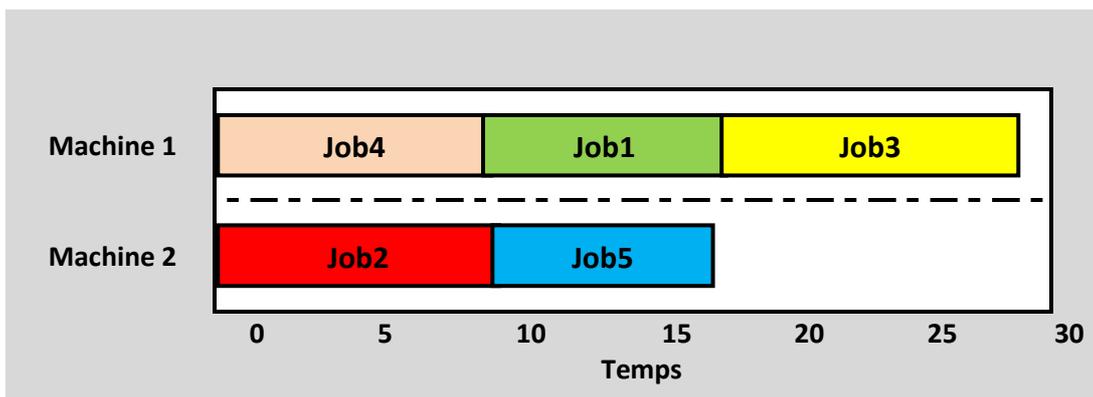


Figure 6.6 Diagramme de Gantt de l'anticorps candidat au premier étage.

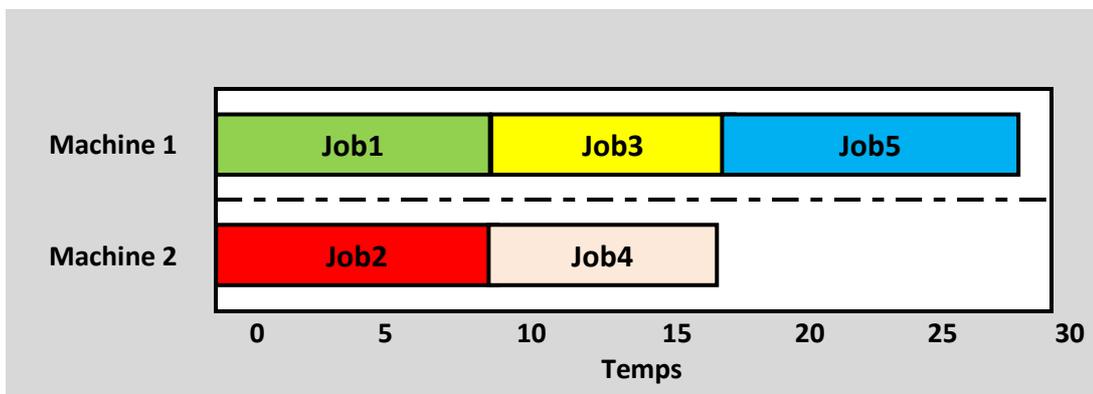


Figure 6.7 Diagramme de Gantt de l'anticorps le plus connu au premier étage.

6.3.4 Conclusion

Dans cette deuxième partie, nous avons présenté la conception de la deuxième approche FSHMAIA proposée. C'est une approche multi agent basée sur les SIA.

Nous avons utilisée les mêmes opérateurs que l'approche FSHMAGA, en plus nous avons proposé un nouveau algorithme de calcul d'affinité.

Dans la prochaine section nous allons faire des comparaisons entre nos deux approche FSHMAIA et FSHMAGA avec d'autre heuristiques tels que NEH_H , règle de Johnson, PCDT et PLDT.

6.4 Résultats et discussion

Dans cette section, nous évaluons nos approches multi agent basé sur les méthodes d'émergence proposé.

Notre but est de faire des comparaisons de nos approches avec les heuristiques, *NEH*, règle de *Johnson* ($m/2, m/2$), *PCDT* et *PLDT*.

Nous avons implémenté ses heuristiques en *MATLAB7.0* qui fonctionne sur un *PC Intel Core Duo 2* de $2,0\text{ GHz}$ et une mémoire *RAM* de 2 Go . Le critère d'arrêt utilisé lors de l'essai avec toutes les instances des heuristiques est réglée à une limite de temps du *CPU* fixé à $n^2 \times m \times 1,5\text{ ms}$. Ce critère d'arrêt permet non seulement de plus de temps que le nombre de Jobs ou des augmentations des étages, mais est également plus sensible à une hausse du nombre de Jobs que le nombre d'étages.

Nous utilisons le pourcentage relatif de l'écart (*PRE*) en tant que mesure de la performance commune pour comparer les méthodes. La bonne solution obtenue pour chaque instance (nommé Min_{sol}) est calculée par l'un des algorithmes. *PRE* est obtenue par la formule suivante:

$$PRE = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100$$

Où Alg_{sol} est la valeur de la fonction objective obtenue pour un algorithme donné pour une instance. De toute évidence, des valeurs plus faibles de *PRE* sont préférables.

6.4.1 Réglage des paramètres

Il est connu que les différents niveaux des paramètres affectent clairement sur la qualité des solutions obtenues par nos deux approches multi agent basé sur *SIA* (*FSHMAIA*) et *AG* (*FSHMAGA*). Un certain nombre de différence entre *FSAMAIA* et *FSHMAGA* peut être obtenu avec les différentes combinaisons des paramètres. Nous avons appliqué un réglage de paramètres sur la taille de la population (*TP*), le nombre de solutions directement copié sur la population suivante (N_r), la probabilité de mutation (P_m), le poids des facteurs de similarité (Pd_1, Pd_2, Pd_3, Pd_4), le seuil d'affinité (*SA*), et l'affinité de réglage (*AR*). Le tableau 6.1 et le tableau 6.2 montrent les niveaux considérés.

Tableau 6.1 Niveaux de paramètres du FSHMAIA.

Paramètres	N° du Niveau	Niveaux
Nombre d'agent	3	10,20,30
Taille Population	3	50, 100, 150
(N_r, P_m)	3	(10, 0.10), (20, 0.15), (30, 0.20)
(Pd_1, Pd_2, Pd_3, Pd_4)	2	(0.20, 0.10, 0.35, 0.35), (0.30, 0.20, 0.25, 0.25)
(SA, AR)	3	(0.40, 0.50), (0.60, 0.50), (0.80, 0.50)

Tableau 6.2 Niveaux de paramètres de FSHMAGA.

Paramètres	N° du Niveau	Niveaux
Nombre d'agent	3	10,20,30
Taille Population	3	50, 100, 150
(N_r, P_m)	3	(10, 0.10), (20, 0.15), (30, 0.20)

Un ensemble de 30 instances en 3 groupes ($n = 40, 70, 100$) est généré et résolu par les algorithmes. Après avoir analysé les résultats pour FSHMAGA, nous choisissons : $NAG=20$, $N_r = 10$, $P_m = 0,15$ et $TP = 100$, et pour le FSHMAIA, les résultats indiquent que les meilleures valeurs pour : $NAG=20$, $N_r = 10$, $P_m = 0,15$ et $TP = 100$, $Pd_1 = 0,3$, $Pd_2 = 0,2$, $Pd_3 = 0,25$ et $Pd_4 = 0,25$, $SA = 0,6$ et $AR = 0,5$. Autrement dit, si la valeur d'affinité devient supérieure à 0,6, la probabilité d'être sélectionné par un mécanisme de classement sélectionné est multiplié par 0,5.

6.4.2 Génération de données

Les données nécessaires pour résoudre le problème d'intégration des politiques de maintenance systématique dans l'ordonnancement flow shop hybride (OFSH) consistent en deux parties, les données relatives à l'ordonnancement de la production et des données relatives à la maintenance systématique. Il est nécessaire de traiter les données de façon à veiller à ce qu'un nombre important d'opérations de la MP sont effectuée sur chaque machine. Si le temps entre deux opérations consécutives de la MP est inférieur à la durée de traitement maximale, certains Jobs ne pourraient être jamais traités. D'autre part, si ce temps devient très grand, il est très probable qu'aucune opération de la MP n'est nécessaire.

La première partie des données inclut le nombre de Job (n), nombre d'étages (m), le nombre de machines identiques à chaque étage (m_i), l'intervalle de temps de traitement ($P_{j,i}$) et des temps de prêts. On $n = \{40, 70, 100\}$, et $m = \{2, 4, 8\}$. Pour définir le nombre de machines à chaque étage, nous avons deux ensembles. Dans le premier, nous avons un certain nombre de distribution aléatoire uniforme de machines comprises entre un et trois machines par étage, et dans le second, nous avons un nombre fixe de deux machines par étage. Les temps de prêts pour l'étage 1 sont mis à 0 pour tous les Jobs. Les temps de prêts à l'étage ($i + 1$) sont les temps d'exécution à l'étage i , de sorte ces données ne doivent pas être générées. Le tableau 6.3 montre les facteurs et leurs niveaux.

Tableau 6.3 Facteurs et leurs niveaux.

Facteurs	Niveaux
Nombre de Jobs	40, 70, 100
Nombre des étages	2, 4, 8
Distribution d'une machine	a. Constant : 2 b. Variable : U (1, 3)
Temps de traitement	U (1, 99)

La deuxième partie des données est divisé en trois sous-parties dont chacune considère une politique. Comme mentionné précédemment, la génération T_{MPF} , T_{MPop} et T_{MP} doit être faite avec le plus grand soin. Pour cela, nous avons besoin de définir une variable artificielle " x_i " pour estimer la charge de travail sur les machines de chaque étage i comme suit:

$$x_i \approx \frac{n}{m_i}$$

De sorte que « x_i » est le nombre prévu des Jobs sur chaque machine à l'étage i . Par conséquent, la plage de cette variable serait la suivante:

$$x_i = \{10, 13.3, 17.5, 20, 23.3, 25, 33.3, 35, 40, 50, 70, 100\}$$

Par exemple, dans le cas de $n = 70$, $g = 2$, $m_1=2$ et $m_2= 3$, nous obtiendrons $x_1= 35$ et $x_2= 23,3$. Les autres données requises pour chaque politique sont générée comme suit:

1. Les données pour la politique I de la *MP*: T_{MPF} sont déterminées en fonction de x_i . Si $x_i < 25$, alors $T_{MPF}= 450$; autrement, $T_{MPF} = 650$. La durée de l'opération *MP* (D_{MP}) est définie à 50%, 100% et 150% du temps de traitement.
2. Les données pour la politique II de la *MP*: Comme mentionné précédemment, il ya 9 combinaisons de n et g . Pour chaque combinaison, $\beta = \{2, 3, 4\}$ est définie. D_{MP} est le même que la politique I. Dans cette politique, t_p est fixé à 1 et t_r à 8 pour toutes les expériences. Les valeurs de θ sont fixées en fonction de la variable x_i . Les niveaux de θ sont choisis de manière à s'assurer qu'un nombre important d'opérations *MP* serait effectué dans chaque machine. Par exemple, il faut noter qu'une valeur petite pour θ se traduirait par une valeur très grande de T_{MPop} alors une valeur très grande serait probablement empêchée la réalisation de certains traitements de Jobs sur des machines sans interruptions dues à la faible quantité de T_{MPop} . Les valeurs de θ sont présentées dans le Tableau 6.4.

