

Université SAAD DAHLAB de BLIDA



Faculté des Sciences

Département de: MATHÉMATIQUES

Mémoire présenté par :

BOUGUERRA Fatiha

En vue d'obtenir le diplôme de Master

Domaine: Mathématiques et Informatique

Filière: Mathématiques

Spécialité: Mathématiques appliquées

Option: Recherche opérationnelle

**Sujet : Résolution du problème de Job Shop par le Système
Immunitaire Artificiel (SIA)**

Soutenue le **13** juillet 2010, devant le jury composé de :

Mr. TAMI	M.A.A à USDB	Président
Mr. SAKRI	M.A.A à C.U.K.M	Examinateur
M ^{elle} . MESSAOUDI	M.A.A à USDB	Examinateur
Mr. DERBALA	M.C.A à USDB	Promoteur

MA-510-03-1

A mes chères parents, mes frères et ma sœur.

A mes oncles, ma grande mère et ma tante.

*A mes amis Scumia H., Amina B., Djouher B et
Amina B.*

Remerciement

*Je remercie tout d'abord, notre vénéré Allah,
Le tout puissant, à qui nous devons le tout.*

Je souhaite remercier M.DERBALA Ali pour avoir accepté d'être mon promoteur et de me suivre dans l'élaboration de ce mémoire. Je lui suis particulièrement reconnaissant pour sa patience, son aide, sa disponibilité et ses nombreuses suggestions qui ont amélioré ce travail.

Je tiens à remercier *tout* d'abord M.TAMI Omar, M^{elle} MESSAOUDI Nadia et M.REDHA Sakri pour nous avoir fait l'honneur d'être membres du jury. Ainsi que pour avoir consacré une partie de Leur temps précieux pour lire et corriger ce mémoire.

Je ne saurais oublier de remercier mes chers parents, qui étaient toujours à mes côtés et m'avaient tant aidé et soutenu. Qu'ils trouvent ici l'expression de ma sincère gratitude et ma profonde reconnaissance.

Je remercie tous ceux qui ont m'aidé de loin ou de près en particulier M.SAFAR dans la réalisation de ce mémoire.

Merci à tout le monde.

Résumé

Le problème d'ordonnancement de type job shop de n tâches, m machines est parmi l'un des problèmes généraux d'ordonnancement de la production.

Il existe plusieurs heuristiques qui donnent des solutions pour des problèmes de petites tailles avec des solutions quasi-optimales.

Ce mémoire étudie le critère de la minimisation des temps d'exécution pour l'ordonnancement des problèmes de différentes tailles. La méthode proposée de calcul de l'algorithme du système immunitaire artificiel est utilisée pour trouver les valeurs optimales de minimisation de temps des problèmes de différentes tailles. L'algorithme du système immunitaire artificiel est testé avec 162 références [10 (ORB1-ORB10), 5 (ARZ5-ARZ9), 40 (LAOI-LA40), 80 (TAO1-TA80), 04 (Yn01-Yn04), 20 (SWV01-SWV20), 3 (FT06, FT10, FT20)].

Les résultats montrent que le SIA algorithme est un algorithme efficace. Il donne de meilleurs résultats que la procédure de recherche tabou de la machine goulot (TSSB). Il est aussi la meilleure solution de la procédure du décalage de la machine goulot (goulot d'étranglement).

Mots clés : ordonnancement, job shop, système immunitaire artificiel, benchmarks.

Abstract

The n-job, m-machine job shop scheduling (JSS) problem is one of the general production scheduling problems. Many existing heuristics give solutions for small size problems with near optimal solutions.

This thesis deals with the criterion of makespan minimization for the job shop scheduling of different size problems. The proposed computational method of artificial immune system algorithm (AIS) is used for finding optimal makespan values of different size problems. The artificial immune system algorithm is tested with 162 benchmark problems [10 (ORB1-ORB10), 5 (ARZ5-ARZ9), 40 (LAOI-LA40), 80 (TAO1-TA80), 04 (Yn01-Yn04), 20 (SWV01-SWV20), 3 (FT06, FT10, FT20)]. The results show that the AIS algorithm is an efficient and effective algorithm which gives better results than the Tabu search shifting bottleneck procedure (TSSB) as well as the best solution of shifting bottleneck procedure.

Key words: Scheduling, Job Shop, Artificial Immune System, Benchmark.

ملخص

تعتبر مسألة ترتيب الأشغال ذات السير المتعدد (job shop) أحد أنماط مسائل ترتيب الأشغال العامة. يوجد العديد من الخوارزميات لحل المسائل الصغيرة بقيمة أحسن ومأمولة لكن تقريبية.

تتضمن الأطروحة دراسة خاصة اختصار أطول مدة لإنجاز هذه المسألة ، والطريقة الحسابية المستعملة هي خوارزمية نظام المناعة الاصطناعي من أجل الحصول على أطول مدة لإنجاز المسائل الصغيرة و الكبيرة للنمط المستعمل.

لقد طبقنا هذه الخوارزمية على 162 مسألة متعامل بها عالميا (Benchmarks), و هي [10 (ORB1- ORB10), 5 (ARZ5- ARZ9), 40 (LA40- LA01), 80 (TA80- TAO1),

04 (Yn04- Yn01), 20 (SWV20- SWV01), 3 (FT10, FT06)] و قد أظهرت النتائج

فعالية هذه الخوارزمية فعاليتها بإعطائها نتائج جيدة.

الكلمات الدالة : ترتيب الأشغال ، السير المتعدد، نظام المناعة الاصطناعي ، خوارزمية.

Table de matières

Dédicace	
Remerciement	
Résumé	
Introduction.....	1
Chapitre 1 : L'ordonnancement de type Job Shop	
1.1 Introduction.....	4
1.2 Caractérisation des machines.....	5
1.3 Les taches.....	7
1.4 Fonction objectif d'un problème d'ordonnancement.....	9
1.5 Atelier à cheminement multiples: le Job Shop	10
1.6 Résolution d'un Job Shop.....	11
1.7 Conclusion.....	12
Chapitre 2 : La méthode de SPE	
2.1 Introduction	14
2.2 Exemple : résolution d'un job shop de trois pièces à trois machines.....	14
Chapitre 3 : Le Système Immunitaire Humain	
3.1 Introduction.....	20
3.2 Les agressions du SIH	21
3.3 Mécanisme de défense non spécifique proliférer.....	22
3.3.1 Cellules phagocytaires.....	23
3.3.2 Barrières du SIH.....	22
3.3.3 Réaction inflammatoire.....	23
3.3.4 Système lymphatique.....	24
3.4 Mécanisme de défense spécifique.....	24
3.5 Système immunitaire cellulaire.....	25
3.6 Système immunitaire humorale.....	25
3.7 Conclusion.....	26
Chapitre 4 : Le Système Immunitaire Artificiel	
4.1 Introduction.....	27
4.2 La maturation de l'affinité.....	28
4.3 Représentation et affinité.....	28
4.4 Modèle de la sélection clonale.....	29
4.5 L'algorithme de la sélection clonale.....	31
4.6 Conclusion.....	32
Chapitre 5 : Application du SIA pour résoudre le problème du Job Shop	
5.1 Introduction	34
5.2 Le système immunitaire artificiel.....	34
5.2.1 Le principe de la sélection clonale.....	34

5.2.2 Le principe d'affinité de la maturation.....	35
5.3 Une analogie entre le Job Shop et le SIA.....	38
5.4 Un algorithme de SIA pour un problème de Job Shop.....	39
5.4.1 L'heuristique du calcul du makespan.....	41
5.4.2 Critère d'arrêt.....	41
5.5 Exemple.....	42
5.6 Conclusion.....	43
Chapitre 6 : Implémentation de SIA	
6.1 Introduction.....	45
6.2 Le langage de la programmation.....	45
6.3 Benchmarks.....	46
6.4 Description de l'application informatique	47
6.4.1 Spécification des instances (problèmes).....	48
6.4.2 Présentation des résultats	49
6.4.3 Le nombre de générations	50
6.5 Résultats généraux	51
6.6 Conclusion.....	55
Conclusion.....	57
Annexe	
1. Définitions Biologiques	
2. Résultats de la OR-Library	
3. Résultats de l'application	
Bibliographie	

Liste des tables

Tableau 2.1 : Temps d'exécution des pièces.....	15
Tableau 2.2 : Les opérations.....	15
Tableau 4.1: Algorithme de la sélection clonale	32
Tableau 5.1: Procédure de clonage.....	34
Tableau 5.2 : Procédure de mutation	36
Tableau 5.3 : Processus d'éditations des récepteurs.....	36
Tableau 5.4 : Analogie entre le Job Shop et le SIA.....	37
Tableau 5.5 : Algorithme de SIA.....	39
Tableau 5.6: Le temps d'exécution de chaque tâche.....	40
Tableau 5.7: Les opérations de chaque tâche.....	40
Tableau 6.1 : Résultats généraux obtenus par application du SIA (comparés avec ceux obtenus par TSSB).....	52
Tableau B.1 : Les différentes instances de la 1 ^{ère} partie.....	3
Tableau B.2: Les bornes ou la solution optimale de chaque instance.....	4
Tableau B.3 : Les auteurs des épreuves d'instances.....	5
Tableau B.4 : Les bornes ou la solution optimale de chaque instance.....	6
Tableau B.5 : Les auteurs des épreuves d'instances.....	6
Tableau C.1 : Résultats du SIA comparés avec la solution optimale des instances de classe 1.....	3
Tableau C.2: Résultats du SIA comparés avec la solution optimale des instances de classe 2.....	4
Tableau C.3 : Résultats du SIA comparés avec la solution optimale des instances de classe 3.....	5
Tableau C.4: Résultats du SIA comparés avec la solution optimale des instances de classe 4.	5
Tableau C.5: Résultats du SIA comparés avec la solution optimale des instances de classe 5.	6

Liste des figures

Figure 1.1 : Types de machines.....	6
Figure 1.2 : Job Shop: les produits empruntent des chemins distincts.....	11
Figure 1.3 : Principe de résolution du <i>jobshop</i> simple.....	12
Figure 2.1 : Diagramme " pièces " calé au plus tôt du sous ensemble A_0	16
Figure 2.2 : Diagramme " pièces " calé au plus tôt du sous ensemble A_1	17
Figure 2.3 : Diagramme " pièces " calé au plus tôt du sous ensemble A_4	17
Figure 2.4 : Arborescence des séparations.....	18
Figure 3.1 : Organes et tissus lymphoïdes.....	24
Figure 4.1 : Affinité du complexe Anticorps.....	28
Figure 4.2 : La sélection clonale.....	30
Figure 4.3 : L'organigramme de la sélection clonale.....	31
Figure 5.1 : La mutation inverse.....	35
Figure 5.2 : La mutation de simple échange de deux positions.....	36
Figure 5.3 : Organigramme d'algorithme de SIA pour résoudre le job shop.....	38
Figure 5.4 : Diagramme de Gantt pour un ordonnancement optimal.....	40
Figure 6.1 : Exemple d'un benchmark présenté par AIS-JSS(Ta4).....	47
Figure 6.2 : L'interface graphique d'AIS-JSS.....	48
Figure 6.3 : Ouvrir une Benchmark.....	48
Figure 6.4 : Générer une nouvelle instance.....	49
Figure 6.5 : La courbe de l'évolution avec de la résolution avec le SIA.....	49
Figure 6.6 : Diagramme de Gantt représentant la solution de SIA pour <i>fit6</i>	50
Figure 6.7 : Amélioration de la solution durant l'évolution des générations de l'Algorithme de SIA.....	50
Figure 6.8 : Présentations graphiques des résultats des instances TA1-TA30.....	53
Figure 6.9 : Présentations graphiques des résultats des instances TA31-TA60.....	54
Figure 6.10 : Présentations graphiques des résultats des instances TA61-TA80.....	55
Figure C.1 : Graphiques des instances de classe 1 (LA1-LA20).....	3
Figure C.2 : Graphiques des instances de classe 1 (LA20-LA40).....	3
Figure C.3 : Graphiques des instances de classe 2.....	4

Figure C.4 : Graphiques des instances de classe 3.....	5
Figure C.5: Graphiques des instances de classe 4.....	6
Figure C.6: Graphiques des instances de classe 5.....	6

Introduction générale

« N » tâches indépendantes doivent être exécutées sur « m » machines disposées en série dont l'objectif est de minimiser la durée totale de l'ordonnancement. Il est appelé problème du « Job Shop » et est souvent classé comme un problème d'optimisation combinatoire « difficile ».

Quand les tailles des données de « N » et « m » sont limitées ou moindres, une résolution exacte de ce problème, par exemple de type « calage à gauche », peut être prodiguée.

Sinon, si leurs tailles sont grandes, ces problèmes deviennent difficiles. Si on se contente d'une solution approchée, une méta-heuristique de type « recherche Taboue », recuit simulé, algorithme génétique etc. peuvent être utilisées.

Ces problèmes se rencontrent dans les ateliers de production ou chaînes de montages, la fabrication aéronautique, dans la construction navale etc.

Ce problème a été étudié auparavant par un binôme d'étudiants, Bataoui et Messaoudi, de notre département [16]. Ils ont défini et ont montré que les ensembles des ordonnancements actifs, où aucun décalage à gauche n'est possible, est le petit sous-ensemble dominant contenant la *solution optimale*. Ils ont présenté la formulation du problème par un graphe disjonctif, qui constitue l'outil de base pour les méthodes de résolution. Elles sont de deux types, une procédure par séparation et évaluation et des heuristiques de types algorithme de liste selon les règles de priorité SPT (shortest processing time), MWKR (most work remeaning) et LWKR (least work remeaning). Un exemple est illustré pour un Job shop « 3 tâches– 2 machines ».

De notre recherche bibliographique, nous avons étudié un article de Chandrasekaran et al. [08], paru récemment dans la littérature. Ils utilisent une nouvelle métaheuristique de type « Système immunitaire artificiel » notée SIA qu'ils l'ont comparé à une « recherche taboue » dans un problème de job shop avec une machine goulot. Les SIA est une génération récente de métaheuristiques qui ne datent que depuis 1999. Leur efficacité est prouvée dans la résolution de plusieurs problèmes d'ordonnancement, de l'intelligence artificielle...

La méthode appelée « Système Immunitaire Artificiel » et notée SIA, s'inspire du fonctionnement du système immunitaire humain [1,2]. Elle est classée comme un algorithme évolutif.

Nous avons organisé notre travail comme suit.

Dans le chapitre 1 sont introduits les problèmes d'ordonnancement et en particulier les problèmes de type Job Shop. Une méthode exacte de la résolution d'un job shop, une procédure par séparation et évaluation appelée communément procédure par « calage à gauche » est proposée dans le Nous proposons dans le chapitre suivant. Le chapitre 3 est consacré à une introduction succincte du système immunitaire humain. C'est une vue générale et elle sera un outil de résolution dans la suite. Nous détaillons les principaux fonctionnements du système immunitaire humain pour construire un « système immunitaire artificiel » qui fournira un algorithme de résolution du problème de type Job Shop dans le chapitre 4.