Tableau 6.4 Valeurs de θ correspondant à la variable x_i .

x_i	10	13.3	17.5	20	23.3	25	33.3	35	40	50	70	100
θ	1400	1600	1850	1950	2100	2200	2550	2600	2750	3100	3650	4350

3. Les données pour la politique III de la *MP*: les niveaux de θ , β et D_{MP} sont les mêmes que la politique II. L'objectif est une fiabilité de 95% après la période t de production, donc $R_0(t) = 0,95$. Afin de calculer T_{MP} , il est nécessaire de déterminer la période t , qui peut être facilement obtenus à partir des temps de traitement d'une instance donnée. Étant donné que les délais de traitement sont répartis uniformément sur $(1, 99)$, $t = x_i \cdot 50$.

Tous les résultats des différents niveaux de facteurs cité dans 54, 162 et 162 scénarios pour les politiques I, II et III, respectivement. Pour chaque scénario, il ya 10 différents problèmes qui en résultent au total 540, 1620, 1620 instances.

6.5 Résultats expérimentaux

Les résultats des expériences, en moyenne pour chaque combinaison de n et m (180 données par la moyenne) dans les trois sous-ensembles (Politiques I, II et III) sont regroupés dans les tableaux 6.5, 6.6 et 6.7. Comme prévu, les deux approches proposées fonctionnent mieux que les autres heuristiques dans les trois politiques. *FSHMAIA* proposé fournit les meilleurs résultats que *FSHMAGA* dans les trois politiques avec un *PRE* de 1,81%, 1,74% et 1,67% respectivement, tandis que *FSHMAGA* obtient un *PRE* de 3,49%, 3,19% et 3,63% dans les politiques I, II et III, respectivement.

Les résultats du *PRE* du *FSHMAIA* sont les plus bas dans tous les 9 groupes (des combinaisons de n et m) ainsi qu'elle maintient sa robustesse dans les trois politiques de la *MP*.

Tableau 6.5 Valeurs de *PRE* des algorithmes pour la politique I

n	m	Algorithmes					
		<i>FSHMAIA</i>	<i>FSHMAGA</i>	<i>NEH_H</i>	<i>John</i>	<i>PCDT</i>	<i>PLDT</i>
40	2	1.95	3.36	8.69	20.16	24.49	30.48
	4	1.94	3.64	9.96	19.98	30.63	32.01
	8	2.37	4.37	7.75	22.13	28.59	30.75
70	2	1.93	4.40	8.29	25.17	31.11	27.26
	4	1.79	2.66	9.80	18.03	25.12	30.91
	8	1.38	3.87	7.11	19.02	21.30	28.06
100	2	1.20	2.47	5.68	23.75	31.17	31.32
	4	1.67	3.11	4.57	24.48	24.08	26.79
	8	2.06	3.52	4.91	20.69	27.18	25.52
PRE moyenne		1.81	3.49	7.42	21.49	27.08	29.23

Tableau 6.6 Valeurs de PRE des algorithmes pour la politique II

<i>n</i>	<i>m</i>	<i>Algorithmes</i>					
		<i>FSHMAIA</i>	<i>FSHMAGA</i>	<i>NEH_H</i>	<i>John</i>	<i>PCDT</i>	<i>PLDT</i>
40	2	1.04	2.79	9.43	16.83	28.37	28.94
	4	2.08	3.51	9.81	18.82	27.89	31.87
	8	2.33	3.20	7.57	18.66	24.28	25.75
70	2	1.73	4.19	8.98	19.61	28.43	28.96
	4	1.56	3.11	9.12	18.71	25.17	29.60
	8	1.51	3.21	8.62	16.98	28.26	31.52
100	2	1.27	2.23	6.25	19.98	26.05	31.97
	4	2.33	3.02	5.31	16.64	27.99	27.53
	8	1.82	3.46	5.40	16.54	26.45	26.41
PRE moyenne		1.74	3.19	7.83	18.09	26.99	29.17

Tableau 6.7 Valeurs de *PRE* des algorithmes pour la politique III

<i>n</i>	<i>m</i>	<i>Algorithmes</i>					
		<i>FSHMAIA</i>	<i>FSHMAGA</i>	<i>NEH_H</i>	<i>John</i>	<i>PCDT</i>	<i>PLDT</i>
40	2	1.38	2.32	8.74	16.06	26.99	25.94
	4	2.18	3.99	8.36	17.93	24.01	29.57
	8	1.21	3.29	7.37	19.70	28.69	28.14
70	2	1.40	3.23	9.59	17.44	27.18	32.06
	4	1.11	4.67	9.88	18.56	29.80	29.59
	8	2.01	2.49	8.40	21.90	27.32	32.74
100	2	1.38	4.40	5.34	18.89	30.13	30.49
	4	2.18	4.94	5.36	20.48	27.44	28.42
	8	2.16	3.34	4.13	18.72	31.70	30.51
PRE Moyenne		1.67	3.63	7.46	18.85	28.14	29.72

Comme on le voit, avec *NEH_H*, on obtient des résultats remarquablement mieux que les autres heuristiques avec *PRE* de 7,42%, 7,83% et 7,46% dans les politiques I, II et III, respectivement, Après *NEH_H*, l'algorithme de *Johnson* Obtient des valeurs de *PRE* égale à 21,49%, 18,09% et 18,85% dans les politiques I, II et III, respectivement. Les pires des algorithmes les moins performants sont *PCDT* et *PLDT* avec près de 30% de *PRE*.

Pour une analyse précise des résultats, nous procédons à une analyse de variance (ANOVA). L'analyse de la variance permet d'étudier le comportement d'une variable continue à expliquer en fonction d'une ou de plusieurs variables explicatives catégorielles. Dans notre cas, ce sont les valeurs de *PRE* que ne devons étudier par rapport aux facteurs ; nombre de job, le nombre d'étages et le type de la politique *MP*

La figure 6.8 montre le graphe de meilleures valeurs de PRE. Comme la montre la Figure 6.8, l'approche *FSHMAIA* proposée donne le bon résultat par rapport aux autres algorithmes.

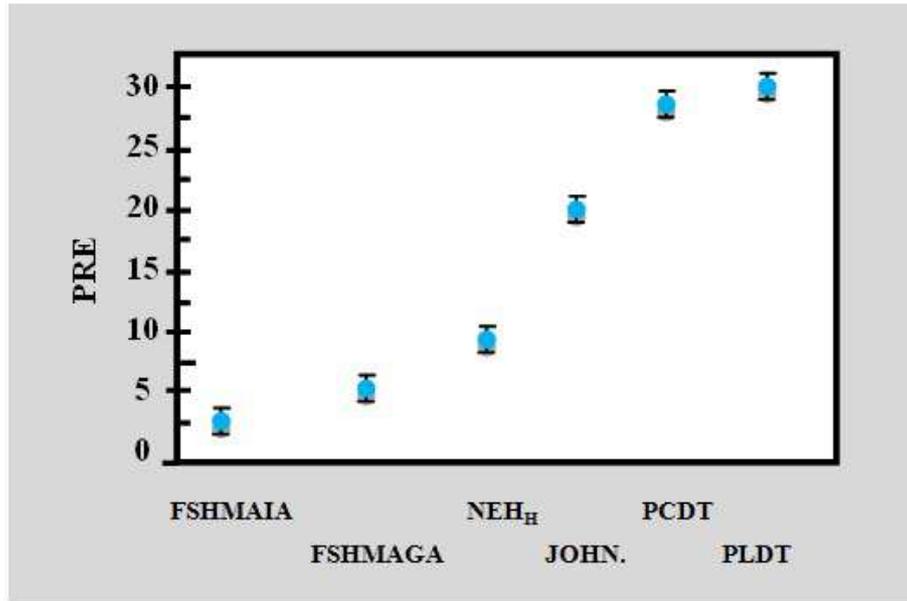


Figure 6.8. Graphe de meilleures valeurs de PRE entre les différents algorithmes

Nous analysons les interactions entre les facteurs tels que par exemple le nombre de job, le nombre d'étages et le type de la politique *MP* sur la performance des algorithmes. À la fin, nous traçons les valeurs de *PRE* obtenus par les algorithmes par rapport aux différents niveaux des facteurs.

En raison de la performance nettement pire de la *PCDT*, *PLDT* et *Johnson* ($m/2$, $m/2$), nous les excluons de l'expérience.

Comme le montre la figure 6.9 L'approche *FSHMAIA* fournit la plus faible *PRE* dans les trois niveaux du nombre de Jobs, et surtout pour un nombre de job égale à 70. Lorsqu'on augmente le nombre de jobs, *NEH_H* est plus performant, et donne de bons résultats. Les valeurs *FSHMAGA* reste presque stable dans les trois niveaux.

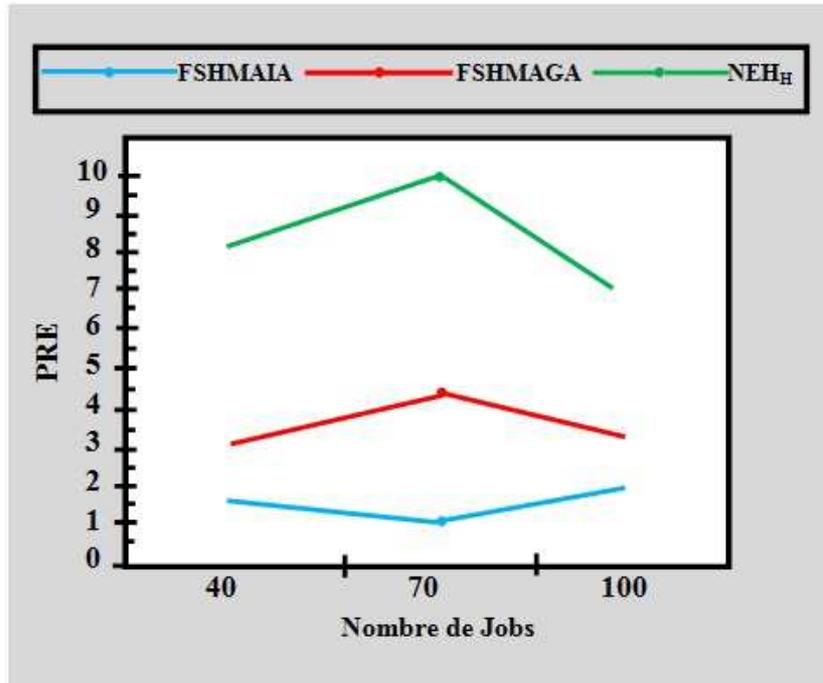


Figure 6.9 Graphe des valeurs de PRE dans le cas du facteur nombre de jobs

La figure 6.10 montre qu'un nombre croissant des étages résulte des meilleures performances pour NEH_H . Les meilleures valeurs de PRE fournies toujours par FSHMAIA

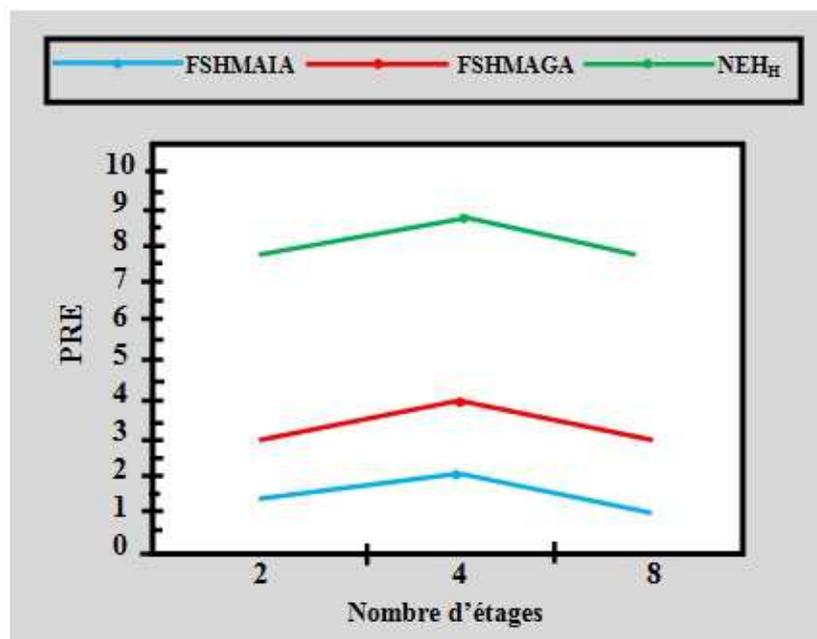


Figure 6.10 Graphe des valeurs de PRE pour le facteur nombre d'étages.

Il n'y a pas d'interaction entre les performances des algorithmes et le facteur de politiques de la *MP*. Dans tous les cas, le *FSHMAIA* donne les meilleurs résultats par rapport aux autres algorithmes (figure 6.11).

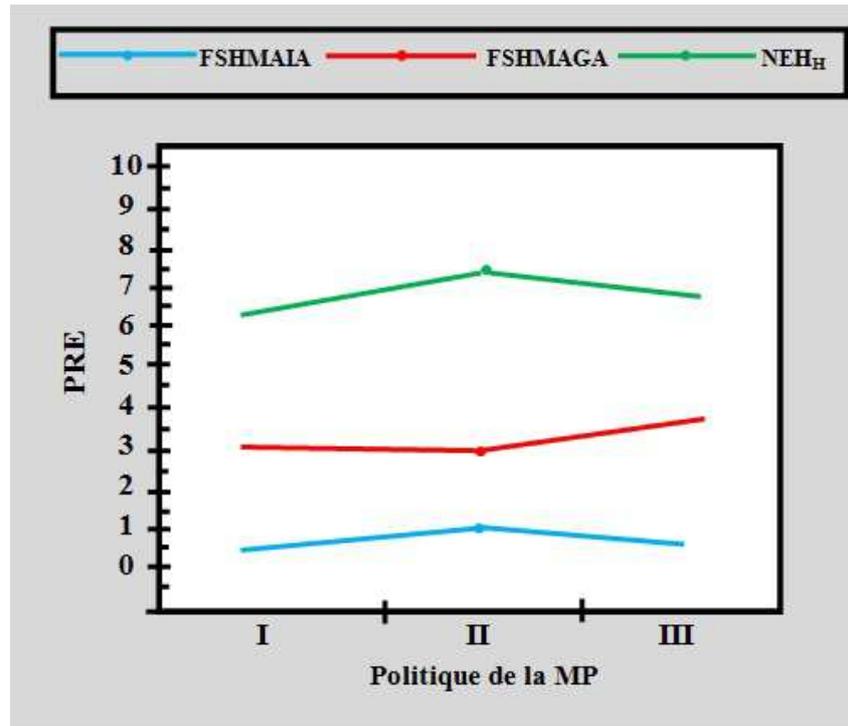


Figure 6.11 Graphe des valeurs de PRE pour le facteur politique de la MP.

6.6 Conclusion

Dans ce chapitre nous avons présenté notre travail qui est l'intégration des politiques de maintenance préventive systématique dans les ateliers flow shop hybrides afin de minimiser le makespan.

Pour résoudre un tel problème complexe, nous avons proposé deux approches multi agent basées sur deux méthodes d'émergence, les AG et les SIA dans lequel nous avons utilisé des opérateurs avancés tels que croisement uniforme paramétré et la mutation ponctuelle unique. Par ailleurs, nous avons défini une nouvelle procédure de calcul d'affinité dans le cadre du *FSHMAIA*.

Nous avons également évalué les adaptations de certaines heuristiques bien connues, y compris *PCDT*, *PLDT*, règle *Johnson (m/2, m/2)* et *NEH_H*.

Un benchmark a été établi avec un grand soin pour évaluer les algorithmes. Le benchmark contenu jusqu'à 100 jobs et 8 étages. Tous les résultats ont montré que le *FSHMAIA* donne des résultats satisfaisants par rapport aux autres algorithmes, outre sa robustesse dans les trois politiques de la *MP*.

Conclusion générale

Conclusion générale

Dans le cadre de ces travaux de recherche, nous avons abordé la résolution de l'un des problèmes de la maintenance industrielle. Nous avons mis en place deux approches multi agent basée sur des méthodes d'émergence, les AG et les SIA, pour la résolution de problème d'optimisation du temps de traitement.

L'objectif principal de notre travail est l'intégration des politiques de maintenance préventive systématique dans l'ordonnancement.

Cette thèse commence par dresser un état de l'art sur notre domaine de recherche, à savoir les SMA, les méthodes d'émergence et la maintenance systématique. Nous avons présenté quelques études réalisées avec les SMA dans le domaine industriel. Nous avons cité ensuite des travaux de recherche réalisées avec les méthodes d'émergences dans le domaine industriel. Peu de travaux réalisés dans le domaine de la maintenance systématique, et plus particulièrement, dans l'intégration de leurs politiques dans un ordonnancement de type flow shop hybride.

Nous avons ensuite, entamé les SMA. Nous avons étudié les concepts essentiels d'agents, les distinctes explications, spécifiques, différentes architectures et les principaux modèles d'agents. Ensuite nous avons vu les systèmes multi-agents (SMA), leur émergence de l'IAD et leurs champs d'applications. Nous avons abordé également les notions d'agents au sein des sociétés, les nouvelles contraintes et les nouvelles fonctionnalités issues d'une coopération entre les différents agents constituant la société. Nous avons vu l'intérêt des SMA, les points forts de ces systèmes et les nouveautés apportées par ces derniers.

Ensuite, nous avons abordé l'importance des notions d'émergence et d'auto-organisation, leurs définitions et fondements adéquats en mettant l'accent sur ce qui assemble, mais surtout ce qui distingue les deux notions. L'utilisation de l'émergence et de l'auto-organisation concerne d'ailleurs un certain nombre de méthodes métaheuristiques, comme notamment, le calcul évolutionnaire, les réseaux neuronaux,

les algorithmes de fourmis et l'optimisation par essaim particulaires. Ces techniques sont présentées et discutées avec un regard critique concernant la notion d'émergence.

Nous nous sommes intéressés par la suite à découvrir les notions indispensables des systèmes immunitaires naturels et les fondements théoriques sur lesquelles est fondée la conduite de ces derniers. Ensuite nous avons intéressé aux *SIA* qui sont l'application des systèmes immunitaires naturels pour la résolution de problèmes complexes.

Nous avons mis l'accent la maintenance préventive systématique, en présentant ses politiques. Nous avons vu aussi l'ordonnancement, et plus particulièrement l'ordonnancement d'un atelier flow shop hybride. Nous avons vu aussi, quelques méthodes de résolution du problème d'ordonnancement flow shop hybride.

A la fin, nous avons proposé deux approches pour la résolution du problème d'optimisation du Makespan lors de l'intégration des politiques de maintenance systématique dans l'ordonnancement d'un atelier flow shop hybride.

C'est une hybridation entre les SMA et des inspirations du corps humain et plus particulièrement de la génétique et du système immunitaire humain. Nous avons supposé que les machines pourraient être périodiquement indisponibles pendant l'ordonnancement de la production.

Les deux approches proposées sont appliquée sur trois politiques de maintenance préventives. Ces politiques sont destinées à maximiser la disponibilité ou à maintenir un niveau minimum de fiabilité au cours de la chaîne de production.

Les critères que nous avons appliqués sont très simples mais efficace. En plus, ils sont adaptables à tous les problèmes d'ordonnancement.