L'objectif du chapitre 5 est de présenter un algorithme, inspiré de la sélection clonale. La présentation et la description de notre application AISA-SJSS se fait dans le chapitre 6. Son implémentation est réalisée par nos soins. Nous l'avons testé sur un grand nombre de problèmes dont les données sont des « benchmarks » tirés de la littérature disponible en annexes B et C. Nos résultats ont été présentés sous forme de fenêtres, visualisés dans le texte de ce mémoire. Ils ont été discutés et comparés à ceux fournis par l'article cité. Des conclusions peuvent être faites. Le « SIA » constituera sûrement un outil efficace pour la résolution des problèmes d'ordonnancement. Une conclusion générale est énoncée. Ce mémoire dispose à sa fin de trois annexes. L'annexe A comprend des définitions de la biologie d'où s'inspirent ces algorithmes. L'annexe B est consacrée à une petite recherche sur la bibliothèque OR des benchmarks. Dans l'annexe C, d'autres résultats de l'application AIS-JSS sont fournis.

1.1. Introduction

Ordonnancer des tâches à exécuter, c'est attribuer les machines d'un ensemble $P = \{P_1, P_2, \dots, P_m\}$ et des ressources de $R = \{R_1, R_2, \dots, R_s\}$ aux tâches de l'ensemble $T = \{T_1, T_2, \dots, T_n\}$ dans l'ordre de terminer leur exécution sous des contraintes imposées.

Les problèmes d'ordonnancement sont caractérisés par ces trois ensembles où interviennent deux notions fondamentales : les tâches et les ressources.

Une *tâche* est un travail dont la réalisation nécessite un certain nombre d'unités de temps, sa durée, et d'unités de chaque ressource.

Une *ressource* est un moyen technique, financier ou humain, dont la disponibilité limitée ou non est connue à priori.

Il y a deux types de contraintes classiques : En tout instant, chaque tâche est à exécuter au plus sur une machine et chaque machine n'est capable d'exécuter qu'une tâche à la fois.

Dans la suite une machine est considérée comme une ressource.

Nous supposons aussi que les ressources ne constitueront aucun problème.

Notre problème d'ordonnancement peut être défini que par les ensembles de machines et de tâches.

1.2. Caractérisation des machines

Elles peuvent être en " *parallèles* ", faisant la même fonction, ou spécialisées dans l'exécution de certaines tâches (de l'anglais *dedicated* que l'on a traduit par " *dédiées* "). Dans ce dernier cas, les machines sont disposées, en général, en séries. On distingue trois types de machines parallèles dépendant de leur vitesse d'exécution.

Si toutes les machines ont la même vitesse d'exécution des tâches, les machines sont appelées " *identiques* " est le problème est noté par (P).

Si les machines diffèrent par leur vitesse d'exécution et la vitesse b_i de chaque machine est constante et ne dépend pas de l'ensemble des tâches, elles sont dites " *uniformes* " (Q).

Si les vitesses d'exécution des machines dépendent des tâches et sont différentes alors elles sont dites " *quelconques* " ou " *différentes* " (unrelated, R).

Dans le cas de machines " *dédiées* ", le plus souvent elles sont en série, il y a trois modèles ou types d'exécution de tâches: le flow shop, l'open shop et le job shop.

Dans un atelier où il y a « M » machines et « N » tâches, si les opérations élémentaires ne sont pas liées par un ordre particulier, on parle de problème *d'open shop*.

Dans ces problèmes, le nombre d'opérations de chaque tâche est égal au nombre de machines. Chaque opération utilisant une machine différente. Les opérations d'une même tâche ne sont pas soumises à des contraintes de précédence, par contre, il ne peut y avoir de chevauchements entre ces opérations.

Dans un *flow shop*, le nombre d'opérations de chaque tâche est égal au nombre de machines. Seulement il y a des contraintes de précédence de type chaîne entre les opérations d'une tâche: l'opération $(i, j + 1)$ ne peut commencer avant la fin de l'opération (i, j) .

Dans le cas du *Job shop*, il existe des contraintes de précédence entre les opérations d'une même tâche. Par contre, l'ordre d'utilisation de machines n'est pas le même pour toutes les tâches. Le nombre d'opérations peut être plus grand que le nombre de machines.

Un autre type d'ordonnancement existe, le *flow shop* hybride où des machines parallèles sont en série. C'est une organisation du processus de production en série. Les produits fabriqués passent dans un premier temps dans une première cellule (des machines en parallèles), puis dans une seconde, etc.

Dans de tel système, les espaces tampons ou Buffers entre les machines sont supposés de capacité illimitée et un job après sa fin d'exécution sur une machine doit attendre dans cet espace avant que son exécution ne commence sur une autre machine. Si les espaces tampons sont de capacité nulle, les jobs ne peuvent pas attendre entre deux machines consécutives et la propriété de " sans attente " est admise.

Le diagramme de la figure ci-dessous schématise les différents types de machines.

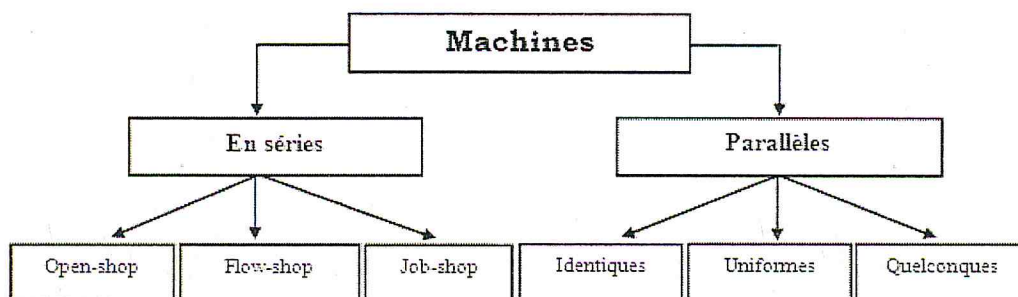


Figure 1.1 : Types de machines.

1.3. Les Tâches

En général une tâche $T_j \in T$ est caractérisée par les données suivantes.

Le vecteur des temps d'exécution $p_j = [p_{1j}, p_{2j}, \dots, p_{mj}]^T$, où p_{ij} est le temps qu'il faut pour la machine P_i pour exécuter la tâche T_j .

Dans le cas de machines " identiques " on a $p_{ij} = p_j, i = 1, 2, \dots, m$.

Si les machines de P sont uniformes alors $p_{ij} = p_j / b_i, i = 1, 2, \dots, m$ où p_j est un temps d'exécution standard (usuellement mesuré sur la machine la moins rapide) et b_i est le facteur vitesse d'exécution de la machine P_i .

Dans le cas de l'ordonnancement shop, le vecteur des temps d'exécution décrit les demandes en exécution de chaque tâche comprenant une " tâche". Pour une tâche $J_j = [p_{1j}, p_{2j}, \dots, p_{mj}]^T$ où p_{ij} est le temps d'exécution de la tâche T_{ij} sur la machine correspondante.

Pour fixer les idées, supposons qu'un travail n'est formé que d'une seule tâche ou opération.

Alors à chaque tâche T_j on associe :

1. son temps d'exécution p_j
2. son temps d'arrivée ou temps de début d'exécution au plus tôt (ou ready time) r_j , le temps où la tâche T_j est prête pour l'exécution. Si les temps d'arrivée sont les mêmes pour toutes les tâches de T , alors on supposera que $r_j = 0$ pour tout j .

3. sa date de fin d'exécution au plus tard (ou due date) d_j . Si la tâche termine son exécution après cette date, elle encourt une pénalité. Elle est appelée "date de fin souhaitée".
4. la date de fin impérative (deadline) \tilde{d}_j . Si la tâche termine son exécution après cette date, elle ne risque pas seulement une pénalité mais des problèmes surgiront. Soit l'atelier sera bloqué ou la machine tombe en panne etc.
5. un poids ou une priorité de la tâche (weight) w_j , qui exprime une urgence dans l'exécution de la tâche T_j .
6. Une date de fin d'exécution C_j (qui est en général une variable).
7. d_{e_j} : la date de début d'exécution de la tâche (qui est en général une variable).
8. son temps d'attente dans l'atelier $W_j(a_j)$

On définit le flow time de la tâche j , l'expression $F_j = c_j - r_j$ Il est aussi appelé "intervalle manufacturier ou temps d'atelier".

Une relation entre ces paramètres existe. $r_j \leq c_j = p_j + W_j \leq d_j \leq \tilde{d}_j$.

On suppose qu'à toutes les tâches sont assignées les ressources nécessaires pour leurs exécutions, quand elles commencent à être exécuter, ou qu'elles attendent pour passer sur une autre machine ou quand elles sont interrompues ou terminent leur exécution.

Le décalage $L_j = C_j - d_j$ (*Lateness*). Le temps total durant lequel la tâche j est autorisée à rester dans l'atelier après son exécution.

Le retard $D_j = T_j = \max \{ C_j - d_j, 0 \}$ (*Tardiness*)

L'avance $E_j = \max \{ 0, -L_j \}$ (*Earliness*)

L'indicateur de retard $U_i = 0$ si $c_i \leq d_i$ et $U_i = 1$ sinon.

Pour évaluer un ordonnancement on utilisera trois mesures de performance importante ou de critères d'optimalité.

1.4. Fonctions objectives d'un problème d'ordonnancement

Les fonctions économiques ou critères d'optimalité les plus utilisées font intervenir la durée totale de l'ordonnancement, le délai d'exécution, les retards de l'ordonnancement et le coût des stocks d'encours. La durée totale de l'ordonnancement notée C_{\max} est égale à la date d'achèvement de la tâche la plus tardive : $C_{\max} = \max c_j$. C'est la longueur de l'ordonnancement (Schedule length ou makespan).

Le Flow time moyen $\bar{F} = \frac{1}{n} \sum_{j=1}^n F_j$ ou le flow time moyen pondéré $\bar{F}_w = \frac{\sum_{j=1}^n w_j F_j}{\sum_{j=1}^n w_j}$

Le critère, flow time, $\sum_{i=1}^n w_i c_i$ permet d'estimer le coût des stocks d'encours.

En effet la tâche "i" est présente dans l'atelier entre les instants r_i et c_i , et donc les stocks dont elle a besoin doivent être disponibles entre ces deux dates; d'où le coût $\sum_{i=1}^n w_i (c_i - r_i)$ est égale à une constante près à $\sum_{i=1}^n w_i c_i$.

Dans beaucoup de problèmes, il faut respecter les délais, donc les dates au plus tard d_i ; on peut chercher à minimiser le plus grand retard $T_{\max} = \max T_i$, ou bien la somme des retards $\sum_{i=1}^n T_i$, ou encore la somme pondérée des tâches en retard $\sum_{i=1}^n w_i T_i$.

Le décalage maximum $L_{\max} = \max \{L_j\}$.

D'autres critères peuvent être utilisés.

Le retard moyen $D_{\text{moy}} = \frac{1}{n} \sum_{j=1}^n D_j$.

Le retard moyen pondéré $D_w = \sum_{j=1}^n w_j D_j / \sum_{j=1}^n w_j$.

Le nombre de tâches en retard $U = \sum_{j=1}^n U_j$, où $U_j = 1$ si $C_j > d_j$ et 0 sinon.

Le nombre de tâches en retard pondéré $U_w = \sum_{j=1}^n w_j U_j$.

• **Exemples :** P // C_{max} signifie : ordonnancement non-préemptif de tâches indépendantes de temps de service arbitraire, arrivant aux instants 0, sur des machines parallèles, identiques dans l'ordre de minimiser la longueur de l'ordonnancement.

J3/ $r_j / \sum_{j=1}^n C_j$ signifie: ordonnancement non-préemptif de longueurs de tâches arbitraires dans un job shop à trois machines, les tâches arrivent en des instants différents et l'objectif est de minimiser le flow time moyen.

O3 / pmtn, $r_j / \sum_{j=1}^n C_j$: signifie ordonnancement préemptif de longueurs de tâches arbitraires dans un open shop à trois machines, les tâches arrivent en des instants différents et l'objectif est de minimiser le flow time moyen.

Remarque : Pour une classification des problèmes d'ordonnancement, le lecteur est invité à lire le cours d'ordonnancement donné à l'université de Blida à l'URL :

http://www.univ-blida.dz/fac_sciences/mathematique/pdf/derbala/cours_ordo/chapter2_ordo.pdf

1.5. Atelier à cheminement multiple: le Job Shop

Dans la suite de ce mémoire, nous discuterons que du problème Job shop.

Nous rappelons que dans un atelier de type Job Shop, les tâches s'exécutent dans l'ordre de la disposition des machines. Elles peuvent ne pas s'exécuter sur toutes les machines. Une tâche peut revenir s'exécuter sur une même machine. Une tâche peut être composée de plus de « m » opérations élémentaires où « m » est le nombre de machines.

« ZT » est la dénomination d'une zone tampon ou zone de stockage.

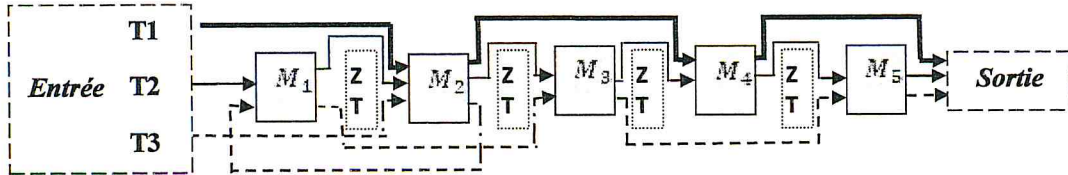


Figure 1.2: Cheminement de 3 tâches sur 5 machines dans un Job Shop.

1.6. Résolution d'un Job shop

Le problème étant combinatoire, il est improbable de trouver un algorithme polynomial pour le résoudre dans le cas général. Pour des problèmes de taille moyenne, une résolution exacte peut être élaborée. Elles recherchent un ordonnancement optimal minimisant un des critères présentés ou une combinaison de plusieurs critères. Les techniques utilisées sont les méthodes par séparation et évaluation, la programmation dynamique, la déduction mathématique, la théorie des jeux, la théorie des graphes. Dans le chapitre 2, pour résoudre un problème de job shop, nous présentons des différentes méthodes, et en particulier une méthode exacte par séparation et évaluation appelée « calage à gauche ».

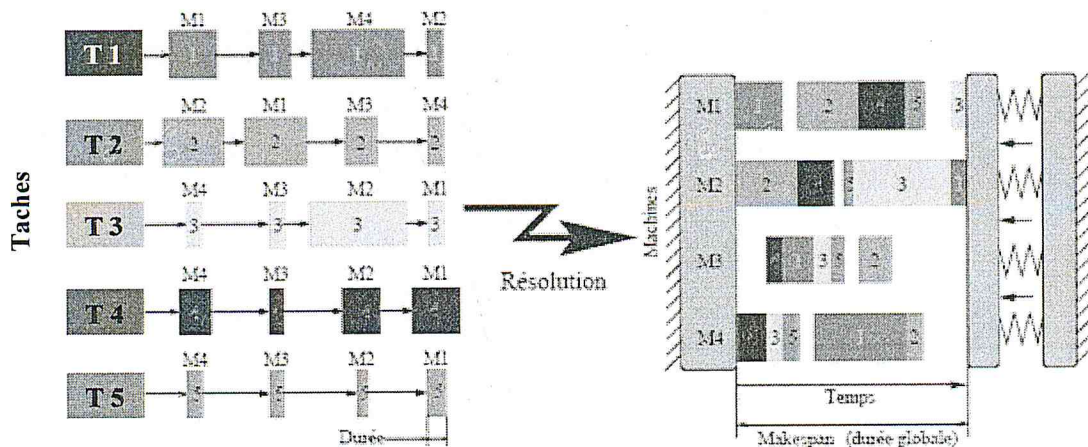


Figure 1.3: Résolution d'un Job shop simple par « calage à gauche », exemple du cours.