Pour résoudre un tel problème complexe, nous avons proposé deux approches multi agent basées sur deux méthodes d'émergence, les *FSHMAGA* et le *FSHMAIA*.

Nous avons utilisé des opérateurs avancés ; la clé aléatoire pour le codage, la sélection de classement, le croisement uniforme et la mutation ponctuelle unique.

Par ailleurs, nous avons défini une nouvelle procédure de calcul d'affinité dans le cadre du *FSHMAIA*. Cette procédure est basée sur le calcul d'un rapport de similarité

des anticorps. Pour cela, nous avons proposé un algorithme basé sur un critère pour calculer le rapport de similarité.

Nous avons également évalué les adaptations de certaines heuristiques bien connues, y compris *Johnson (m / 2, m / 2)*, *NEH_H*, *PCDT* et *PLDT*,

Un benchmark a été établi avec un grand soin pour évaluer les algorithmes. Le benchmark contenu jusqu'à 100 jobs et 8 étages. Tous les résultats ont montré que le *FSHMAIA* donne les meilleurs résultats par rapport aux autres algorithmes, dans les trois politiques de la *MP*.

Comme une recherche intéressante pour l'avenir, on pourrait :

- ✓ Faire des comparaisons entre les approches proposées et d'autres méthodes telles que les colonies de fourmis et l'optimisation par essaim de particules.
- ✓ Tester ces deux approches sur d'autres benchmarks basés sur des données réelles issues des systèmes industriels.
- ✓ Étudier l'intégration des politiques de maintenance dans d'autres problèmes d'ordonnement tels que les ateliers à cheminement libre (Open shop).

Références Bibliographiques

- [01] A.KELLY, M. J. HARRIS, "Management of Industrial Maintenance", London, Butter worth's Management Library, 1978
- [02] S.NAKAJIMA, "Total Productive Maintenance. Introduction to TPM", Cambridge, Productivity Press, 1988.
- [03] A. ABOU, "contribution a la mise en œuvre d'une maintenance centralisée : conception et optimisation d'un atelier de maintenance ", Thèse de doctorat, Université de Joseph Fourier - GRENOBLE 1, Octobre 2003.
- [04] V.ZILLE, "Modélisation et évaluation des stratégies de maintenance complexes sur des systèmes multi-composants", Thèse de doctorat, l'Université de Technologie de Troyes, 2009
- [05] F. MONCHY, "Maintenance - Méthodes et organisation", coll. « Technique et ingénierie », Dunod, p513, 2003.
- [06] R. C. BALDWIN, "Enabling an E-maintenance infrastructure", 2004.
- [07] A. KELLY, "Maintenance and its Management", Conference Communication, Farnham, England, 1989.
- [08] Norme AFNOR, "Terminologie de la maintenance ", NF-EN 13306. X60-319, p57, Juin 2001.
- [09] S.NAKAJIMA, "TPM Development Program: Implementing Total Productive Maintenance", Cambridge, Productivity Press, 1989.
- [10] R.C. BALDWIN, "How do you spell e-maintenance?", 2004.
- [11] Norme AFNOR, "Statistique et Qualité, introduction à la fiabilité", X NF, 06-501, Novembre 1977.
- [12] J.P.SOURIS, "La maintenance source de profits", Paris, Les Éditions d'Organisation, 1990.
- [13] G.ZWINGELSTEIN, "La maintenance basée sur la fiabilité". Hermès, 1996.
- [14] A.DESPUJOLS, "Optimisation de la maintenance par la fiabilité". Techniques de l'ingénieur, dossier MT9310, 2004.
- [15] I.GERTSBAKH, "Models of Preventive Maintenance". North-Holland, Amsterdam, 1977.

- [16] M.GLADE, "Modélisation des couts de cycle de vie : prévision des couts de maintenance et de la fiabilité Application a l'aéronautique", Thèse de doctorat, Ecole centrale de Lyon, 2007.
- [17] C.GREGOIRE, "Modélisation et simulation de stratégies de maintenance d'un système de graissage a l'aide de réseaux de Pétri". Rapport de projet de fin d'études d'ingénieur de l'Université de Technologie de Troyes, EDF R&D Chatou, Printemps 2004.
- [18] P. LYONNET, "Optimisation d'une politique de maintenance". Edite par TEC ET DOC, 1999.
- [19] A. MULLER, "Contribution a la maintenance prévisionnelle des systèmes de production par la formalisation d'un processus de pronostic", Thèse de doctorat, Université Henri Poincaré - Nancy 1, 2005.
- [20] S.OZEKICI, "Reliability and maintenance of complex systems", NATO ASI Series –series F / Computer and system safety, vol. 15, Springer, 1996.
- [21] E. ADAM, "Systèmes multi-agents : éléments introductifs", Université de Valenciennes et du Hainaut-Cambrésis France, 2008.
- [22] G. CLAIR, E. KADDOUM, M. P. GLEIZES, and G. PICARD, "Approches multi-agents auto-organisatrices pour un contrôle manufacturier intelligent et adaptatif", In Cépaduès, editor, Journées Francophones sur les Systèmes Multi-Agents, pp. 191–200, 2008.
- [23] C. BAEIJS, "Fonctionnalité Emergente dans une Société d'Agents Autonomes". Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1998.
- [24] J. FERBER, "Les systèmes multi-agents vers une intelligence collective", Inter Editions, 1995.
- [25] M. H. STEFANINI, "TALISMAN : une architecture Multi-Agents pour l'analyse du français écrit", Thèse de Doctorat en Informatique, Université Grenoble II, 1993.
- [26] J. L. KONING, "Des règles d'interaction comme modèle de contrôle en univers multi agent", Habilitation à diriger les recherches de l'INP. Grenoble. Laboratoire Leibniz. Grenoble. 2001.
- [27] J. BORDINI, "Multi-Agent Programming, Languages, Platforms and Applications", ISBN-10: 0-387-24568-5,2005.
- [28] J.P. BRIOT, Y. DEMAZEAU, "Introduction aux agents : Principes et architecture des systèmes multi-agents", Collection IC2, Hermès, 2001.
- [29] M.DASTANI, A. E. SGHROUCHNI, "Programming Multi Agents Systems", First International Workshop, ProMAS, 2003.
- [30] Y. DEMAZEAU, "La plate forme PACO et ses applications", 2^{ème} journée nationale du PRC-IA sur les systèmes multi-agents, Montpellier, 1993.
- [31] R. COURDIER, "Systèmes multi-agents : Intelligence artificielle et intelligence collective",Hermès, 2005.

- [32] S. TANDABANY, "Méthode de programmation orienté Agent", Hermès, 2005.
- [33] W. SHEN, D.H. NORRIE, "Agent-based systems for intelligent manufacturing: a state-of-the-art survey", KAIS, vol. 1, n. 2, pp. 129–156, 1999.
- [34] P.M. RICORDEL, "Programmation Orientée Multi-Agents - Développement et Déploiement de Systèmes Multi-Agents Voyelles", Institut National Polytechnique de Grenoble, 2001.
- [35] W. SHEN, Q. HAO, "Applications of agent based systems in intelligent manufacturing: An updated review", Advanced Engineering Informatics, vol. 20, pp. 415-431, 2006.
- [36] A. DROGOUL, "Systèmes multi-agents", Université de Paris-6, 2005
- [37] J. DREO, P. SIARRY, "Métaheuristiques pour l'optimisation et auto organisation dans les systèmes biologiques", Université de Paris XII Val-de-Marne, Laboratoire d'Etude et de Recherche en Instrumentation, 2003.
- [38] W.SHEN, D.H. NORRIE, J.P.BARTHÈS, "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing", Taylor and Francis, London, UK, 2001.
- [39] H. V. D. PARUNAK, S. BRUECKNER, M. FLEISCHER, and J.ODELL, "A design taxonomy of multi-agent interactions". Lecture Notes in Computer Science, vol. 4, pp123–137, 2003.
- [40] S.TOUAF,"Diagnostic logique des systèmes complexes dynamiques dans un contexte multi-agent", Thèse de doctorat, Université de Joseph Fourier - GRENOBLE 1, Mars 2005.
- [41] J. DEGUET, "Intégration de l'émergence au sein des systèmes multi-agent, une étude appliquée à la recherche heuristique", Thèse de doctorat, Université de Joseph Fourier - GRENOBLE 1, Mai 2008.
- [42] P.GAUTHIER, "Méthodologie de développement de systèmes multi-agents adaptatifs
Conception de logiciels à fonctionnalité émergente". Thèse de doctorat, Université Paul Sabatier - Toulouse, Décembre 2004.
- [43] J. GABER, M. BAKHOUYA, "Approche Auto-adaptative Inspirée du Système Immunitaire pour la Mise en Œuvre de l'Ubiquité", In: 27th Annual IEEE Conference on Local Computer Networks (IEEE LCN02) 230-240, 2004.
- [44] P. MERZ, "Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies". PhD thesis, Universität-Gesamthochschule Siegem, 2000.
- [45] J. C. CRÉPUT, A. KOUKAM, "The memetic self-organizing map approach to the vehicle routing problem. Soft computing applications in industry", vol. 12, pp. 189–205, 2008.
- [46] T. COUDERT, B. ARCHIMED, B. GRABOT, "un système multi-agents pour la coopération production/maintenance", Journal of Decision Systems, VOL 13, pp.27-62 -2004.
- [47] Y. HARRATH, "Algorithmes génétiques et fouille de données pour un ordonnancement réactif dans un atelier de type job-shop", Thèse de doctorat, Université de Franche-Comté Besançon , 2002.

- [48] D. ROY, F. VERNADAT, "Reactive Shop-Floor Control with multi-agent System". IFAC/IFIP Int Conf. on management and control of production and logistics, Brazil, pp. 426-431, 1997.
- [49] A. Mahdi, "Utilisation de métaheuristiques hybrides pour la résolution de problèmes d'agencement d'atelier de découpe 2D et d'ordonnancement", 2007.
- [50] T. DEWOLF, T. HOLVOET, "Emergence and self-organization: a statement of similarities and differences". In Proceedings of the International Workshop on Engineering Self-Organizing Applications, pp.96-110, 2004.
- [51] J. DEGUET, Y. DEMAZEAU, L. MAGNIN, "Elements about the emergence issue, a survey of emergence definitions". In Proceedings of the European Conference on Complex Systems (ECCS'05), Paris, France, 2005.
- [52] M. DEMPSTER, "A Self-Organizing Systems Perspective on Planning for Sustainability", Thèse de doctorat, University of Waterloo, School of Urban and Regional Planning, 1998.
- [53] J. GEORGÉ, "Résolution de Problèmes par Émergence. Étude d'un Environnement de Programmation Émergente". Thèse de doctorat, Université Toulouse III – Paul Sabatier, U.F.R. Mathématiques, Informatique et Gestion, 2004.
- [54] J. GOLDSTEIN, "Emergence as a construct: History and issues Emergence", vol. 1, pp. 49-72 1999.
- [55] F. HEYLIGHEN, "Self-organization, emergence and the architecture of complexity", In the 1st European Conference on System Science, 1989.
- [56] J. HOLLAND, "Emergence: from chaos to order". Addison-Wesley, 1998.
- [57] O. HOLLAND, C. MELHUISE, "Stigmergy, self-organization, and sorting in collective robotics", Artificial Life, vol. 5, pp. 173-202, 1999.
- [58] C. LUCAS, "Emergence and evolution - constraints on form", 2003.
- [59] J. P. MÜLLER, "Vers une méthodologie de conception de systèmes multi agents de résolution de problème par émergence", In JFIADSMA'98, 1998.
- [60] F. BENBOUZID SITAYEB, "Contribution à l'étude de la performance et de la robustesse des ordonnancements conjoints production/maintenance –cas du flow shop". Thèse de doctorat, Université de Franche-Comté Besançon, Juin 2005.
- [61] F. HEYLIGHEN, "The science of self-organization and adaptively". In The Encyclopedia of Life Support Systems, UNESCO Publishing-Eolss Publishers, 2002.
- [62] J. KAABI-HARRATH, "Contribution à l'ordonnancement des activités de maintenance dans les systèmes de production". Thèse de doctorat, Université de Franche-Comté Besançon, Juin 2005.
- [63] E. BONABEAU, G. THÉRAULAZ, "Auto-organisation et comportement, chapitre Auto-organisation et comportements collectifs : la modélisation des sociétés d'insectes", 1997.

- [64] F. SEMPÉ, "Auto-organisation d'une Collectivité de Robots : Application à l'Activité de patrouille en présence de perturbation", Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, 2004.
- [65] S. CAMAZINE, J. DENEUBOURG, N. FRANKS, J. SNEYD, G. THERAULAZ, E. BONABEAU, "Self-Organization in Biological Systems", Princeton University Press, 2000.
- [66] G. NICOLIS, I. PRIGOGINE, "Self-organization in Non-equilibrium Systems", New York, 1977.
- [67] K. SURESH, N. KUMARAPPAN, "Combined genetic algorithm and simulated annealing for preventive unit maintenance scheduling in power system", IEEE Power Engineering Society General Meeting, PES, 18-22 June, Montreal, Quebec, Canada, 2006.
- [68] M. DORIGO, V. MANIEZZO, A. COLORNI, "Ant system: Optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man and Cybernetics-Part B, vol. 26, pp. 29-41, 1996.
- [69] M. DORIGO, C. BLUM, "Ant colony optimization theory: A survey. Theoretical Computer Science", pp. 243-278, 2005.
- [70] A. COLORNI, M. DORIGO, et V. MANIEZZO. "Distributed optimization by ant colonies". In Proceedings of the First European Conference on Artificial Life, pp.134-142, 1991.
- [71] E. BONABEAU, M. DORIGO, et G. THERAULAZ, "Swarm Intelligence: From Natural to Artificial Systems". Oxford University Press, USA, 1999.
- [72] E. BONABEAU, et G. THERAULAZ, "Intelligence collective", Hermès, 1994.
- [73] E. BONABEAU, G. THERAULAZ, J. L. DENEUBOURG, et S. CAMAZINE, "Self-organization in social insects". Trends in Ecology and Evolution, vol. 12: pp. 188-193, 1997.
- [74] S. AUPETIT, N. MONMARCHÉ, M. SLIMANE, C. GUINOT, G. VENTURINI, "Clustering and Dynamic Data Visualization with Artificial Flying Insect", Genetic and Evolutionary Computation Conference (GECCO), 2003.
- [75] R.P. DOMINGOS, R. SCHIRRU, C.M.N.A. PEREIRA, "Particle Swarm Optimization in reactor core designs", Nuclear Science and Engineering, vol. 7, p. 152, 2006.
- [76] J. KENNEDY, R.C. EBERHART, "Particle Swarm Optimization", In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.
- [77] C.M.N.A. PEREIRA, C.M.F. LAPA, A.C.A. MOL, A.F. LUZ, "A PSO approach for preventive maintenance scheduling optimization", In: International Nuclear Atlantic Conference, 2009.
- [78] L. N. D. CASTRO, J. I. TIMMIS, "Artificial immune systems: A new computational intelligence approach", Berlin, Springer – Verlag, 2003.
- [79] L. N. D. CASTRO, "Artificial Immune Systems: Theory and Applications", VI-Brazilian

- Symposium on Neural Networks, 2000.
- [80] P. KOURILSKY, "Immunologie moléculaire", Académie des Sciences Collège de France, 2007.
- [81] S. A. HOFMEYR, S. FORREST, "Immunity by Design; An Artificial Immune System", Dept. of Computer Science University of New Mexico, 2004.
- [82] L. N. D. CASTRO, F. J. V. ZUBEN, "Artificial immune system: Part II- A survey of applications", Technical Report, DCA-RT, Feb 2000.
- [83] L. N. D. CASTRO, "An Introduction to the Artificial Immune Systems", ICANNGA - Prague, 22-25th April, 2001.
- [84] J. BROWNLEE, "Clonal Selection Theory & Clonal selection classification algorithm (CSA)", Master of Information Technology, Swinburne University of Technology, 2004.
- [85] L. N. D. CASTRO, J. TIMMIS, "In Artificial Neural Networks in Pattern Recognition Artificial Immune Systems: A Novel Paradigm to Pattern Recognition", University of Paisley, UK, pp. 67-84, 2002.
- [86] H. KHELIL, A. BENYETTOU, "Application du système immunitaire artificiel ordinaire et amélioré pour la reconnaissance des caractères artificiels", Laboratoire Signal Image Parole SIMPA, Université des sciences et de la technologie d'Oran, 2006.
- [87] M. M. MANTHA, "The truth about your immune system; what you need to know", Harvard College, États-Unis, 2004.
- [88] L. N. d. CASTRO , F. J. V. ZUBEN, "Learning and Optimization Using the Clonal Selection Principle", Transactions on Evolutionary Computation / Special Issue on Artificial Immune Systems, vol. 6, n. 3, pp. 239-251, 2002.
- [89] L.N D. CASTRO, J. TIMMIS, "Artificial immune system as a Novel Soft Computing paradigm", Computing laboratory, University of Kent at Canterbury, Soft Computing Journal, vol. 7 July, 2003.
- [90] L. N. D. CASTRO, F. J. V. ZUBEN, "The Construction of a Boolean Competitive Neural Network Using Ideas from Immunology", Neuro-computing, 50C, pp. 51-85, 2003.
- [91] L. N. D. CASTRO, F. J. V. ZUBEN, "Artificial immune systems: Part I – Basic theory and applications", Technical Report TR – DCA, Dec 1999.
- [92] A.WATKINS, "Exploiting immunological metaphors in the development of serial/parallel and distributed learning algorithms", PhD de l'université de Kent.
- [93] M. MOUTSCHEN, "Immunologie générale", Université de Liège, 2009.
- [94] N.K. JERNE, C. SALMON et R. ANDRE, "Les anticorps", Vulgaris-Médicale 2009.
- [95] U. AICKELIN, S. CAYZER, "The danger theory and its application to Artificial Immune Systems", 1st International Conference on Artificial Immune Systems, Canterbury, UK, pp. 141-148, 2002.
-

- [96] F. BURNET, "The clonal selection theory of acquired immunity", Nashville, TN, Vanderbilt University Press, 1959.
- [97] D. DASGUPTA, F. GONZALEZ, "Artificial Immune System (AIS) research in the last five years", The 2003 Congress on Evolutionary Computation, CEC '03, vol.1, pp. 123-130, 2003.
- [98] R.GOLDSBY, T. KINDT, B. OSBORNE, "Kuby Immunology", Fourth Edition, WH Freeman at MacMillan Press, 2000.
- [99] J. GREENSMITH, U. AICKELIN, J. TWYXCROSS, "Detecting Danger: Applying a Novel Immunological Concept to Intrusion Detection Systems", 6th International Conference on Adaptive Computing in Design and Manufacture (ACDM 2004), Bristol, UK, 2004.
- [100] S. A. HOFMEYR, "An interpretative introduction to the immune system", In Design Principles for the Immune System and Other Distributed Autonomous Systems, New York, Oxford University Press, 2000.
- [101] N. K. JERNE, "Towards a Network Theory of the Immune System", Annales d'Immunologie, Institut Pasteur, vol. 125, pp. 373-389, 1974.
- [102] P. MATZINGER, "An innate sense of danger", Seminars in Immunology, vol. 10, pp. 399-415, 1998.
- [103] A. HABOUSSI, "Systèmes Immunitaires Artificiels pour le diagnostic des systèmes complexes". Mémoire de Magister, université de BATNA, 2012.
- [104] J. TWYXCROSS, U. AICKELIN, "Towards a conceptual framework for innate immunity", Proceedings of the 4th International Conference on Artificial Immune Systems ICARIS-2005, LNCS 3627, Springer-Verlag, Banff, Canada, pp.112-125, 2005.
- [105] A. WATKINS, "AIRS: A resource limited artificial immune classifier", Thèse de l'université de Mississipi du département d'informatique. Mississipi, 2001.
- [106] A. WATKINS, L. BOGGESESS, "A New Classifier Based on Resources Limited Artificial Immune Systems". Proceeding congress de Evolutionary Computation, IEEE World Congress on Computational Intelligence held in Honolulu, pp 1546-1551. IEEE. USA, 2002.
- [107] A. WATKINS, J. TIMMIS, L. BOGGESESS, "Artificial immune recognition system (AIRS): an immune inspired supervised learning algorithm", Journal de l'académie Kluwer Genetic Programming and Evolvable Machines, pp 291-317, 2004.
- [108] J.D. FARMER, N.H. PACKARD, A.S. PERELSON, "The immune system, adaptation, and machine learning", Physica 22D 187-204, 1986.
- [109] D. DASGUPTA, "an Overview of Artificial Immune Systems and Their Applications", In Dasgupta, (editor), Artificial Immune Systems and Their Applications, Berlin: Springer-Verlag, pp.3-21, 1998.