La résolution du problème consiste à déterminer un ordre d'exécution des opérations sur chaque machine de manière à minimiser la durée globale d'exécution de l'ensemble des opérations (le makespan).

De manière générale, le coût global de production est lié au temps nécessaire à la fabrication des différents produits (voir figure 1.3). En conséquence, l'objectif consiste à réduire la durée globale de fabrication, appelée « makespan »

1.7. Conclusion :

Pour des problèmes de grandes tailles, les méthodes exactes sont en général inefficaces. Si on se contentera d'une solution approchée, les méthodes de résolution sont stochastiques. On peut aussi utiliser des **Méta-heuristiques et les heuristiques**. Ces méthodes sont souvent dynamique ou dites en-ligne. Leur inconvénient est qu'elle ne garantisse pas de solution optimale. Elles permettent de réduire considérablement le temps de recherche pour trouver une solution acceptable. Une nouvelle et récente Méta-heuristique, le système immunitaire artificiel, de l'anglais « Artificial Immune System » sera aussi introduite au chapitre 4.

Chapitre 2 :

La méthode de SEP

2.1. Introduction

Les méthodes arborescentes (Branch and Bound Methods) sont des méthodes exactes d'optimisation qui pratiquent une énumération intelligente de l'espace des solutions. Il s'agit d'énumération complète améliorée. Elles partagent l'espace des solutions en sous ensembles de plus en plus petits, la plupart étant éliminés par des calculs de bornes avant d'être construits explicitement.

Appliquées à des problèmes NP-difficiles, ces méthodes restent bien sur exponentielles, mais leur complexité en moyenne est bien plus faible que pour une énumération complète. Elles peuvent pallier le manque d'algorithmes polynomiaux pour des problèmes de taille moyenne. Pour des problèmes difficiles de grande taille, leur durée d'exécution est très grande et il faut se retourner vers des heuristiques ou solutions approchées.

Pour un Job shop, on peut inventer plusieurs méthodes arborescentes. Cependant elles auront trois composantes communes :

- Une règle de séparation
- une fonction d'évaluation
- une stratégie d'exploration

2.2. Résolution d'un problème Job shop de trois pièces à trois machines (Exemple)

Pour réaliser la fabrication de trois pièces, on dispose de leur gamme de fabrication.

La pièce i demande q_i opérations et la $j^{\text{ème}}$ opération de la pièce i occupe τ_{ij} unités de temps de la machine M_{ij} . Les durées d'usinage sont très longues par rapport aux durées de transport et de réglage et on négligera ces dernières. Cette méthode ne permet de résoudre que des problèmes de taille raisonnable. Le problème qui consiste à placer les opérations sur les machines de telle sorte que :

- L'opération j du produit i soit terminée avant le début $j+1$.
- Une machine ne traite qu'une seule opération à la fois
- On minimise la durée totale (l'opération qui se termine la dernière se termine le plus tôt possible)

Les gammes sont fournies dans le tableau suivant:

Opération pièce	O_1	O_2	O_3	O_4
1	3	1	2	1
2	1	2	1	3
3	2	1	/.	/.

Tableau2.1: Temps d'exécution des pièces.

Opération pièce	O_1	O_2	O_3	O_4
1	1	2	3	3
2	2	1	3	2
3	3	2	/.	/.

Tableau2.2: Les opérations.

Les bornes minorantes sont calculées par relaxation en ignorant, initialement, les contraintes dues au fait qu'une machine ne peut effectuer qu'une seule et unique opération à la fois. Ces contraintes seront intégrées progressivement au fur et à mesure des séparations jusqu'à l'obtention de solutions réalisables qui seront optimales pour le sous-ensemble de solutions auquel elles appartiennent.

En fin d'exploration, la meilleure solution réalisable trouvée sera optimale. L'ensemble des solutions candidates A_0 est l'ensemble de tous les ordres possibles de toutes les opérations à exécuter sur les différentes machines.

Pour trouver un minorant à la meilleure solution possible de cet ensemble, on ignore les contraintes dues aux machines en faisant comme si on avait autant d'exemplaires de chaque machine que nécessaire pour réaliser la solution "au plus tôt" pour chaque pièce, ce qui donne l'ordonnancement non réalisable suivant représenté sur un diagramme où chaque ligne correspond à une pièce et chaque case à la $j^{\text{ème}}$ opération de la tâche i sur la machine (k).

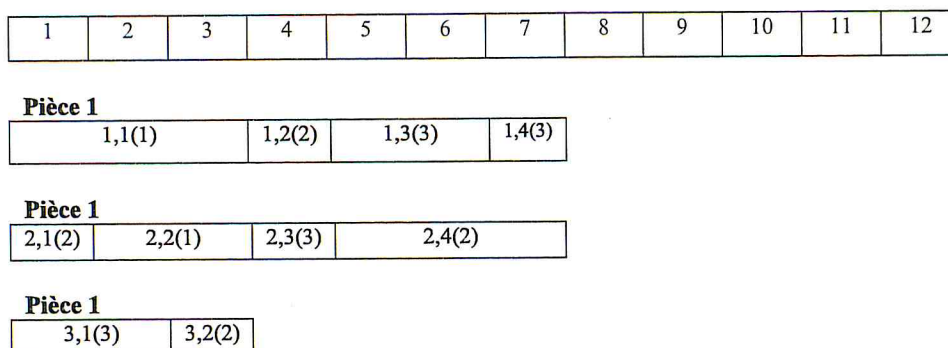


Figure 2.1 : Diagramme " pièces " calé au plus tôt du sous ensemble A_0 .

La plus longue gamme ayant une durée totale de sept (7) unités de temps, 7 est un minorant pour la durée totale pour l'ensemble des solutions du problème d'où $b_0 = 7$. La solution ci dessus serait optimale si elle était réalisable. Elle comporte des conflits sur l'utilisation des machines. A l'instant 1, un conflit survient entre 1,1 (1) (la première opération du produit un) et 2,2(1) (la deuxième opération du produit deux) sur l'utilisation de la machine 1.

Nous devons séparer l'ensemble de solutions A_0 en deux sous ensembles de solutions de telle sorte que toute solution réalisable se retrouve obligatoirement dans l'un des deux et que les solutions non réalisables qui contenaient ce premier conflit ne s'y retrouvent plus.

Pour cela, on partitionne A_0 en deux sous ensembles complémentaires

A_1 où 1,1(1) est placée sur la machine 1 avant 2,2 et

A_2 où 2,2 est placée sur la machine 1 avant 1,1.

Avant de séparer A_1 ou A_2 , nous allons calculer les nouvelles bornes les concernant de manière à pouvoir appliquer une procédure par séparation et évaluation progressive en séparant toujours le sous ensemble de meilleur borne.

Placer 1,1 avant 2,2 revient à déplacer le début de 2,2 de 2 unités et donc à déplacer la fin du produit deux de 7 à 9 unités. D'où $b_1 = 9$.

Placer 2,2 avant 1,1 revient à déplacer le début de 1,1 de 3 unités et donc de déplacer la fin du produit un de 7 à 10 unités. D'où $b_2 = 10$.

La meilleure borne est fournie par A_1 que nous allons séparer en cherchant le premier conflit-machine sur son diagramme pièces calé au plus tôt.

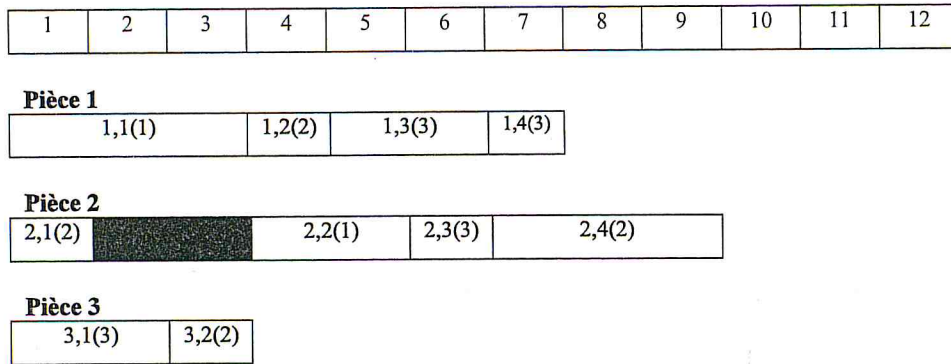


Figure 2.2 : Diagramme " pièces " calé au plus tôt du sous ensemble A_1 .

Ici, le diagramme d' A_2 n'est pas représenté.

Un conflit survient à l'instant 5 entre 1,3(3) et 2,3(3) pour l'utilisation de la machine 3.

A_1 sera séparé en deux sous ensembles A_3 , où 1,3 (3) est placée avant 2,3(3) et A_4 , où 2,3(3) est placée avant 2, 3(3).

Placer 1,3(3) avant 2,3(3) revient à déplacer 2,3(3) d'une unité de temps et donc de déplacer la fin du produit 2 de l'instant 9 à 10.

Placer 2,3(3) avant 1,3(3) revient à déplacer 1,3(3) de deux unités de temps et donc de déplacer la fin du produit 1 de l'instant 7 à 9.

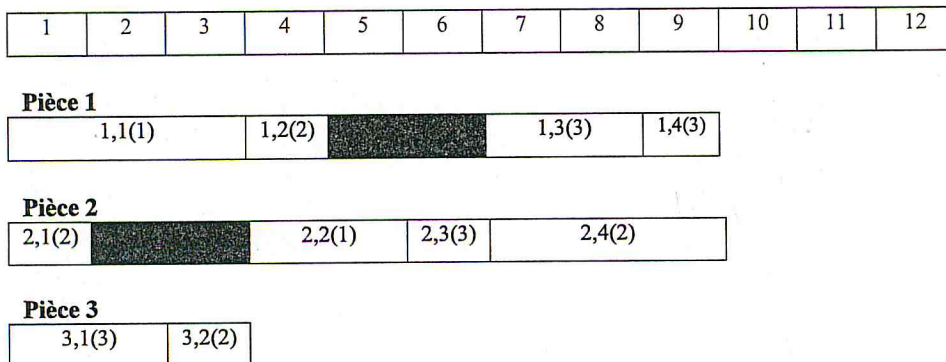


Figure 2.3 : Diagramme " pièces " calé au plus tôt du sous ensemble A_4 .

Ce diagramme ne comporte plus de conflit pour l'utilisation des machines. C'est une solution réalisable pour le problème posé, comme sa durée totale est strictement

inférieure à toutes les bornes des sous ensembles que nous n'avons pas encore séparés. C'est l'unique solution optimale parmi les solutions calées au plus tôt.

Remarque : Si on programme cette méthode, on n'utilise pas les diagrammes " pièces " mais les dates au plus tôt de toutes les opérations sur le graphe potentiel-tâche comportant les contraintes de précédence dues aux gammes puis celles ajoutées progressivement à chaque séparation pour tenir compte des conflits dus aux machines. La borne étant la date au plus tôt de la tâche fictive de fin de réalisation.

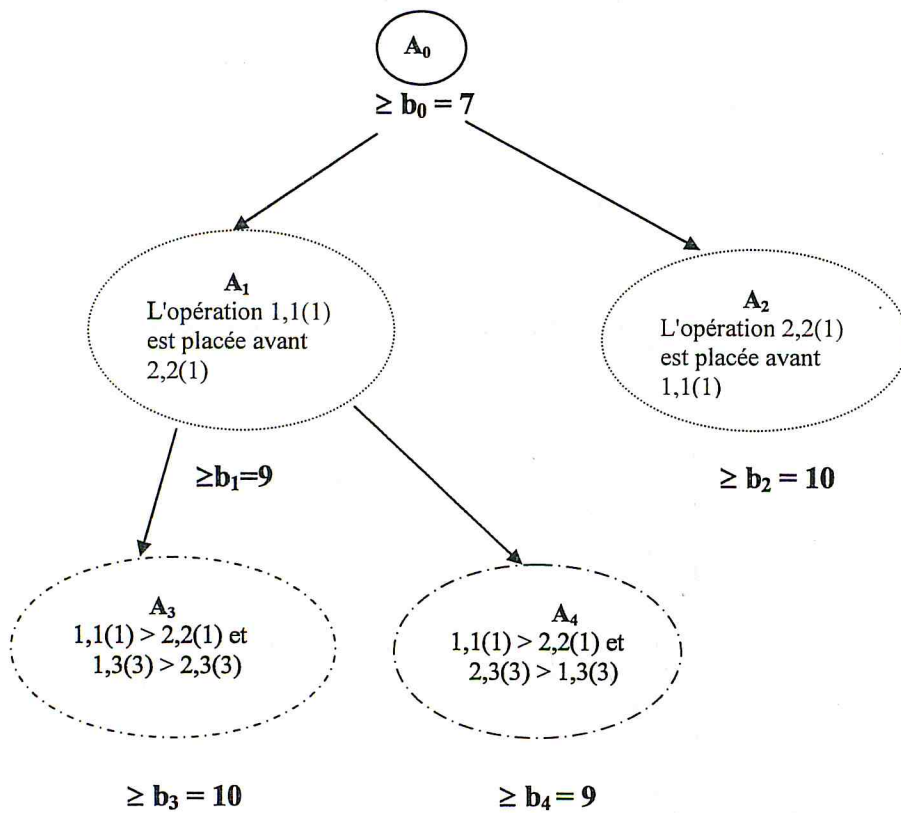


Figure 2.4: Arborescence des séparations.

3.1. Introduction

Le système immunitaire d'un organisme est un mécanisme de défense contre les pathogènes, tels que les virus, les bactéries, les parasites, les cellules cancéreuses, certaines particules ou molécules « étrangères » (dont certains poisons).

Le mot « immunité » vient du latin *immunitas* qui désigne un « exempté de charge ».

Citons quelques définitions indispensables.

Un pathogène est tout corps peut provoquer une maladie.

Un virus est un agent infectieux très petit.

Une bactérie est un micro-organisme dont la cellule ne comporte pas de noyau.

Un parasite est un être vivant microscopique ou non, pouvant pénétrer dans l'organisme ou vivre à sa surface.

On dénombre plusieurs variantes de systèmes immunitaires parmi les espèces animales, et parfois un même organisme peut accueillir plusieurs systèmes immunitaires.

Il y a lieu de remarquer que le cerveau humain possède son propre système immunitaire, distinct de celui du reste du corps.

Nous introduisons le système immunitaire humain.

Les principaux agents du système immunitaire sont les cellules immunitaires appelées leucocytes (ou globules blancs) produites par des cellules souches, au sein de la moelle osseuse.

Une Cellule est l'élément constitutif fondamental de tout être vivant.

Une cellule souche est capable de se différencier en chacun des types de cellules sanguines: leucocyte (globule blanc), globule rouge ou plaquette.

Chapitre 3 : Le Système Immunitaire Humain

Il existe deux types de mécanismes de défense :

a. Les mécanismes de défense non-spécifique ou innée ou naturelle, comme la protection de la peau et les muqueuses, l'acidité gastrique, les cellules phagocytaires ou les larmes ;

b. Les mécanismes de défense spécifique, comme l'action dirigée des lymphocytes et la production d'anticorps spécifiques.

Un lymphocyte est un globule blanc du sang et du tissu lymphoïde responsable de l'immunité spécifique.

On appelle réponse immunitaire l'activation des mécanismes du système immunitaire face à une agression de l'organisme. L'ensemble de ces systèmes (y compris chez l'homme la vaccination) permet la résilience immunitaire.

Elle est une notion qui recouvre la somme des mécanismes efficaces de défense d'un organisme vis-à-vis d'un agent pathogène (du grec *pathos* : malheur).