- [110] S. FORREST, "Computer Immunology", Communications of the ACM, vol. 40, N°10, pp. 88-96, 1997.
- [111] S. FORREST, "Immunology as Information Processing", in Design Principles for Immune Systems and Other Distributed Autonomous Systems, 2000.
- [112] J. E. HUNT, D. E. COOKE, "Learning Using an Artificial Immune System ", Journal of Network and Computer Applications, vol. 19, 1996.
- [113] M. NEAL, J. TIMMIS, J. HUNT, "An artificial immune system for data analysis ", Pub-Med, 2000.
- [114] L. DOYEN, "Modélisation et évaluation de l'efficacité de la maintenance des systèmes réparables", Rapport Technique, Université Joseph Fourier, Rapport de DEA de Mathématiques Appliquées, 2001.
- [115] R.E. BARLOW, F. PROSCHAN, "Mathematical Theory of Reliability", Wiley, New-York, 1965.
- [116] G. BUDAI, R. DEKKER, R.P. NICOLAI, "A review of planning models for maintenance and production", Economic Institute report, 2006.
- [117] R. DEKKER, "Applications of maintenance optimization models: a review and analysis", Reliability Engineering and System Safety, vol. 51, N°3, pp. 229-240, Mars 1996.
- [118] C. BERENGUER, A. GRALL, B. CASTANIER, "Simulation and evaluation of condition based maintenance policies for multi-component continuous-state deteriorating systems", Proceedings of the European Safety and Reliability Conference - ESREL'2000 - 15-17 may 2000, Edinburgh, Ecosse, pp. 275-282, 2000.
- [119] E. CHATELET, C. BERENGUER, "Etat de l'art des methodes et outils de simulation des stratégies de maintenance", Rapport technique – Lot 1, 2001.
- [120] R.E. Barlow, L.C. Hunter, "Optimum preventive maintenance policies", Operation research, vol. 8, pp. 90-100, 1960.
- [121] E. DELOUX, "Politiques de maintenance conditionnelle pour un système à dégradation continue soumis à un environnement stressant", Thèse de doctorat, Ecole des mines de Nantes, 2008.
- [122] O. DOYEN, "Modélisation et évaluation de l'efficacité de la maintenance des systèmes réparables", Thèse de doctorat, INPG, 2004.
- [123] G. CELEUX, F. CORSET, A. LANNOY, et B. RICARD, "Designing a Bayesian network for preventive maintenance from expert opinions in a rapid and reliable way", Reliability Engineering and System Safety, vol. 91, N°7, pp. 849-856, Juillet 2006.
- [124] J.I ANSELL et M.J. PHILLIPS, "Practical aspects of modelling of repairable systems data using proportional hazards models", Reliability Engineering and System Safety, vol. 58, pp. 165-171, 1997.

- [125] A. BARROS, "Maintenance des systèmes multi composants sous surveillance imparfaite : modélisation stochastique et optimisation", PhD thesis, Université de Technologie de Troyes, 2003.
- [126] C. BUNEA et T. BEDFORD, "The robustness of maintenance optimisation to modeling assumptions", In Proceedings of the European Safety and Reliability Conference -ESREL'2001 - 16-20 September, Torino, Italy, 2001.
- [127] R. DEKKER, "Applications of maintenance optimization models: a review and analysis", Reliability Engineering and System Safety, vol. 51, pp. 229-240, 1996.
- [128] Y. DIJOUX et O. GAUDOIN, "The alert-delay competing risks model for maintenance analysis", Journal of Statistical Planning and Inference, In Press, 2008.
- [129] B. SMYTH et E. MCKENNA, "Competence models and the maintenance problem", Computational Intelligence: Special Issue on Maintaining Case-Based Reasoning Systems, In Press, 2002.
- [130] U. GURLER, A. KAYA, "A maintenance policy for a system with multi-state components: an approximate solution", Reliability Engineering and System Safety, vol. 76, N° 2, pp. 117-127, Mai 2002.
- [131] L. HAITAO, A. ELSAYED, Y. LING, "Chan Maintenance of continuously monitored degrading systems", European Journal of Operational Research, vol. 175, N° 2, pp. 821-835, Decembre 2006.
- [132] J.S. IVY, S.M. POLLOCK, "Marginally monotonic maintenance policies for a multi-state deteriorating machine with probabilistic monitoring and silent failures", IEEE Transactions on Reliability, vol. 54, N° 3, pp. 489-97, Septembre 2005.
- [133] W. FELLER, "An Introduction to Probability Theory and Its Applications", Wiley Series in Probability and Mathematical Statistics, 2nd edition, 1971.
- [134] W. KUBIAK, J. BLAZEWICZ, P. FORMANOWICZ, J. BREIT, G. SCHMIDT, Two-machine flow shops with limited machine availability, European Journal of Operational Research, pp. 528-540, 2002.
- [135] T. CHENG, G. WANG, "An improved heuristic for two-machine flow shop scheduling with an availability constraint, Operational Research Letters, vol. 26, pp.223-235, 2005.
- [136] H. ALLAOUI, A. ARTIBA, "Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints", Computers and Industrial Engineering, vol. 47, pp. 431-450, 2004.
- [137] O. ENGIN, A. DÖYEN, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", Future Generation Computer Systems, vol. 20, pp. 1083-1095, 2004.
- [138] I. ADIRI, J. BRUNO, E. FROSTING, "single machine flow-time scheduling with a single breakdown", Acta Inform, vol. 26, pp. 679-696, 1998.
- [139] C.R. REEVES, A genetic algorithm for flow shop sequencing, Computers and Operations Research
-

- vol. 22, 5-13, 2005.
- [140] E. KUTANOGLU, “Minimizing job tardiness using integrated preventive maintenance planning and production scheduling”, *IIE Transactions*, vol. 35, N°6, pp. 503-513, 2003.
- [141] J. BLAZEWICZ, J. BREIT, P. FORMANOWICZ, W. KUBIAK, G. SCHMIDT, “Heuristic algorithms for the two-machine flow shop problem with limited machine availability Omega”, vol. 29, pp. 599-608, 2001.
- [142] J. BREIT, “A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint”, *Computers and Operations Research*, vol. 33, pp. 2143–2153, 2006.
- [143] R. RUIZ, J.C. G. DÍAZ, C. MAROTO, “Considering scheduling and preventive maintenance in the flow shop sequencing problem”, *Computers and Operations Research*, vol. 34, pp. 3314–3330, 2007.
- [144] A. ALLAHVERDI, “Two-stage production scheduling with separated setup times and stochastic breakdowns”, *Journal of Operational Research Society*, vol. 46, pp. 896-904, 1995.
- [145] M. NAWAZ, E.E. ENSCORE, I. HAM, “A heuristic algorithm for the m-machine, n-job flow shop sequencing problem OMEGA”, *The International Journal of Management Science*, vol. 11, N°1, pp. 91–95, 1983.
- [146] D.L. YANG, C.J. HSU, W.H. KUO, “A two-machine flow shop scheduling problem with a separated maintenance constraint”, *Computers and Operations Research*, vol. 35, N° 3, pp. 876-883, 2008.
- [147] G. SCHMIDT, “Scheduling with limited machine availability”, *European Journal of Operational Research*, vol. 121, pp. 1–15, 2000.
- [148] W.R. BLISCHKE, D.N.P. MURTHY, “Reliability Modeling, Prediction, and Optimization”, John Wiley & Sons Inc., USA, 2000.
- [149] J.J. MCCALL, “Maintenance policies for stochastically failing equipment: a survey *Management Science*”, vol. 11, pp. 493-524, 1956.
- [150] W.P. PIERSKALL, J.A. VOELKER, “A survey of maintenance models: the control and surveillance of deteriorating systems”, *Naval Logistics Research Quarterly*, vol. 23, pp. 353–388, 1976.
- [151] Y.S. SHERIF, M.L. SMITH, “Optimal maintenance models for systems subject to failure”, a review *Naval Logistics Research Quarterly*, vol. 23, pp. 47–74, 1976.
- [152] J. HOLLAND, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, USA, 1975.
- [153] D.E. GOLDBERG, “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading Addison-Wesley, USA, 1989.
-

- [154] B.A. NORMAN, J.C. BEAN, "A genetic algorithm methodology for complex scheduling problems", *Naval Research Logistics*, vol. 46, pp. 199-211, 1999.
- [155] M. ZANDIEH, S.M.T. F. GHOMI, S.M. M. HUSSEINI, "An immune algorithm approach to hybrid flow shops scheduling with sequence dependent setup times", *Journal of Applied Mathematics and Computation*, vol. 180, pp. 111-127, 2006.
- [156] R.H. YEH, "State-age-dependent maintenance policies for deteriorating systems with Erlang sojourn time distributions", *Reliability Engineering and System Safety*, vol. 58, pp. 55-60, 1997.
- [157] A.K.S. JARDINE, T. JOSEPH, D. BANJEVIC, "Optimizing condition-based maintenance decisions for equipment subject to vibration monitoring", *Journal of Quality and Maintenance Engineering*, vol. 5, pp. 1355-2511, 1999.
- [158] C.T. LAM, R.H. YEH, "Optimal replacement policies for deteriorating systems under various maintenance strategies", *IEEE Transactions on reliability*, vol. 43 pp. 423-430, 1994.
- [159] C.M.F. LAPA, C.M.N.A. Pereira, M.P. D. BARROS, "A model for preventive maintenance planning by genetic algorithms based in cost and reliability", *Reliability Engineering and System Safety*, vol. 91, N°2, pp. 233-240, Février 2006.
- [160] S.WU, D. CLEMENTS-CROOME, Optimal maintenance policies under different operational schedules. *IEEE Transactions on Reliability*, vol. 54, N°2, pp. 338-346, 2005.
- [161] N. ABRAMSON, "Information Theory and Coding", McGrawHill, New York, 1963.
- [162] A. ABDELHADI, L-H, MOUSS," A New Method Based on Multi Agent System and Artificial Immune System for Systematic Maintenance", *Research Journal of Applied Sciences, Engineering and Technology (RJASET)*, Vol. 7, N°19, pp. 4008-4017, ISSN: 2040-7459, Mai 2014.
- [163] A. ABDELHADI, L-H, MOUSS, " An efficient hybrid approach based on multi agent system and emergence method for the integration of systematic preventive maintenance policies in hybrid flow-shop scheduling to minimize makespan", *Journal of Mechanical Engineering Research (JMER)*, VOL. 5, N°6, pp. 112-122, ISSN 2141-2383, September 2013
- [164] A. ABDELHADI, L-H, MOUSS, "The use of artificial immune system algorithms in monitoring industrial", *IEEE Explorer*, pp. 50-55, Print ISBN:978-1-4673-1657-6, June 2013.
- [165] A. ABDELHADI, L-H, MOUSS, " An Overview of Artificial Immune System Algorithms for Industrial Monitoring", *International Review on Computers and software (IRECOS)*, VOL. 6, N°2, pp. 269-274, ISSN 1828-6003 Mars 2011
- [166] A. ABDELHADI, L-H. MOUSS, O. KADRI, " Une nouvelle méthode basée sur le système multi-agents et le système immunitaire artificiel pour la maintenance systématique", *ICIEM'2014 International Conference on Industrial Engineering and Manufacturing*, Mai 6-7, 2014, Batna-Algeria
- [167] A. ABDELHADI, L-H. MOUSS, O. KADRI, " Une approche hybride basé sur le système multi-

- agents et les méthodes d'émergence pour la maintenance systématique“, 8th International Conference Integrated Design And Production Cpi 2013 , October 21-23, 2013 – Tlemcen – Algeria
- [168] A. ABDELHADI, L-H. MOUSS, O. KADRI, "Les algorithmes du système immunitaire Artificiel pour la surveillance industrielle", ICIEM'10 International Conference on Industrial Engineering and Manufacturing, Mai 9-10, 2010, Batna–Algeria.

Annexes

Annexe 1. Production Scientifique

II.1 Publications dans des revues internationales

- A. **ABDELHADI**, L-H, MOUSS, ” A New Method Based on Multi Agent System and Artificial Immune System for Systematic Maintenance”, Research Journal of Applied Sciences, Engineering and Technology (RJASET), Vol. 7, N°19, pp. 4008-4017, ISSN: 2040-7459, **Mai 2014**. <http://maxwellsci.com/print/rjaset/v7-4008-4017.pdf>
- A. **ABDELHADI**, L-H, MOUSS, ” An efficient hybrid approach based on multi agent system and emergence method for the integration of systematic preventive maintenance policies in hybrid flow-shop scheduling to minimize makespan”, Journal of Mechanical Engineering Research (JMÉR), VOL. 5, N°6, pp. 112-122, ISSN 2141-2383, **September 2013**. <http://www.academicjournals.org/JMER>
- A. **ABDELHADI**,”Algorithms for the integration of Arabic Language in Mobile Phones“, Book, LAP Lambert Academic Publishing, ISBN: 978-3-659-38488-2, Published on: **Mai 2013**.
<https://www.lap-publishing.com/catalog/details//store/gb/book/978-3-659-38488-2/algorithms-for-the-integration-of-arabic-language-in-mobile-phones>.
- A. **ABDELHADI**, L-H, MOUSS, “The use of artificial immune system algorithms in monitoring industrial“, IEEE Explorer, pp. 50-55, Print ISBN:978-1-4673-1657-6, **June 2013**.
<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6475437>

- A. **ABDELHADI**, L-H, MOUSS, " An Overview of Artificial Immune System Algorithms for Industrial Monitoring", International Review on Computers and software (**IRECOS**), VOL. 6, N°2, pp. 269-274, ISSN 1828-6003 **Mars 2011**, www.IRECOS.org.
- A. **ABDELHADI**, L-H, MOUSS," Efficient Tool for the Recognition of the Leaves of Plants", International Journal of Computer Science Issues (**IJCSI**), Vol. 8, N°2, ISSN: 1694-0814, **March 2011**, www.IJCSI.org.
- A. **ABDELHADI**, L-H. MOUSS, O. KADRI , " Efficient Algorithms for the integration of Arabic Language in Mobile Phone", International Journal of Computer and Electrical Engineering (**IJCEE**), Vol.3, N°3, pp. 379-383, ISSN: 1793-8163, **June 2011**, www.IJCEE.org.

II.2 Publications dans des revues nationales:

- 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", Revue RST, ISSN : 1112-9697, Vol. 4 N°1, p. 35-52, 2013. <http://revues.univ-batna.dz/rst2/images/vol4n1/283.pdf>
- O. KADRI, L. H. MOUSS, A. **ABDELHADI**, R. MAHDAOUI, " Vers une Optimisation de l'Algorithme AntTreeStoch", Revue RST, ISSN : 1112-9697, Vol. 3 N°1, p. 125-134, 2012. <http://revues.univ-batna.dz/rst2/images/vol3n1/14KADRI.pdf>

II.3 Communications internationales:

- A. **ABDELHADI**, L-H. MOUSS, O. KADRI, " Une nouvelle méthode basée sur le système multi-agents et le système immunitaire artificiel pour la maintenance systématique", **ICIEM'2014** International Conference on Industrial Engineering and Manufacturing, Mai 6-7, **2014**, Batna–Algeria.

- A. ABDELHADI, L-H. MOUSS, O. KADRI, “ Une approche hybride basé sur le système multi-agents et les méthodes d'émergence pour la maintenance systématique“, 8th International Conference Integrated Design And Production Cpi **2013** , October 21-23, 2013 – Tlemcen – Algeria. http://cpi2013.univ-nantes.fr/Documents/Programme-version%20finale_CPI%202013.pdf
- A. **ABDELHADI**, L-H. MOUSS, O. KADRI, «The use of Artificial Immune System Algorithms for Industrial Monitoring ”, **SETIT**, March 21-24, **2012**, Sousse-Tunisia.
- L-H. MOUSS, O. KADRI, F. MERAH, A. **ABDELHADI**, D. MOUSS " An efficient hybrid approach based on SVM and Binary ACO for feature Selection ", **ICMAS'11** International Conference on Modeling and Applied Simulation September 12-15, **2011**, Rome, Italy.
- A. **ABDELHADI**, L-H. MOUSS, O. KADRI, "Les algorithmes du système immunitaire Artificiel pour la surveillance industrielle", **ICIEM'10** International Conference on Industrial Engineering and Manufacturing, Mai 9-10, **2010**, Batna–Algeria.
- A. **ABDELHADI**, H-L. MOUSS, O. KADRI, "Efficient Tool for the Recognition of the Leaves of Plants", **ICIEM'10** International Conference on Industrial Engineering and Manufacturing, Mai 9-10, **2010**, Batna–Algeria.
- O. Kadri, L-H. MOUSS, A. **ABDELHADI**, et R. MAHDAOUI, "Vers une optimisation de l'algorithme anttreestoch", **ICIEM'10** International Conference on Industrial Engineering and Manufacturing, Mai 9-10, **2010**, Batna–Algeria.
- O. KADRI, F. MERAH, L-H. MOUSS, A. **ABDELHADI**, "Un algorithme de colonie de fourmis pour la réduction de la taille du vecteur d'état d'un système industriel », **ICIEM'10** International Conference on Industrial Engineering and Manufacturing, Mai 9-10, **2010**, Batna–Algeria.

II.4 Communications nationales:

- A. **ABDELHADI**, S. ADOUANE, O. KADRI, L-H. MOUSS, "Efficient Tool for the Recognition of the Leaves of Plants", **JEESI**, Mai 19, 2009, Algiers, Algeria.

- A. ABDELHADI, O. KADRI, L-H, MOUSS, "Efficient Algorithms for the integration of Arabic Language in Mobile Phone", JEESI, Mai 19, 2009, Algiers-Algria.

Annexe 2. Partie du code source

II.1 Introduction

Nos approches multi agent sont développées par la plateforme de développement Netlogo. Nous donnons un aperçu sur cette plateforme.

II.2 Pourquoi 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 [36]. 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.

Netlogo convient tout particulièrement à la modélisation de systèmes complexes évoluant au cours du temps. Les « modélisateurs » peuvent donner des instructions à des

centaines ou des milliers d'agents réactifs, opérant indépendamment les uns des autres. Ce qui permet d'explorer les liens entre les comportements des individus à leur niveau et les schémas généraux (comportements de groupe ou de masse) qui émergent des interactions entre de nombreux individus.

Netlogo représente la nouvelle génération de toute une série de langages de modélisations multi-agents. Il a été construit à partir des bases fournies par le logiciel StarLogoT auquel il apporte toute une série de nouvelles fonctionnalités significatives ainsi qu'un langage et une interface utilisateur entièrement remaniés. La figure II.1 présente l'environnement de développement Netlogo.

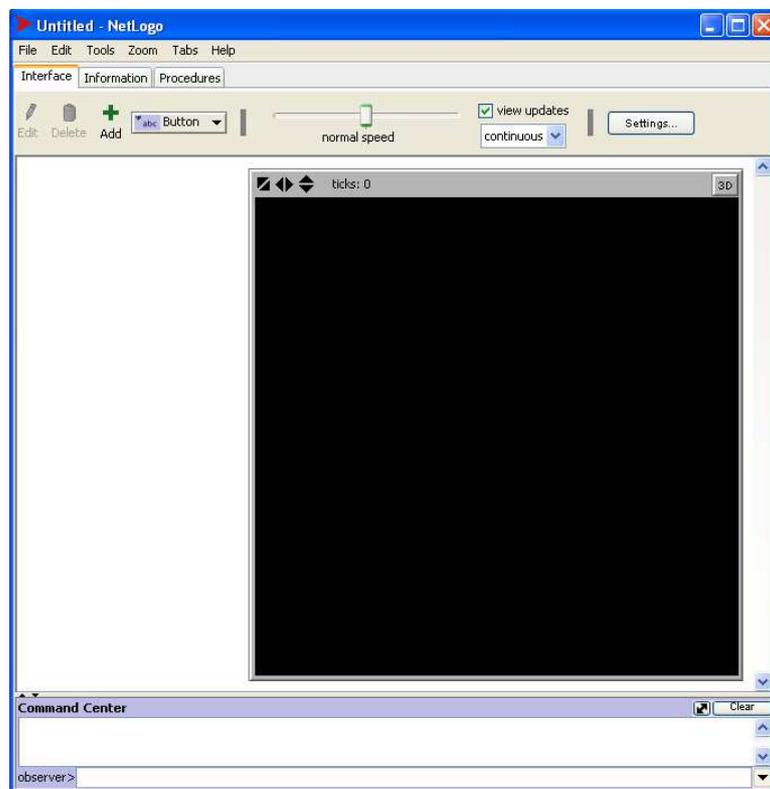


Figure II.1 Environnement de développement Netlogo

Netlogo étant écrit en Java, il peut tourner sur tous les systèmes majeurs (Mac, Windows, Linux et autres). Il fonctionne en tant qu'application indépendante. Les modèles peuvent même être sauvegardés sous forme d'applets Java et tourner dans tous les navigateurs internet modernes.