3.2. Les agressions du SIH :

L'organisme se défend contre les agressions, des processus qui ont pour conséquence de détruire des êtres vivants.

Ces agressions peuvent revêtir différentes formes :

- Les agressions dites physico-chimiques :
 - Mécaniques : frottements, chocs, chutes, etc.
 - A Facteurs climatiques : altitude, changement brusque de température, rayonnements, sécheresse, poussières, etc.
 - Agression par des agents chimiques : acides, bases, etc.
- Les agressions par d'autres êtres vivants (infection):

Un organisme constitue en effet pour un autre organisme un endroit idéal pour ses propres cellules et aussi pour un certain nombre de micro-organismes qui pourraient y proliférer.

3.3. Mécanismes de défense non-spécifique « proliféré »

L'immunité non spécifique est l'action immédiate, qui fait intervenir des cellules responsables de la phagocytose. Il s'agit alors d'une infection, et les agresseurs dans ce cas peuvent être : des virus, des bactéries, des champignons, des levures, etc....

3.3.1. Cellules phagocytaires

Les leucocytes phagocytaires capturent puis digèrent les microbes.

Elles sont à :

- 70% des granulocytes neutrophiles
- 5% des monocytes, qui se transformeront par la suite en macrophages,
- 1,5% des granulocytes éosinophiles.

Pour leur définition, voir Annexe A.

Avant qu'un agent infectieux puisse pénétrer notre corps, il faut qu'il traverse une série de barrières biochimiques et physiques.

3.3.2. Barrières du SIH

• La *peau* (ou tissu externe) est le premier, le plus grand et le plus important élément du système immunitaire. Elle prévient l'entrée de la plupart des corps étranger.

Les personnes qui perdent trop de peau, comme dans le cas des brûlures, peuvent succomber à des infections.

• Les *muqueuses* (ou tissu continus) qui recouvrent les voies oropharyngiennes et digestives, les voies respiratoires et urogénitales constituent également une barrière physique.

En surface de certaines muqueuses, un film de mucus animé par les battements de cils vibratiles permet de fixer, enrober puis évacuer vers l'extérieur la plupart des particules ou être vivants intrus.

Les cellules très étroitement juxtaposées sont imperméables à la plupart des agents infectieux.

- Les cellules possèdent enfin divers systèmes « *passifs* » de défense chimiques et biochimiques (enzymes, acides gras, acide lactique, flore intestinale, pH du corps...).
- Il existe aussi des barrières naturelles qui isolent tous nos organes de l'environnement pour nous protéger des microbes pathogènes.

3.3.3. Réaction inflammatoire

Pour les détails sur les définitions, les notions citées ci-dessous, le lecteur est invité à consulter le livre de Janeway d'immunobiologie [4].

1. L'inflammation est caractérisée par quatre (04) signes :

- ♦ La chaleur et la rougeur sont dues à la vasodilatation (augmentation) des capillaires et au ralentissement de la circulation du sang
- ♦ La douleur est due à la pression des fibres nerveuses
- ♦ L'œdème dû à l'exsudation du plasma

2. La réaction inflammatoire :

- ♦ réaction vasculaire
- ♦ réaction cellulaire

3. Les phagocytes et lymphocytes T cytotoxiques :

- ♦ les leucocytes phagocytaires
- ♦ les lymphocytes T cytotoxiques

4. Les protéines antimicrobiennes (défense humorale).

En surface de certaines muqueuses, un film de mucus animé par les battements de cils vibratiles permet de fixer, enrober puis évacuer vers l'extérieur la plupart des particules ou être vivants intrus.

Les cellules très étroitement juxtaposées sont imperméables à la plupart des agents infectieux.

Les cellules possèdent enfin divers systèmes « passifs » de défense chimiques et biochimiques (enzymes, acides gras, acide lactique, flore intestinale, pH du corps...).

Il existe aussi des barrières naturelles qui isolent tous nos organes de l'environnement pour nous protéger des microbes pathogènes.

3.3.3. Réaction inflammatoire

Pour les détails sur les définitions, les notions citées ci-dessous, le lecteur est invité à consulter le livre de Janeway d'immunobiologie [4].

1. L'inflammation est caractérisée par quatre (04) signes :

La chaleur et la rougeur sont dues à la vasodilatation (augmentation) des capillaires et au ralentissement de la circulation du sang

- ♦ La douleur est due à la pression des fibres nerveuses
- ♦ L'œdème dû à l'exsudation du plasma

2. La réaction inflammatoire :

- ♦ réaction vasculaire
- ♦ réaction cellulaire

3. Les phagocytes et lymphocytes T cytotoxiques :

- ♦ les leucocytes phagocytaires
- ♦ les lymphocytes T cytotoxiques

4. Les protéines antimicrobiennes (défense humorale).

3.3.4. Système lymphatique

Les globules blancs passent la majeure partie de leur temps hors du système circulatoire, et patrouillent dans le liquide interstitiel des cellules où se déroulent la plupart des luttes contre les agents pathogènes. Certains macrophages résident en permanence dans les organes (poumons, foie) ou dans le système lymphatique.

Le système lymphatique comprend divers organes (thymus, moelle osseuse, rate, amygdales, appendice et ganglions lymphatiques) qui jouent un rôle important dans le système immunitaire.

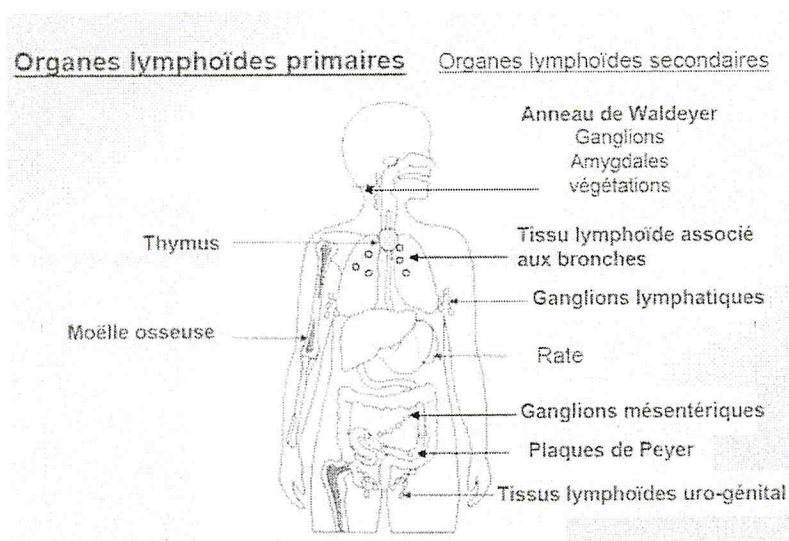


Figure 3.1 : Organes et tissus lymphoïdes

Les capillaires lymphatiques absorbent une partie du liquide interstitiel qui baigne les tissus. Le liquide, alors appelé lymph, finit par retourner dans la circulation sanguine via le canal thoracique. Sur son parcours, la lymphe traverse de nombreux ganglions lymphatiques dans lesquels tout agent pathogène rencontre des globules blancs.

3.4. Mécanismes de défense spécifique

L'immunité adaptative est la forme d'immunité qui est stimulée par des microbes, et possède une fine spécificité pour des substances étrangères, ainsi qu'elle répond

Chapitre 3 : Le Système Immunitaire Humain

plus efficacement contre les expositions successives d'un microbe. Elle est caractérisée par la reconnaissance du soi et du non-soi.

Le soi d'un individu est défini par des récepteurs sur la membrane de ses cellules, ces récepteurs constituent des marqueurs de son identité ou des marqueurs du soi (carte d'identité).

Le non-soi d'un individu est défini par des récepteurs cellulaires ou toute autre molécule différente du soi et ainsi reconnus comme étrangers par notre organisme. Il déclenche une réaction immunitaire.

3.5. Le système immunitaire *cellulaire*

Il s'occupe des cellules infectées par des virus, bactéries, et les cellules cancéreuses. L'action s'effectue via les cellules T, aussi appelées *lymphocytes T* (*T* parce que ces cellules mûrissent dans le thymus après leur naissance dans la moelle osseuse).

On distingue deux grandes familles de Lymphocytes T :

- les lymphocytes T cytotoxiques (T_C) reconnaissent les cellules infectées en utilisant des récepteurs pour tester la surface des autres cellules. Si elles reconnaissent une cellule infectée, elles peuvent la détruire ainsi que le virus qu'elle contient.
- les lymphocytes T *Helper* (T_H) qui interagissent avec les macrophages (qui ingèrent les substances dangereuses) et produisent également des cytokines (interleukine) induisant la prolifération des Lymphocytes *B* et *T*.

Aux lymphocytes T s'ajoutent aussi les cellules dites « NK » pour Natural Killers. Ces cellules sont impliquées dans une réponse à mi-chemin entre spécifique et non spécifique, selon les situations.

3.6. Le système immunitaire *humoral*

Il agit contre les bactéries et les virus dans les liquides du corps humain, tels que le sang en sécrétant des substances susceptibles dites immunoglobulines aussi appelées *anticorps*, d'aider à la destruction des agents pathogènes- historiquement. Le sang et la lymphe étaient nommés les *humeurs* du corps.

3.7. Conclusion

La fonction physiologique du système immunitaire est de protéger les individus contre les infections. Mais il peut poser des problèmes soit en excès soit en défaut.

En effet si le système immunitaire s'attaque aux cellules de l'organisme qui ne sont pas pathologiques (par mauvaise reconnaissance), il va alors se créer une maladie auto-immune qui va se caractériser par une inflammation continue de certains tissus ou par la nécrose complète de certains tissus (par exemple le diabète).

S'il y a un défaut du système immunitaire, dans ce cas les pathogènes ou les cancers pourront se développer plus aisément.

Chapitre 4 :

Le Système Immunitaire Artificiel

4.1. Introduction

Timmis en 2000 [5] , a définie les Systèmes Immunitaires Artificiels noté **SIA** comme des systèmes informatiques basés sur des métaphores du système immunitaire naturel.

Et en 2002, Timmis et de Castro [5] ont proposé une nouvelle définition des **SIAs**. Ces chercheurs considèrent les **SIAs** comme des systèmes adaptatifs, s'inspirant des théories de l'immunologie, ainsi que des fonctions, des principes et des modèles immunitaires, afin d'être appliqués à la résolution de.

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 différents problèmes en particulier le problème d'ordonnancement de type Job shop.

L'apprentissage des cellules immunitaires consiste à acquérir ou à modifier une représentation d'un environnement (pathogènes, antigène,...), de façon à permettre à celui-ci des interactions efficaces ou de plus en plus efficaces avec ces cellules.

Après une réaction immunitaire à un antigène donné, quelques ensembles de cellules seront dotés d'une grande durée de vie afin de fournir des réactions immunitaires plus rapides et plus puissantes à des futures infections par le même antigène ou à des antigènes semblables, ce qu'on appelle la mémorisation des cellules immunitaires.

Les algorithmes de **SIA** peuvent être divisés en trois grandes parties, chacune d'elles s'inspirant d'un comportement ou d'une théorie du système immunitaire biologique: la sélection clonale, la sélection négative et positive, les réseaux immunitaires et la théorie du danger.

Pour notre étude nous nous intéressons à la sélection clonale.

4.2. La maturation de l'affinité

L'affinité d'un anticorps pour un antigène est la force de liaison entre un seul site de liaison de l'anticorps et l'antigène.

Lors d'une réponse immunitaire, avant que les lymphocytes B (cellules B) ne se transforment en plasmocytes (cellules productrices d'anticorps), ils subissent des mutations. Seules les cellules ayant les plus hautes affinités avec l'antigène sont sélectionnées.

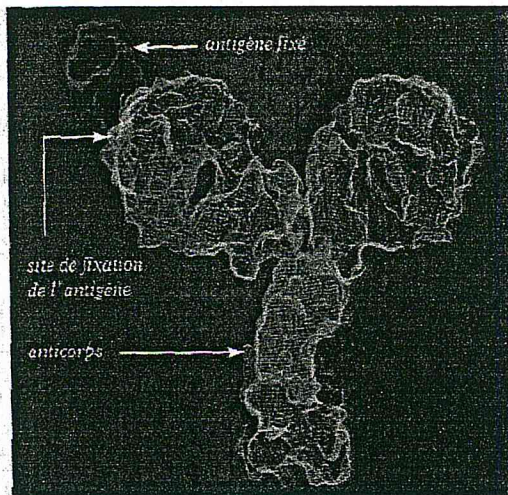


Figure 4.1 : Affinité du complexe Anticorps-Antigène

4.3. Représentation et affinité

Afin de pouvoir être utilisés en informatique, les différents composants du système immunitaire biologique, plus particulièrement les cellules immunitaires et les antigènes, doivent être modélisés sous forme numérique, informatique.

Par ailleurs le système immunitaire biologique a pour base de fonctionnement l'identification des antigènes et l'affinité entre les différents récepteurs des cellules (RCB et RCT) et ces antigènes qu'ils soient complets ou décomposés (présentés par les CPA).

Pour que les cellules modélisées puissent reconnaître les "antigènes", cette notion d'affinité est reprise et appliquée aux cellules informatiques.

Les cellules **B** et **T** portent à leur surface un grand nombre de récepteurs, chacun n'est capable de reconnaître qu'un seul type d'antigène et tous les récepteurs d'une même cellule sont identiques. Dans le **SIA dont le domaine d'application est l'informatique**, on ne parlera plus de cellules **B** et **T** ni de leurs récepteurs respectifs mais d'anticorps en général par soucis de facilité de modélisation et de représentation.

4.4. Modèle de la sélection clonale [6] :

La sélection clonale est la théorie expliquant comment le système immunitaire interagit avec les antigènes. Cette théorie est applicable aux cellules **B** ainsi qu'aux cellules **T**.

La seule différence est que les cellules **B** subissent une hyper-mutation somatique durant leur prolifération contrairement aux cellules **T**.

Grâce à ce procédé, le corps humain est capable de contrer un très grand nombre d'éléments externes. Les SIA s'inspirent de cette théorie, Mais vu que seules les cellules **B** sont capables de muter pour optimiser la réponse immunitaire, seules ces cellules nous intéressent.

Cette optimisation est due au fait que les cellules **B** une fois en contact avec l'antigène, elles se multiplient et donnent plusieurs clones et chaque clone subit une mutation.

Cette mutation sert à trouver des clones de la cellule mère possédant une plus grande affinité avec l'antigène. Lors des mutations successives, le risque d'être affronté à un problème d'optimum local ou l'on ne peut plus trouver de meilleur clone même en passant des générations de cellules.

Le corps humain évite ce problème en ajoutant à ces clones des éléments nouvellement créés, qui n'ont aucune relation avec la cellule mère. Cet ajout d'éléments "aléatoires" permet de résoudre le problème en faisant évoluer les nouveaux éléments s'ils possèdent une plus grande affinité avec l'antigène.

Cette théorie a engendré un développement par Darmoul et al. [7] de plusieurs approches immunitaires artificielles appliquées à l'ordonnancement.

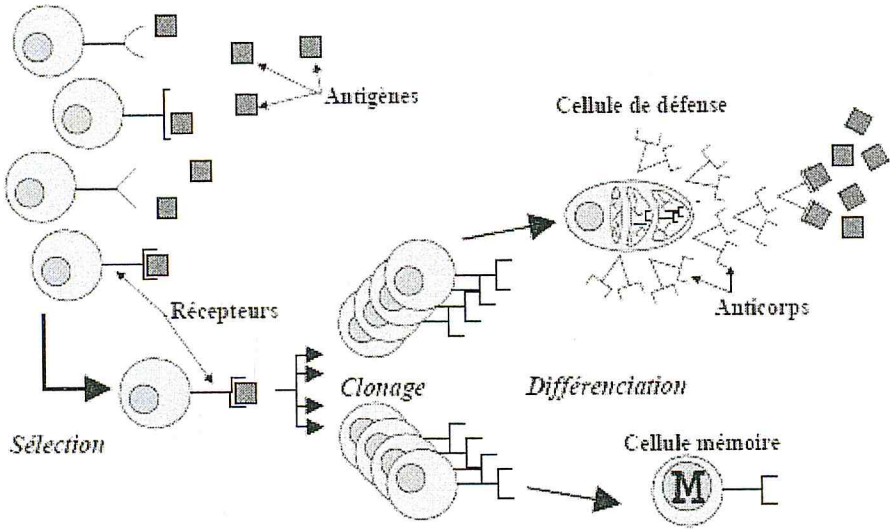


Figure 4.2 : La sélection clonale

4.5. L'algorithme de la sélection clonale :

Il existe plusieurs algorithmes de la sélection clonale, nous nous intéressons dans la suite de la thèse à celui représenté dans les deux figures suivantes:

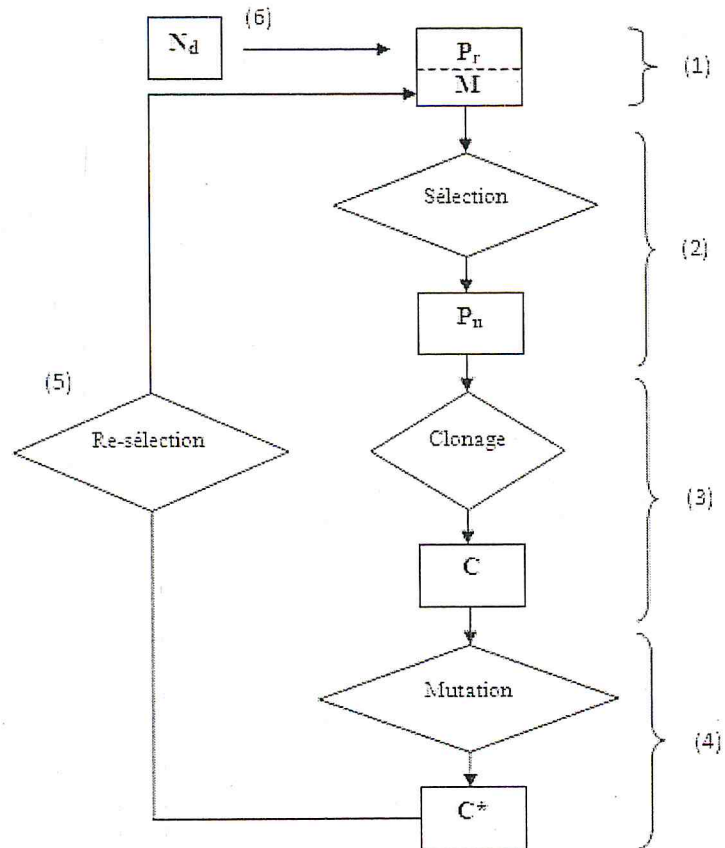


Figure 4.3 [6]: L'organigramme de la sélection clonale

Algorithme de la sélection clonale:
(1) Générer aléatoirement un ensemble (P) de solutions candidates, composé de d'un sous ensemble des cellules mémoires (M) et un autre du reste des cellules mortes (P _r) de la population (P=M+P _r) ;
(2) Déterminer (sélectionner) les n anticorps meilleurs de la population (P _n), Suivant le principe d'affinité.
(3) Reproduire (cloner) les n anticorps de la population augmente temporairement la population des clones (C). la taille de la population des clones est une fonction croissante dépend de la fonction affinité.
(4) Appliquer les processus de mutations à la population des clones. Une population des anticorps matures sera générer (C*);
(5) Re-sélectionner les anticorps améliorés de C* pour construire l'ensemble de la mémoire M. (une partie d'anticorps de la population P peut être remplacé par une autre d'anticorps améliorées de C*.
(6) Remplacer d anticorps par d'autres nouveaux (Introduction de la gamme). Les cellules de valeurs d'affinité minorants sont plus probables d'être remplacées.
(7) Recommencer le cycle en procédant à (2).

Tableau 4.1 [6]: Algorithme de la sélection clonale

4.6. Conclusion :

Les principales différences entre tous les algorithmes de la sélection clonale sont les méthodes utilisées dans la génération aléatoire des anticorps, la mutation et l'affinité entre les anticorps et les antigènes. Nous donnons plus d'explications dans le chapitre suivant.

Les algorithmes de sélection clonale sont le plus souvent utilisés : dans des applications d'optimisation vue que les cellules **B** deviennent de plus en plus affines aux antigènes, des applications de détection d'intrusions également où l'on ne peut répertorier tous les éléments indésirables et où ces éléments sont extrêmement variés, ainsi que d'autres applications comme la reconnaissance de caractères.

Chapitre 5:

Application du SLA pour résoudre

un problème du Job Shop

5.1. Introduction

Des algorithmes du SIA sont souvent utilisés dans des applications d'ordinateur et de réseaux de sécurité, de détection d'erreurs et de fautes, d'optimisation, d'analyse de données etc. Des résolutions du problème de Job Shop connu pour être difficile dans le cas d'un nombre de tâches et de machines élevé ont été élaborées en utilisant des recherches de type taboues. De notre recherche bibliographique sur le thème des méta-heuristiques de type SIA, nous exposons un algorithme élaboré par Chandrasekaran & al. [8]. Il est déroulé sur un grand nombre de problèmes et comparé à la méthode « recherche taboue avec étranglement » [9]. Il a fourni des écarts de valeur du makespan très réduits. Les outils de résolution et l'analogie avec les problèmes d'optimisation combinatoire, plus particulièrement les problèmes d'ordonnement de type Job shop seront détaillés.

5.2. Le Système Immunitaire Artificiel (SIA) :

Il s'avère que les mécanismes opérationnels du système immunitaire sont très efficaces du point de vue calculs [10]. Le système immunitaire artificiel est élaboré sur les principes suivants de la *sélection clonale* et de *l'affinité de la maturation*.

5.2.1. Le principe de la sélection clonale

Chaque ordonnancement est représenté par une permutation de tâches. Il correspond aussi à un anticorps et a une valeur du makespan qui est associée à une valeur d'affinité de cet anticorps.

La valeur du makespan d'un ordonnancement est inversement proportionnelle à celle de l'affinité d'un anticorps. Elle est donnée par la formule [11,12]:

$$\text{Affinité}(p) = \frac{1}{\text{makespan}(p)}$$

De cette relation, une valeur minorée du makespan nous fournira une valeur majorée de l'affinité. Ultérieurement, le clonage des anticorps est fait directement et proportionnellement aux valeurs des fonctions affinité. D'où, dans la génération d'une nouvelle population clonée un grand nombre de clones d'anticorps auront des valeurs inférieures du makespan que ceux à valeurs hautes du makespan dans la première population.

Chapitre 5 : Application du SIA pour résoudre le problème du Job Shop

La fonction « affinité » doit être une fonction du makespan, pour pouvoir déterminer le nombre de clones qui seront proliférés.

Le nombre de clones le plus usité [11,13] est donné par le rapport du produit de la taille d'une population par l'affinité d'un individu sur la somme des affinités des individus, soit :

$$\text{Nbr-clone} = \left\lfloor \frac{\text{taille_population} \times \text{affinité d'individu}}{\text{affinité totale}} \right\rfloor$$

Le mécanisme de sélection clonale est utilisé pour trouver de nouveaux ordonnancements réalisables en re-planifiant des activités, en permutant, en réaffectant et/ou en réordonnant des tâches (la mutation).

La procédure du clonage est représentée par le tableau suivant :

Le processus de clonage
<i>Pour chaque anticorps</i> Calculer le nombre de clones Générer les copies correspondantes (correspond)

Tableau 5.1: Procédure de clonage

5.2.2. Le principe d'affinité de maturation

Le principe d'affinité de maturation consiste en deux méthodes appelées « mutation et production des récepteurs ».

A. La *mutation* [14]: elle est utilisée pour la génération des clones. C'est une procédure en deux phases.

(a) *La mutation inverse* (de l'anglais: inverse mutation): pour une permutation de tâches s , soient i et j deux positions choisies de façon aléatoire dans s (voir figure 5.1).

Un voisin de s noté $V(s)$ est obtenu en inversant la séquence des tâches (jobs) entre les deux positions i et j .

Si $s = (\sigma_1, \sigma_2, \dots, i, \sigma_{i+1}, \dots, \sigma_{j-1}, j, \dots, \sigma_n)$
 Alors $V(s) = (\sigma_1, \sigma_2, \dots, j, \sigma_{j-1}, \dots, \sigma_{i+1}, i, \dots, \sigma_n)$

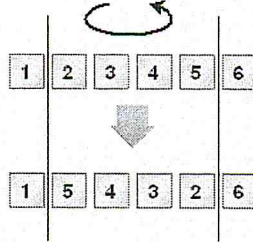


Figure 5.1 : La mutation inverse.

Si la valeur du makespan de la séquence mutée (après application de la mutation inverse) est plus petite que celle de la permutation originale (qui correspond à une génération d'un clone d'un anticorps), alors la permutation mutée sera prise en considération et deviendra la nouvelle permutation originale. Sinon, la séquence mutée sera nouvellement mutée par une transposition aléatoire de deux tâches.

(b) *La mutation de simple échange de deux positions* (pairwise interchange mutation):
 Nous donnons une séquence s , soient i et j deux positions choisies aléatoirement dans la séquence s (voir figure 5.2).

Un voisin de s est obtenu en échangeant les tâches en positions i et j .

Si $s = (\sigma_1, \sigma_2, \dots, i, \sigma_{i+1}, \dots, \sigma_{j-1}, j, \dots, \sigma_n)$
 Alors $V(s) = (\sigma_1, \sigma_2, \dots, j, \sigma_{i+1}, \dots, \sigma_{j-1}, i, \dots, \sigma_n)$

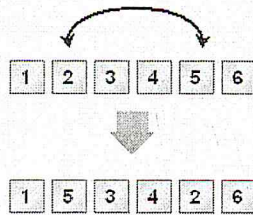


Figure 5.2: La mutation de simple échange de deux positions

Si la valeur du makespan de la séquence mutée (après application de la mutation par transposition de tâches) est plus petite que celle de la permutation originale, nous enregistrons la permutation mutée à la place de l'originale.

Dans le cas où l'algorithme ne peut pas produire un meilleur ordonnancement après les deux procédures de mutation, alors la séquence originale est maintenue.

Le tableau suivant regroupe les deux phases de la mutation en une procédure de mutations.

Le processus de mutations
<p><i>Pour chaque clone d'un anticorps</i></p> <p>Début</p> <p>Appliquant le processus de mutation inverse pour générer une nouvelle permutation de tâches (p_1):</p> <p>Sélectionner cette permutation (p_1):</p> <p>Déterminer sa longueur d'ordonnement (Makespan):</p> <p>Si Makespan [permutation (p_1)] < Makespan [permutation clone] alors</p> <p style="padding-left: 20px;">Clone := nouvelle permutation (p_1)</p> <p>Sinon</p> <p>Début</p> <p>Appliquer le processus de « mutation de simple échange de deux positions » sur la permutation (p_1)</p> <p>pour générer une nouvelle permutation de tâches (p_2)</p> <p>Sélectionner la permutation (p_2):</p> <p>Déterminer sa longueur d'ordonnement (Makespan):</p> <p>Si Makespan [permutation (p_2)] < Makespan [permutation clone] alors</p> <p style="padding-left: 20px;">Clone := nouvelle permutation (p_2) :</p> <p>Sinon</p> <p style="padding-left: 20px;">Clone := Clone :</p> <p>Fin,</p> <p>anticorps := Clone :</p> <p>Fin</p>

Tableau 5.2 : Procédure de mutation

B. Production des *récepteurs* [15] : après les processus de clonage et de mutation, un pourcentage d'anticorps dans la population d'anticorps est éliminé. Il s'agit d'un mauvais ou pire % β de la population totale. Ce mécanisme permet de déterminer de nouvelles permutations qui correspondent à des nouveaux domaines de recherche dans l'espace de recherche.

Nous résumons le processus d'édition de récepteurs dans le tableau 5.3.

Processus d'édition des récepteurs
<p><i>Pour une population courante</i></p> <p>Début</p>

Chapitre 5 : Application du SIA pour résoudre le problème du Job Shop

Éliminer un % β d'anticorps mauvais de cette population: Créer de nouveaux anticorps du même pourcentage (% β) : Changer ceux éliminés par les nouvellement créés : Fin
--

Tableau 5.3 : Processus d'éditions des récepteurs

5.3. Une analogie entre le Job Shop et le SIA :

Une analogie entre le Job Shop et le SIA peut être formulée. Dans certaines approches, l'antigène est assimilé à des données du problème d'ordonnancement (tâches, ressources, contraintes) et les lymphocytes B à des solutions (ordonnements) possibles. L'affinité entre lymphocyte B et antigène est assimilée à une fonction objectif à optimiser.

À un atelier, à une tâche et une machine, à un ordonnancement et à la fonction objectif « minimiser le Makespan » correspondent respectivement un organisme, un pathogène et un antigène, un anticorps et une fonction affinité.

Nous récapitulons cette analogie sous la forme du tableau ci dessous.

Systèmes immunitaires	Job Shop problème
Organisme	L'atelier
Pathogène, Antigène	Taches, opérations, ressources
Anticorps	Ordonnancement
L'affinité entre l'anticorps et Antigène	1/Makespan
Mémoire immunitaire	Mémoriser l'affectation des taches sur les machines

Tableau 5.4: Analogie entre le Job Shop et le SIA.

5.4. Un algorithme de SIA pour résoudre le problème de job shop :

L'organigramme représenté dans la figure 5.3 donne des détails de l'algorithme SIA pour résoudre le problème d'ordonnancement de type job shop.