II.3 Les concepts de base de Netlogo

Le monde de Netlogo est constitué d'*Agents réactif*, qui peuvent suivre des instructions. Les activités des différents agents s'exécutent simultanément. Il y a 3 types d'agents : tortues, patches et observateur.

- Les **Tortues** sont les agents qui se déplacent dans le monde. Ils correspondent aux agents.
- Le **Monde** est en 2D ou 3D, divisé selon une grille (torique ou non) de patchs.
- Un **Patch** est une portion de sol sur laquelle les tortues peuvent se situer et se déplacer. Les patches correspondent au concept d'environnement vu en cours
- L'**Observateur** regarde de l'extérieur le monde des tortues et des patchs (n'est pas situé dans le monde).

II.4 Partie du code source

// Chaque solution potentielle est représentée par un agent (turtle)

```
extensions [array]
turtles-own [
  ch      ;; le chromosome
  task-order
  fitness
]
globals [
  ch1
  ch2
  ch3
  winner
  gantt   ;; L'agent qui possède la meilleure solution
]
to setup
  clear-all
  ;; creation d'un agent reference
```

```
set gantt array:from-list n-values machine_number [array:from-list n-values
gantt_length [false]]
set ch3 array:from-list n-values task_number [array:from-list n-values machine_number
[array:from-list n-values 2[user-input "entrer la valeur suivante?"]]]
create-turtles agent_number [
set task-order ordonnancement
set ch chaleatoire
calculate-fitness
hide-turtle ;; the turtles' locations are not used, so hide them
]
End
```

to-report ordonnancement

```
let i 0
let order 0
repeat task_number - 1 [
set i i + 1
set order sentence order i ]
set order shuffle order
report order
end
```

to-report chaleatoire

```
let i 0
let j 0
set ch1 0
set i 0
set ch2 array:item ch3 item 0 task-order
repeat task_number - 1 [
set i i + 1
set j item i task-order
set ch1 array:item ch3 j
set ch2 sentence ch2 ch1
]
set ch2 array:from-list ch2
report ch2
```

end

to calculate-fitness

set fitness insertion

 ;insertion dans le tableau gantt en utilisant la methode recherche

 ; chercher la valeur maximale dans les lignes du tableau de gantt

end

to-report insertion

set gantt array:from-list n-values machine_number [array:from-list n-values

gantt_length [false]]

let machine 0

let periode 0

let t 0

let maxt 0

let i 0

while [i < task_number] [set t 0

 let j 0

 while [j < machine_number] [

 set machine array:item (array:item (array:item ch i) j) 0

 set periode array:item (array:item (array:item ch i) j) 1

set t recherche machine periode t

set j j + 1]

 set i i + 1

 if t > maxt [set maxt t]]

report maxt

end

to creation-generation-suivante

let old-generation turtles with [true]

let crossover-count (floor (agent_number * crossover-rate / 100 / 2))

repeat crossover-count

[

 let parent1 min-one-of (n-of 3 old-generation) [fitness]

 let parent2 min-one-of (n-of 3 old-generation) [fitness]

 let child-bits crossover ([task-order] of parent1) ([task-order] of parent2)

```
ask parent1 [ hatch 1 [ set task-order item 0 child-bits ] ]
ask parent2 [ hatch 1 [ set task-order item 1 child-bits ] ]
]
repeat (agent_number - crossover-count * 2)
[
ask min-one-of (n-of 3 old-generation) [fitness]
[ hatch 1 ]
]
ask old-generation [ die ]
ask turtles
[
calculate-fitness
]
End
to-report crossover [task-order1 task-order2]
; creation du premier enfant
let element1 one-of task-order1
let pos1 position element1 task-order1
let i 0
let e1 element1
if pos1 != 0
[ set e1 item 0 task-order2
set i 1]
repeat length task-order1
[
ifelse i = pos1
[set e1 sentence e1 element1]
[ if element1 != item i task-order2 [set e1 sentence e1 item i task-order2]
set i i + 1]
]
; creation de deuxième enfant
let element2 one-of task-order2
let pos2 position element2 task-order2
let j 0
```

```
let e2 element2
if pos1 != 0
  [ set e2 item 0 task-order1
    set j 1]
repeat length task-order2
  [
  ifelse j = pos2
    [set e2 sentence e2 element2]
    [ if element2 != item j task-order2 [set e2 sentence e2 item j task-order1]
      set i i + 1]
  ]
report list e1 e2
end
to mutate ;; turtle procedure
  set bits map [ifelse-value (random-float 100.0 < mutation-rate) [1 - ?] [?]]
bits
end
```


Résumé

L'objectif principal de notre thèse est de proposer des approches hybride basées sur les SMA d'un côté, et l'exploitation de méthodes d'émergences tels que les Algorithmes Génétiques (AG) et les algorithmes du Système Immunitaire Artificiel (SIA), pour mieux intégrer des politiques de maintenance préventive systématique dans les ateliers d'ordonnancement. L'objectif est de minimiser les temps d'exécutions pendant le déroulement de l'ordonnancement.

Nous avons utilisé des opérateurs avancés ; la clé aléatoire pour le codage, la sélection de classement, le croisement uniforme et la mutation ponctuelle unique.

Par ailleurs, nous avons défini une nouvelle procédure de calcul d'affinité dans le cadre de l'approche FSHMAIA. Cette procédure est basée sur le calcul d'un rapport de similarité des anticorps. Pour cela, nous avons proposé un algorithme basé sur quarts critère pour calculer le rapport de similarité.

Nous avons également évalué les adaptations de certaines heuristiques bien connues, y compris Johnson ($m/2, m/2$), NEH_H , PCDT et PLDT,

Un benchmark a été établi pour évaluer les algorithmes. Le benchmark contenu jusqu'à 100 jobs et 8 étages. Tous les résultats ont montré que le FSHMAIA donne les meilleurs résultats par rapport aux autres algorithmes, dans les trois politiques de la MP.

L'originalité de cette thèse réside dans l'utilisation des SMA avec les AG et les SIA dans l'intégration des politiques de maintenance préventive systématique dans un ordonnancement de type flow shop hybride.

Mots clés : *Systemes multi agents, Emergence, maintenance systématique, Système immunitaire artificiel, Algorithme génétique, Ordonnancement, Flow shop hybride.*

Abstract

The main objective of our thesis is to propose hybrid approaches based on MAS on one side, and the exploitation of emergences methods such as Genetic Algorithms (GA) and the algorithms of Artificial Immune System (AIS) to better integrate systematic preventive maintenance policies in scheduling workshops.

The objective is to minimize the time of executions during the course of scheduling. We use advanced operators; the random key for encoding, the selection of classification, the uniform crossover and single point mutation.

Furthermore, we have defined a new affinity calculation procedure within the FSHMAIA approach. This procedure is based on the calculation of an antibody of the similarity ratio. For this, we have proposed an algorithm based on quad criterion for calculating the ratio of similarity.

We also evaluated the adaptations of some well-known heuristics, including Johnson ($m/2, m/2$) NEH_H , PCDT and PLDT,

A benchmark has been established to evaluate the algorithms. The benchmark content up to 100 jobs and 8 floors. All results have shown that the FSHMAIA gives the best results compared to other algorithms in the three policies of PM.

The originality of this thesis lies in the use of MAS with GA and AIS in the integration of systematic preventive maintenance policies in a hybrid flow shop scheduling.

Keywords: *Multi Agent system, Emergence, Systematic maintenance, Artificial Immune System, Genetic Algorithm, Scheduling, Hybrid flow shop.*

ملخص

الهدف الرئيسي من هذه الأطروحة هو اقتراح النهج الهجين مبني على أساس نظام متعدد الوكلاء (MAS) من جهة ، واستغلال طرق التولد (EMERGENCES) مثل الخوارزميات الوراثية (GA) و خوارزميات الجهاز المناعي الاصطناعي (AIS) لتحسين دمج سياسات الصيانة الوقائية المنتظمة في ورشات عمل الجدولة . الهدف هو تقليل وقت العمل خلال الجدولة.

لقد استخدمنا عوامل متقدمة مثل: استعمال المفتاح عشوائي لترميز، والاختيار التصنيفي، التهجين الموحد وطفرة النقطة الواحدة. علاوة على ذلك، فإننا قد حددنا إجراء حساب تقارب جديد ضمن نهج FSHMAIA ويستند هذا الإجراء على حساب وجود الأجسام المضادة لنسبة التشابه. لهذا، اقترحنا خوارزمية على أساس معيار رباعية لحساب النسبة التشابه. لقد قمنا كذلك بتقييم التعديلات لبعض الخوارزميات المعروفة، مثل جونسون (JOHNSON)، NEH_H ، PCDT و PLDT، وقد أنشئنا معيارا لتقييم الخوارزميات. محتوى التقييم يصل إلى 100 عمل و 8 طوابق. وقد أظهرت النتائج أن نهج FSHMAIA يعطي أفضل النتائج مقارنة بخوارزميات أخرى في السياسات الثلاث من الصيانة الوقائية المنتظمة.

أصالة من هذه الرسالة تكمن في اقتراح نهج هجين مبني على نظام متعدد الوكلاء مع الخوارزميات الوراثية و خوارزميات الجهاز المناعي الاصطناعي في دمج سياسات جدولة الصيانة وقائية في ورش التدفق الهجين.

كلمات البحث : نظام متعدد الوكلاء، التولد، الصيانة الروتينية، الجهاز المناعي الاصطناعي، الخوارزمية الجينية، الجدولة، ورشة التدفق الهجين،