Chapitre 5 : Application du SIA pour résoudre le problème du Job Shop

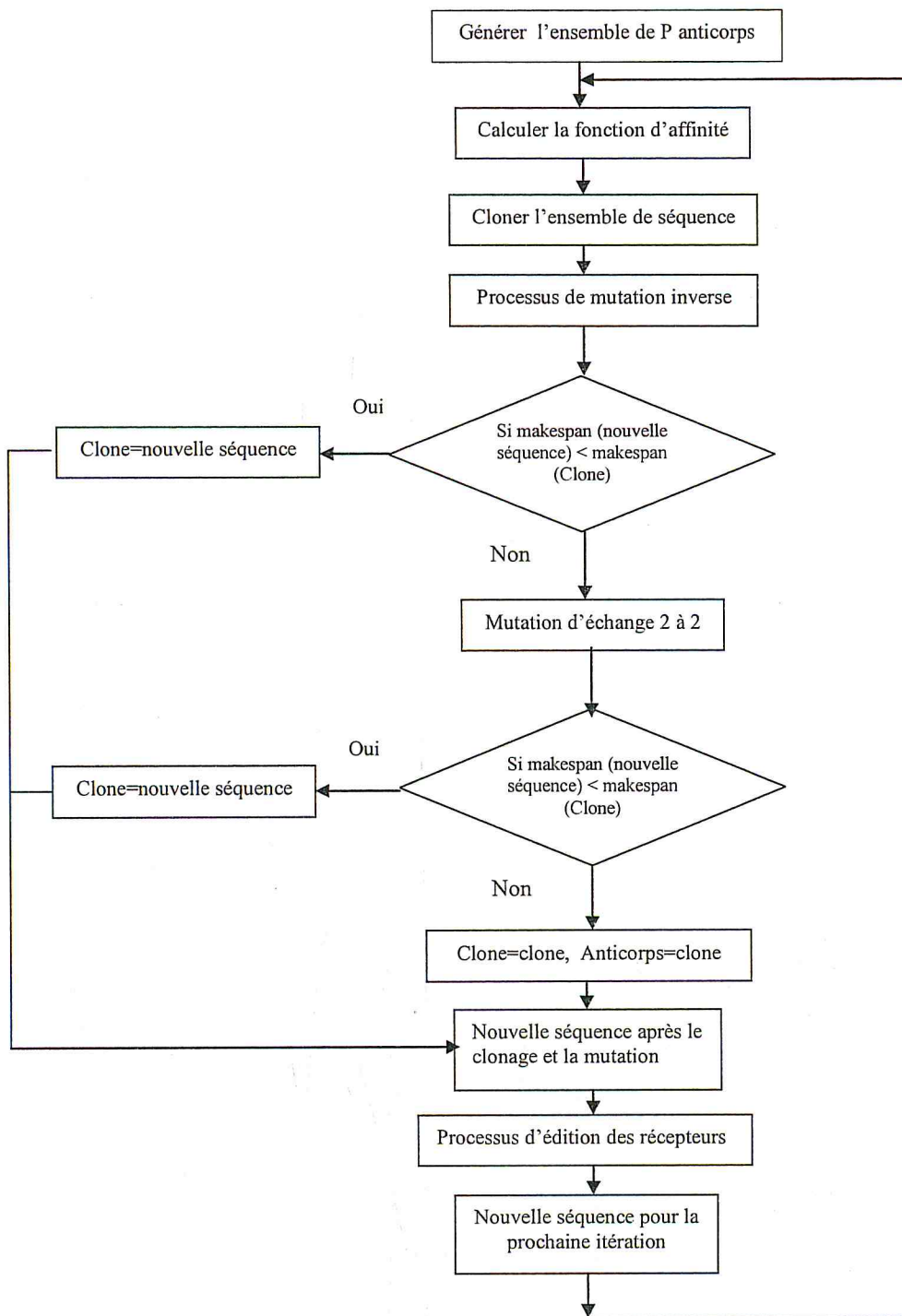


Figure 5.3: Organigramme d'algorithme de SIA pour résoudre le job shop

Chapitre 5 : Application du SIA pour résoudre le problème du Job Shop

Des ordonnancements réalisables sont représentés par des permutations ou chaînes de n valeurs entières correspondantes aux n tâches. Les n tâches seront exécutés selon l'ordre de leur apparition dans la séquence. Ces chaînes sont associées à des anticorps de **SIA**. L'algorithme évolue et détermine une solution selon l'évolution de ces anticorps (Tableau 5.5).

Algorithme de SIA
Initialisation : initialiser les temps d'exécutions et les opérations des tâches.
Début
Générer une population de P anticorps (des permutations de tâches).
Répéter
<i>Pour chaque itération :</i>
Choisir un anticorps ou une permutation de tâches dans la population des anticorps ;
Déterminer l'affinité de chaque anticorps ;
(i) Appliquer le processus de clonage.
(ii) Appliquer le processus de mutation (pour chaque clone)
(iii) Appliquer le processus d'édition des récepteurs
Jusqu'à le critère d'arrêt = faux.
Fin

Tableau 5.5 : Algorithme de SIA

5.4.1. La règle du calcul du makespan

A chaque fois où une permutation est sélectionnée, nous l'ordonnons suivant la règle de priorité dénommée Most Work Remaining décrite brièvement comme suit:

MWKR [16]: nous sélectionnons l'opération i appartenant à la tâche ayant la plus grande quantité de travail W_i restant à exécuter, ou W_i est maximal.

5.4.2. Critère d'arrêt :

Comme tous les algorithmes de méta-heuristiques basés sur la génération aléatoire des populations, il faut pour sortir de l'algorithme un critère d'arrêt. Pour notre algorithme étudié nous avons choisis celui qui dépend du nombre d'itérations et il est $100 \times n$, où n est le nombre de tâches.

5.5. Un exemple illustratif

Nous considérons le problème de trois machines suivant :

Temps d'exécution des opérations			
Tâche	1	2	3
T1	3	3	2
T2	1	5	3
T3	3	2	3

Tableau 5.6: le temps d'exécution de chaque tâche

Opérations des séquences machines			
Tâche	1	2	3
T1	3	3	2
T2	1	5	3
T3	3	2	3

Tableau 5.7: les opérations de chaque tâche

Nous supposons qu'un anticorps est donné par la permutation suivante [3 2 1], où 1, 2 et 3 représentent respectivement les tâches T1, T2 et T3. Cette permutation doit être opérée 3 fois dans le même ordre car chaque tâche à 3 opérations.

On peut donc considérer la permutation initiale comme la formule suivante [3 2 1 3 2 1 3 2 1].

Il y a trois 2 (trois opérations de la tâche 2) dans la permutation qui correspondent respectivement aux première opération du tâche T2 sur la machine 1, sa seconde opération sur la machine 3, et sa troisième opération sur la machine 2.

Nous avons construit le diagramme de Gantt en suivant le même ordre pour avoir un ordonnancement optimal (voir figure 5.2).

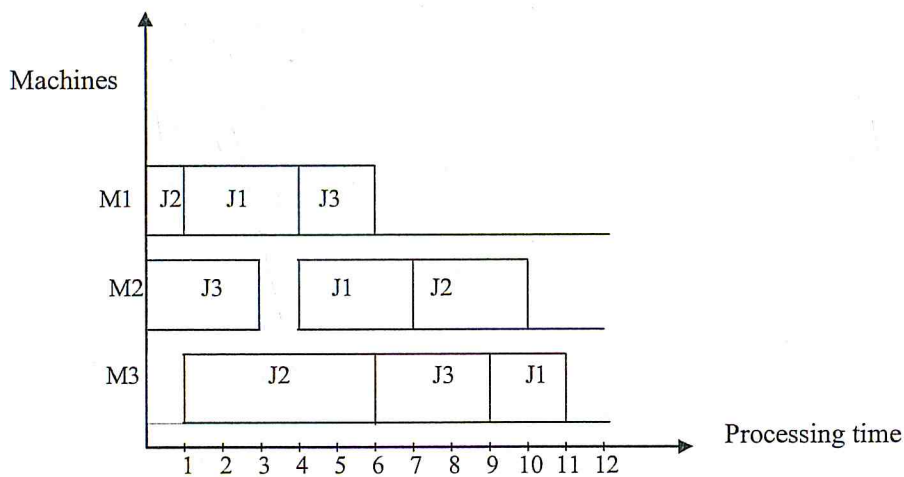


Figure 5.4 : Diagramme de Gantt pour un ordonnancement optimal (par AIS-JSS)

5.6. Conclusion

L'approche **SIA** a été utilisée pour résoudre le problème du tâche shop avec la fonction objectif est de minimiser le makespan.

Le déroulement de l'algorithme apparaît simple mais avec des techniques efficaces (processus du clonage, processus des mutations et processus d'édition des récepteurs).

Chapitre 6:

Implémentation de SIA

6.1. Introduction

Ce présent chapitre est consacré à l'implémentation informatique de la méta-heuristique SIA, Système Immunitaire Artificiel, pour résoudre le problème de Job Shop.

Un langage de Programmation Orienté objet est utilisé. Nous confirmons les résultats proposés dans l'article de Chandrasekaran et al. [8]. Nous avons testé le programme sur 162 problèmes (instances) de tests (Benchmark) tirée de la littérature [[http:// www.mscmga.ms.ic.ac.uk/info.html](http://www.mscmga.ms.ic.ac.uk/info.html) ou <ftp://mscmga.ms.ic.ac.uk/info.html>].

Nous avons pu réaliser des graphiques du même type que ceux d'auteurs et d'autre par nous-mêmes. Les résultats trouvées sont très proches des bornes ou la solution optimal et ils sont meilleurs que ceux de la procédure « recherche taboue de la machine goulot » reprise de l'article.

Dans ce chapitre nous décrivons succinctement le logiciel développé " AIS-JSS" comme résultat de notre implémentation, et nous synthétisons des résultats obtenues de quelques instances.

6.2. Le langage de la programmation

Nous avons utilisé le Delphi version 7, pour la mise en œuvre informatique d'algorithme proposé dans notre travail.

Delphi est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (Application Development) basé sur le langage Pascal. Nous pouvons l'utiliser, développer toutes sortes d'applications, d'utilitaires généraux jusqu'aux programmes sophistiqués d'accès aux données, en passant par les applications client-serveur En utilisant Delphi.

Il fournit une bibliothèque de composants réutilisables (boutons, boîtes de dialogue, menus, etc...) et un ensemble d'outils de conception RAD incluant des modèles d'applications et des fiches. Ces outils simplifient le prototypage et le développement d'applications et réduisent la durée du développement.

Lorsqu'on démarre Delphi, on est immédiatement positionné dans l'environnement de programmation visuel. C'est dans cet environnement que Delphi fournit tous les outils dont on a besoin pour concevoir, développer, tester et déboguer les applications.

Avant la description de notre application AIS-JSS, nous discutons brièvement sur les benchmarks (affin de détailler dans l'annexe c) et nous détaillerons dans l'annexe C.

6.3. Benchmarks

En ordonnancement d'ateliers de type Job Shop, il est souvent utile d'avoir des *benchmarks* pour tester les modèles d'ordonnancement et surtout les méthodes de résolution appropriées.

Les benchmarks sont des problèmes-types construits par plusieurs auteurs pour tester les performances des approches de résolution, en procédant surtout à des comparaisons sur les mêmes instances [17].

Il s'agit d'instances théoriques (à l'instar des benchmarks des autres problèmes), formulées pour servir à l'étude du problème de Job Shop.

Un grand nombre de benchmarks de Job Shop sont proposés par plusieurs auteurs. Certains sont largement employés dans les recherches, alors que, d'autres sont moins connus.

Les benchmarks présentent les caractéristiques suivantes :

- Chaque benchmark est décrit par un ensemble de n tâches et m machines. Ce sont des problèmes job shop $n \times m$.
- Toutes les tâches comprennent exactement m opérations. Chaque tâche passe alors une et une seule fois sur chaque machine.
- Pour chaque opération, le temps d'exécution (nombre entier), ainsi que la machine sur laquelle elle doit s'exécuter, sont spécifiés.
- Les durées opératoires des opérations sont des entiers compris entre 1 et 100.
- A part le temps opératoire, il n'existe pas d'autres contraintes temporelles comme: la date d'échéance, le temps de préparation, la date de disponibilité etc.

- Les solutions optimales de ces benchmarks sont connues; ce qui constitue un bon outil pour les expérimentations et les comparaisons.

La figure 6.1 nous montre un exemple général de benchmark traitée par notre application AIS-JSS et que nous constatons les caractéristiques générales des benchmarks.

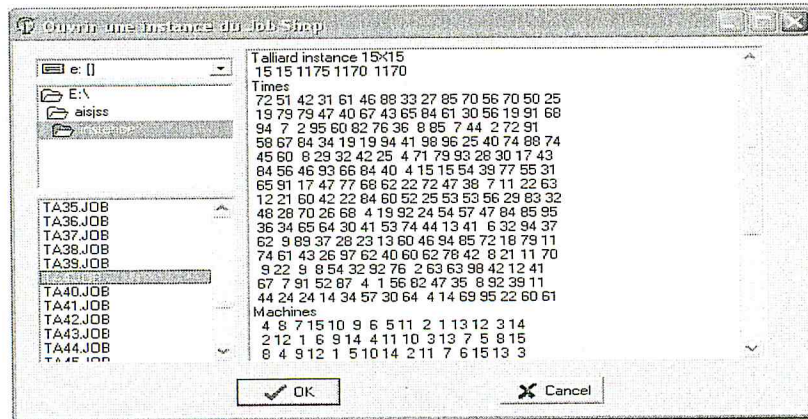


Figure 6.1 : Exemple d'un benchmark présenté par AIS-JSS(Ta4).

Comme le montre la figure 6.1, chaque instance doit vérifier un certain format pour qu'elle soit identifiée et résolue par l'AIS-JSS. Ce format se caractérise par :

- L'auteur(s) d'instance
- La définition de la taille de l'instance à la première ligne: $n \times m$,
- Times
- Chaque ligne des n lignes suivantes décrit les m temps d'exécutions de chaque tâche
- Machines
- Chaque ligne des n lignes suivantes décrit les m opérations de chaque tâche

6.4. Description de l'application informatique

L'AIS-JSS (Artificial Immune System for Job Shop Scheduling) est un logiciel simplifié développé pour la résolution du problème d'ordonnancement de Job Shop.

Nous avons préféré le développement d'un outil simple, autonome, dédié au problème traité. Cet outil offre par son interface simple (figure 6.2) un banc d'essai important pour expérimenter et explorer la métaheuristique.

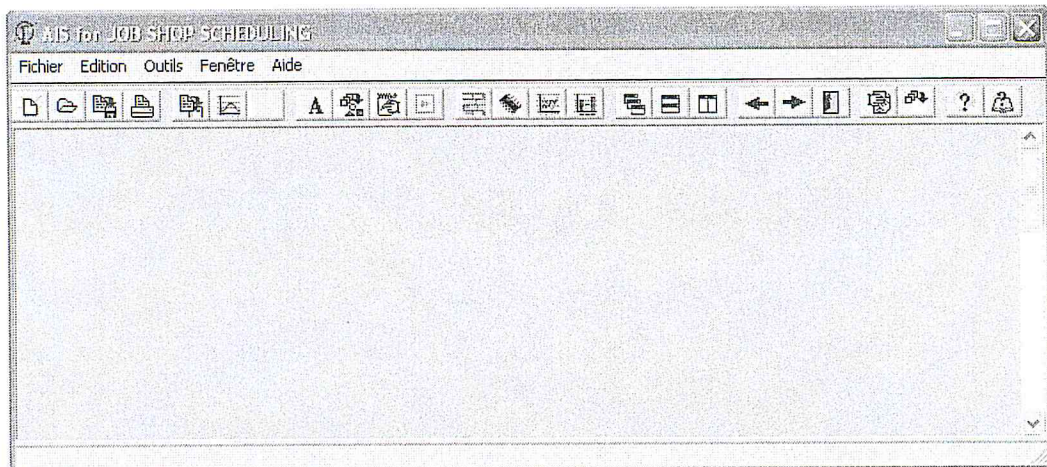


Figure 6.2 : L'interface graphique d'AIS-JSS.

Sans s'enfoncer dans la description détaillée de cet outil. Nous décrivons seulement les étapes essentielles permettant son exploitation.

6.4.1. Spécification des instances (problèmes)

La première étape d'utilisation d'AIS-JSS, est la détermination d'instance de Job Shop constituant le problème à résoudre. Les instances doivent être de type Job Shop simple constitué de m machines et n tâches. Les instances sont de taille petite à moyenne (n et m varient entre 4 et 100). Deux alternatives de détermination des instances sont possibles. L'une d'elles est l'utilisation des benchmarks connus, fournis par le logiciel.

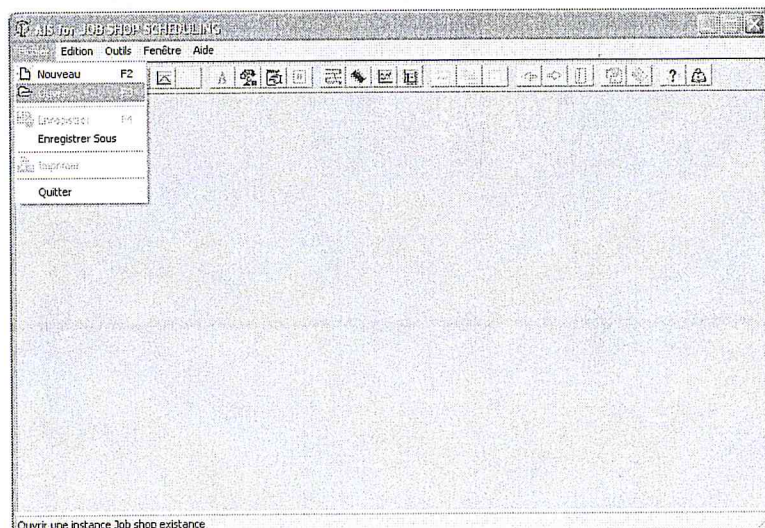


Figure6.3 : Ouvrir une Benchmark.

L'autre, c'est la construction de nouvelle instance à l'aide du logiciel.

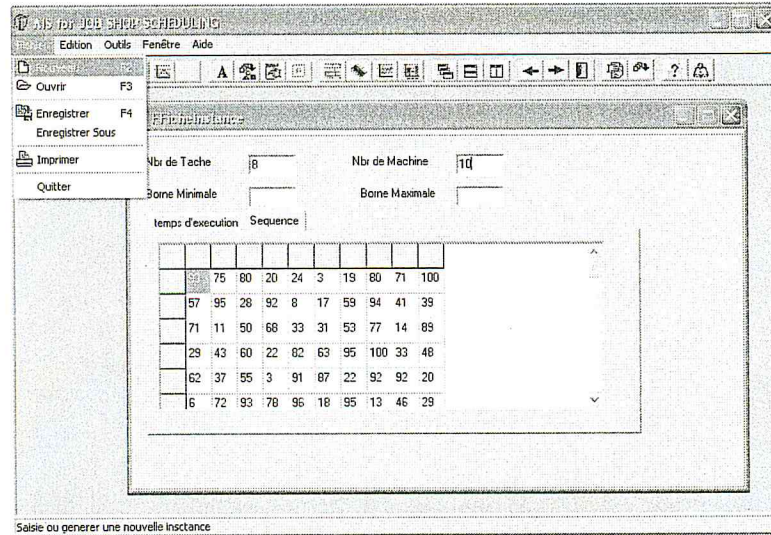


Figure 6.4 : Générer une nouvelle instance.

6.4.2. Présentation des résultats :

Après le lancement de l'exécution de la procédure AIS, la courbe d'évolution de la résolution s'affiche durant le processus de recherche (figure 6.5). Cette courbe indique à chaque moment (itération) le makespan de la solution actuelle.

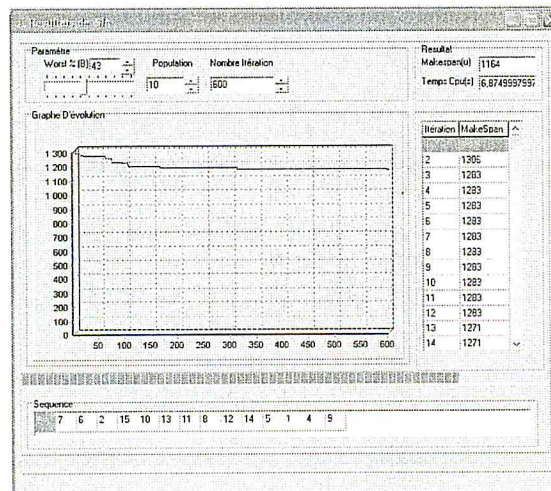


Figure 6.5 : La courbe de l'évolution avec de la résolution avec le SIA.

Quand la recherche termine, le logiciel affiche les résultats de résolution sur des graphiques et des tableaux.

La figure 6.6 présente le diagramme de Gantt-machine d'une solution de SIA trouvée par AIS du benchmark ft6.

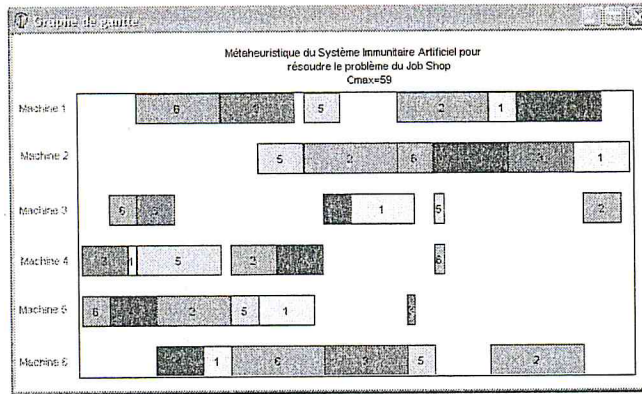


Figure 6.6 : Diagramme de Gantt représentant la solution de SIA pour ft6.

6.4.3. Le nombre de générations

Le processus de recherche par un système immunitaire pour tous les problèmes testé suit une allure générale illustrée par la courbe de la figure 6.7. Cette courbe montre l'existence de deux temps de recherche. Un premier temps, se caractérise par une décroissance rapide du makespan minimal des générations successives, ce qui signifie que la recherche est efficace. A un deuxième temps, la recherche commence à se stabiliser, et l'amélioration de la solution se fait très lentement jusqu'à stabilisation de la recherche. L'évolution durant les dernières générations devient inutile.

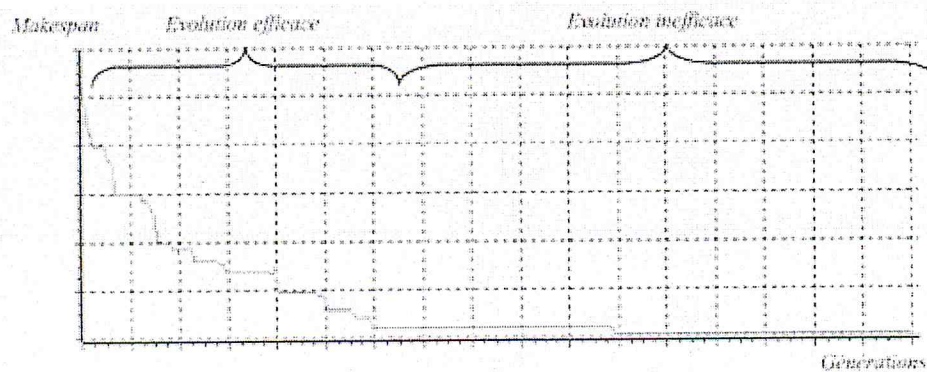


Figure 6.7: Amélioration de la solution durant l'évolution des générations de l'Algorithme de SIA.

D'après les résultats obtenus par les séries d'expériences réalisées sur les différents paramètres de l'Algorithme SIA, nous constatons que cette méthode est très intéressante pour la résolution du problème de Job Shop en raison des bonnes solutions retournées. Les paramètres de la méthode, jouent aussi un rôle important dans le processus de résolution. Ce qui fait qu'un bon choix de ces paramètres est une condition primordiale de la réussite de la recherche.

6.5. Résultats généraux

Dans le but d'explorer la performance relative de la Méta-heuristique en résolution du problème de Job Shop, une série d'essais a été effectuée sur **80 benchmarks** (TA01-TA80) de tailles et de complexité diverses. Pour chaque problème, nous lui calculons sa valeur d'ordonnancement par l'Algorithme JSS, puis nous comparons les résultats obtenus avec ceux de la procédure TSSB tirées de la littérature [9,18]. Le tableau 6.1 et les graphiques des figures 6.9, 6.10 et 6.11 rapportent les résultats obtenus. Notons que le paramétrage de la méthode est pris par défaut ($\beta=30\%$), et (le nombre d'itération= $100 \times n$).

Chapitre 6 : Implémentation de SIA

Instance	TSSB	AIS	Instance	TSSB	SIA	Instance	TSSB	SIA
TA1	1231	1233	TA28	1602	1601	TA55	2681	2686
TA2	1244	1247	TA29	1583	1519	TA56	2781	2782
TA3	1206	1210	TA30	1573	1474	TA57	2943	2948
TA4	1170	1178	TA31	1764	1766	TA58	2885	2893
TA5	1215	1211	TA32	1824	1782	TA59	2655	2665
TA6	1210	1220	TA33	1829	1779	TA60	2723	2726
TA7	1223	1230	TA34	1841	1831	TA61	2868	2875
TA8	1187	1192	TA35	2009	2014	TA62	2895	2875
TA9	1297	1251	TA36	1825	1822	TA63	2755	2757
TA10	1241	1242	TA37	1796	1779	TA64	2702	2710
TA11	1357	1324	TA38	1699	1676	TA65	2725	2735
TA12	1367	1329	TA39	1803	1802	TA66	2845	2849
TA13	1369	1271	TA40	1684	1632	TA67	2842	2832
TA14	1345	1352	TA41	2019	1866	TA68	2784	2788
TA15	1348	1296	TA42	1956	1869	TA69	3075	3073
TA16	1351	1303	TA43	1902	1810	TA70	2995	2999
TA17	1458	1459	TA44	1987	1935	TA71	5464	5465
TA18	1412	1378	TA45	2011	2002	TA72	5181	5187
TA19	1336	1278	TA46	1997	1941	TA73	5568	5568
TA20	1347	1325	TA47	1906	1789	TA74	5339	5347
TA21	1649	1542	TA48	1982	1914	TA75	5396	5393
TA22	1627	1513	TA49	1993	1919	TA76	5344	5350
TA23	1556	1472	TA50	1975	1811	TA77	5436	5445
TA24	1624	1611	TA51	2760	2764	TA78	5394	5400
TA25	1580	1506	TA52	2756	2763	TA79	5358	5361
TA26	1672	1548	TA53	2717	2719	TA80	5183	5188
TA27	1688	1620	TA54	2839	2842			

Tableau 6.1 : Résultats généraux obtenus par application du SIA.
(comparés avec ceux obtenus par TSSB)

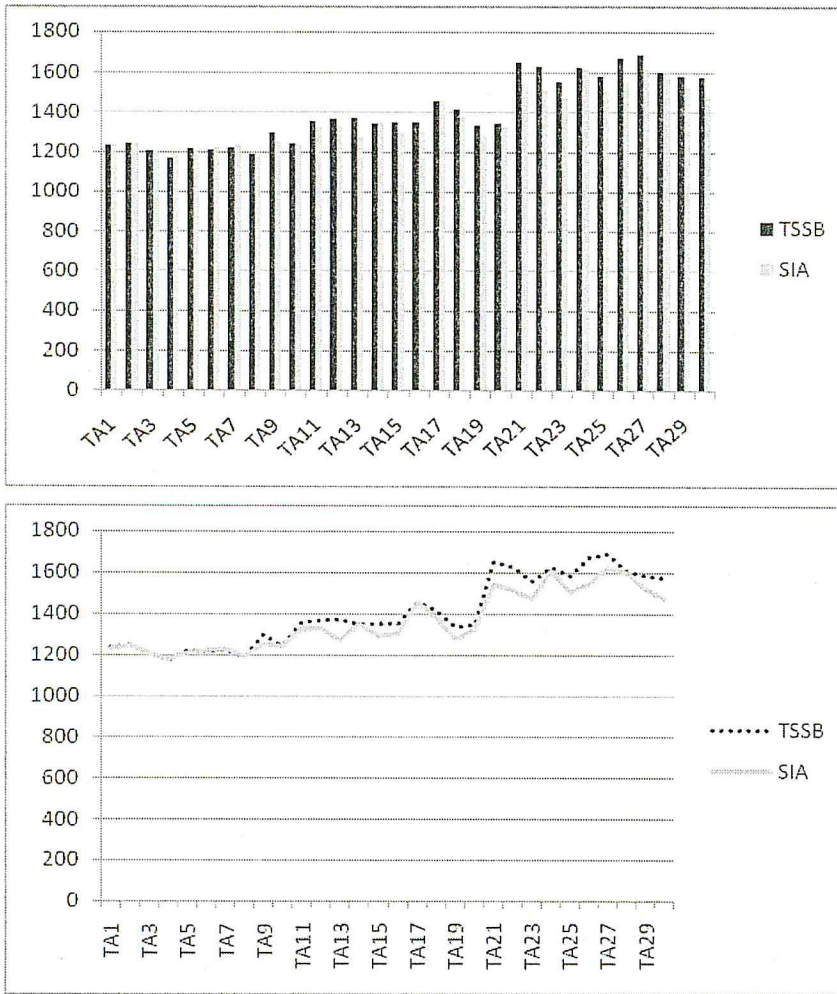
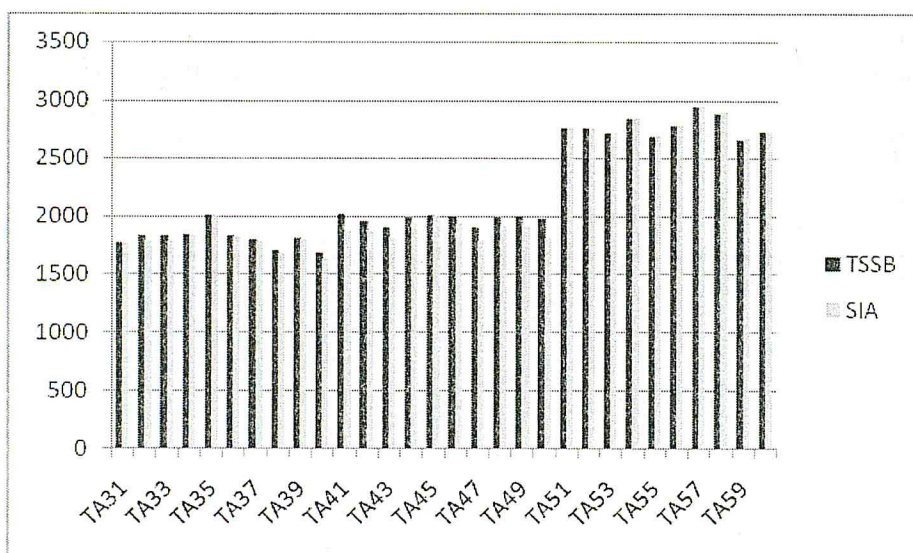


Figure 6.8 : Présentations graphiques des résultats des instances TA1-TA30.



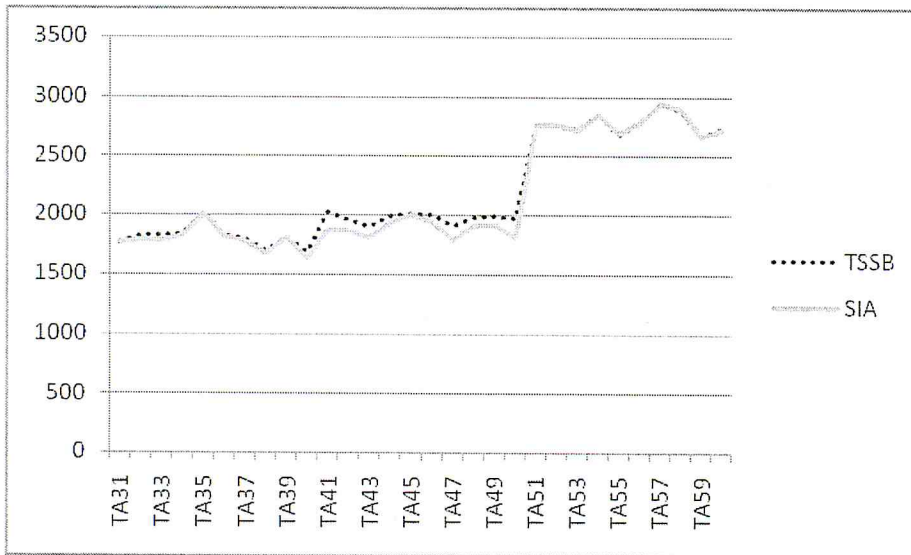
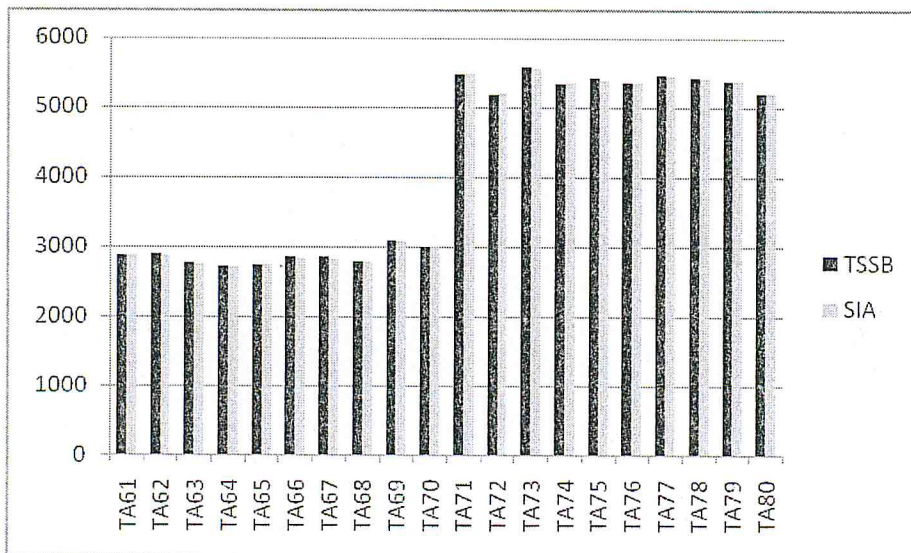


Figure 6.9 : Présentations graphiques des résultats des instances TA31-TA60.

Nous remarquons que pour les problèmes de petites et moyennes tailles que le SIA est plus performant que la TSSB.



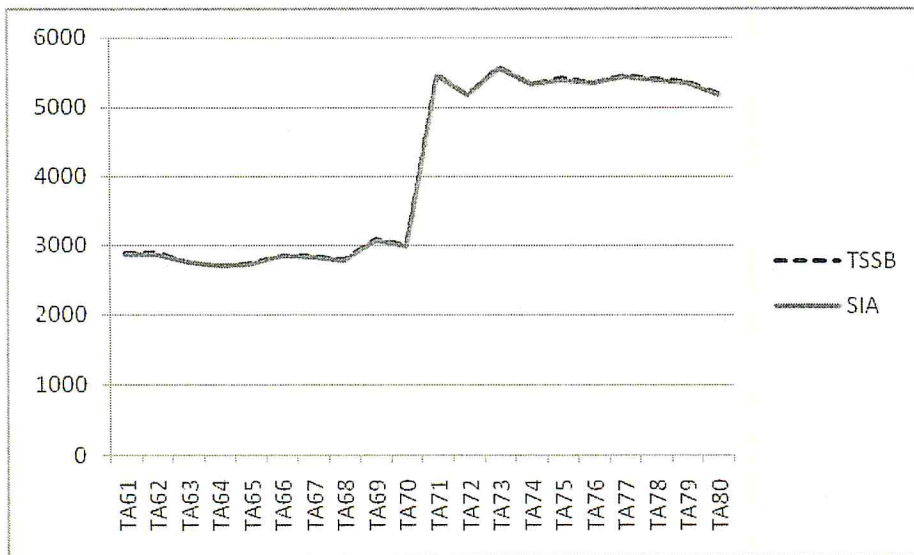


Figure 6.10: Présentations graphiques des résultats des instances TA61-TA80

Les graphiques nous montrent que le méta heuristique SIA est efficace et utile pour la résolution du problème du job shop. Le SIA nous donne des bonnes valeurs très proches de l'optimales.

D'après ces résultats, apparaît l'intérêt majeur du méta heuristique en le comparant avec la solution de la procédure TSSB. Cette dernière reste toujours un peu loin de l'optimal, alors que notre méthode produit de très bonnes solutions, en terme du makespan.

Ces résultats permettent aussi de valider notre implémentation. En effet, pour les problèmes faciles, Pour des problèmes reconnus comme difficiles, la méthode fournit des solutions très proches de l'optimum dans la majorité des cas, et même elles peuvent atteindre dans certains cas les solutions optimales.

6.6. Conclusion

Au travers ce chapitre, nous avons présenté et expliqué l'implémentation de la méta-heuristique retenue, en les appliquant à la résolution du problème de Job Shop. Notre application ne porte que sur la version standard de la méthode. Ceci exclut plusieurs approches plus avancées dont la recherche ne cesse de les proposer.

Le reste des résultats expérimentaux seront présentés dans l'annexe C, où nous disposeront de 82 autres instances.

Conclusion générale

La résolution du problème de job shop qui a pour fonction objectif de minimiser le makespan, se fait par l'utilisation de plusieurs méta-heuristiques nous citons quelques une : le recuit simulé, la recherche tabou et l'algorithme génétique.

Dans ce mémoire nous avons étudié le système immunitaire artificiel (SIA), une nouvelle méta-heuristique récente.

Cette méthode se base sur la génération aléatoire de la population (espace de recherche), et des processus biologiques comme le clonage, la mutation qui fournissent à une amélioration de l'espace de recherche. Et aussi le processus d'édition de récepteurs qui sert reproduire des nouveaux individus dans la population.

De notre recherche bibliographique nous avons pris des instances mises par des chercheurs comme TAILLARD, LAURENCE, YAMADA, ADAMS,...et d'autres où nous parlerons sur eux dans l'annexe C. Nous avons repris aussi des résultats de la procédure recherche tabou de la machine goulot appliqués sur les mêmes instances citées avant pour les comparés avec les nôtres obtenus par notre application nommée AIS-JSS.

Les résultats obtenus selon l'AIS-JSS sont très proches de ceux les optimums des instances et elles sont meilleurs que ceux de la procédure recherche tabou de la machine goulot.

Cette Méta-heuristique (SIA) est donc l'objet d'une recherche intense pour la résolution de divers problèmes NP-difficiles dont le problème d'ordonnement de Job Shop fait partie.

Plusieurs problèmes sont ouverts où on peut appliquer le SIA pour les résoudre par exemple, le flow shop hybride, le job shop avec des machines parallèles...et avec des critères différents de makespan comme le flow time, flow time moyen, le nombre de taches en retard...

Annexe A

Définitions biologiques

Activation des macrophages : les macrophages au repos ne détruiront certaines bactéries que s'ils sont activés par une cellule **T**. L'activation des macrophages est importante pour contrôler l'infection.

Affinité : force de liaison d'une molécule à une autre impliquant un seul site d'interaction, comme la liaison d'un fragment **Fab** monovalent d'anticorps à un antigène monovalent.

Anticorps : protéine qui se fixe spécifiquement à une substance particulière, son antigène. Chaque molécule d'anticorps a une structure unique qui lui permet de lier spécifiquement son antigène, mais tous les anticorps partagent la même structure générale qui permet de les classer dans les immunoglobulines ou **Ig**. Les anticorps sont produits par les plasmocytes en réponse à une infection ou à une immunisation. Ils peuvent neutraliser les pathogènes ou faciliter leur capture et leur destruction par les phagocytes.

Antigène : toute molécule qui se fixe spécifiquement à un anticorps. Leur nom provient de leur capacité de générer des anticorps. Cependant, certains antigènes ne peuvent susciter la production d'anticorps par eux-mêmes.

Apoptose ou mort cellulaire programmée : mort cellulaire dans laquelle la cellule active un programme particulier qui se caractérise par la dégradation de l'**ADN**, la condensation et la dégénérescence nucléaire, les résidus cellulaires étant ensuite phagocytés. Les cellules en prolifération subissent fréquemment l'apoptose, qui est un processus naturel au cours du développement.

Autoanticorps : anticorps spécifiques des antigènes du soi.

Auto-immunité : immunité adaptative spécifiques des antigènes du soi, ce type de réponse étant qualifié d'auto-immune.

Bactéries : micro-organismes procaryotes qui comptent diverses espèces et souches et causent de nombreuses maladies infectieuses. Les bactéries peuvent vivre sur les téguments et muqueuses, dans les espaces extracellulaires, dans des vésicules cellulaires ou dans le cytosol, les différentes espèces bactériennes pouvant déclencher des maladies distinctes.

Cellule B ou lymphocyte B : un des deux types majeurs de lymphocytes. Le récepteur à l'antigène sur les lymphocytes **B**, habituellement appelé récepteur **B**, est immunoglobuline de membrane. Quand elles sont activées par l'antigène, les cellules **B** se différencient en cellules productrices d'anticorps de la même spécificité antigénique que son récepteur de surface. Les cellules **B** se divisent en deux classes.

Cellules cibles : les fonctions des cellules **T** effectrices sont toujours testées par les changements qu'elles produisent sur les cellules cibles qui portent les antigènes. Ces cellules peuvent être des lymphocytes **B**, activés pour produire des anticorps ; des macrophages, activés pour tuer des bactéries ou des cellules tumorales ; ou des cellules marquées qui sont tuées par les cellules **T** cytotoxiques.

Cellules dendritiques : aussi appelées cellules réticulaires interdigitées, elles se situent dans les aires des cellules **T** des tissus lymphoïdes et ont une morphologie ramifiée dite dendritique. Ce sont les agents de stimulation les plus puissants des réponses cellulaires **T**. Les cellules dendritiques dérivent des précurseurs de la moelle osseuse. Elles sont différentes des cellules dendritiques folliculaires qui présentent les antigènes aux cellules **B**.

Cellules effectrices accessoires : dans l'immunité adaptative, ces cellules participent à la réponse sans reconnaître directement l'antigène spécifique ; elles comprennent les phagocytes les mastocytes et les cellules NK.

Cellules mémoires : lymphocytes dont dépend la mémoire immunologique. Elles sont plus sensibles à l'antigène que ne le sont les lymphocytes naïfs et répondent rapidement quand elles entrent à nouveau en contact qui les a induites initialement. On a identifié des cellules T mémoires ainsi que des cellules B mémoires.

Cellules NK (Natural killer) : grands lymphocytes non-T, non-B, qui détruisent certaines cellules tumorales. Les cellules NK sont importantes dans l'immunité innée aux virus et aux autres pathogènes intracellulaires.

Cellules présentatrices d'antigènes (APC) : cellules hautement spécialisées qui appréhendent les antigènes et présentent les fragments peptidiques à la surface cellulaire associés à des molécules requises pour l'activation des cellules T. Les principales cellules présentatrices d'antigènes aux cellules T sont les cellules dendritiques, les macrophages et les cellules B.

Cellules T ou lymphocytes T : sous-population de lymphocytes définis par leur développement dans le thymus.

Champignons : organismes eucaryotes uni- et multicellulaires, comprenant les levures et les moisissures, qui peuvent causer toutes une variété de maladies. L'immunité contre les champignons est complexe et implique à la fois les réponses humorale et cellulaire

Clone : population de cellules qui dérivent d'un précurseur cellulaire unique.

Clone de cellule T : population de cellule T dérivée d'un seul progéniteur de cellule T.

Clonotypique : caractère propre à des cellules individuelles ou aux membres d'un clone. C'est ainsi qu'un anticorps monoclonal qui réagit avec le récepteur d'une lignée clonée de cellules T dit clonotypique ; il reconnaît un clonotype ou le récepteur clonotypique de ces cellules.

Complexe majeur d'histocompatibilité (CMH) : ensemble de gènes sur le chromosome 6 humain. Il code plusieurs glycoprotéines membranaires appelées molécules du CMH, et aussi des protéines impliquées dans l'apprêtement des antigènes et dans d'autres aspects de la défense de l'hôte. Les molécules du CMH sont souvent appelées antigènes majeurs d'histocompatibilité.

Complexe antigène - anticorps : associations d'antigènes et d'anticorps unis de manière non covalente, qui peuvent varier de taille, allant de petits complexes solubles jusqu'à des complexes si grands qu'ils deviennent insolubles et commencent à précipiter. On les appelle aussi complexes immuns.

Complément : le système du complément est un ensemble de protéines plasmatiques qui agissent ensembles pour attaquer les formes extracellulaires des pathogènes. L'activation du

Cytokines : protéines produites par des cellules et qui modifient le comportement des autres cellules. Les cytokines produites par les lymphocytes sont souvent appelées lymphokines ou interleukines, mais le terme générique de cytokine est le plus utilisé, elles agissent par les récepteurs spécifiques présent sur certaines cellules.

Ganglion lymphatique : organe lymphoïde secondaire. Ils se trouvent en de nombreux endroits de l'organisme où les vaisseaux lymphatiques convergent, et ce sont eux qui permettent de déclencher la réponse immunitaire adaptative. Les cellules présentatrices d'antigène et les antigènes amenés par les vaisseaux lymphatiques à partir d'un site d'infection sont présentés aux nombreux lymphocytes circulants qui traversent les ganglions lymphatiques. Certains de ces lymphocytes reconnaîtront l'antigène et y répondront, déclenchant ainsi la réponse immunitaire adaptative.

Immunisation : induction délibérée d'une réponse immune adaptative par l'introduction d'un antigène dans l'organisme.

Immunisation active : immunisation contre un antigène, le transfert d'anticorps à un individu non immunisé étant appelé immunisation passive.

Immunisation passive : l'injection d'un anticorps ou d'un sérum immun à un receveur naïf est appelée immunisation passive.

Immunité : capacité de résister à l'infection.

Immunité cellulaire ou réponse cellulaire: toute réponse immune adaptative dans laquelle les cellules T spécifiques de l'antigène jouent le rôle principal. Elle se définit de manière opérationnelle comme toute immunité adaptative qui ne peut être transférée par des anticorps sériques à un receveur naïf.

Immunité humorale : immunité spécifiques assurée par les anticorps. L'immunité humorale peut être transférée à des receveurs non immunisés par des sérums humains contenant des anticorps spécifiques.

Immunologie : étude de tous les aspects de la défense de l'hôte contre l'infection et des conséquences nocives des réactions immunitaires.

Leucocyte : terme générique pour définir un globule blanc. Les leucocytes comprennent les lymphocytes, les polynucléaires et les monocytes.

Lymph : liquide extracellulaire qui s'accumule dans les tissus et qui est ramené par les vaisseaux lymphatiques à travers le système lymphatique jusqu'au canal thoracique puis dans le sang.

Lymphocytes : toutes les réponses immunes adaptatives dépendent des lymphocytes. Les lymphocytes forment une classe de leucocytes qui portent différents récepteurs de surface à l'antigène. Des récepteurs sont codés par des segments géniques réarrangés. On distingue deux types principaux de lymphocytes, les lymphocytes B (cellules B) et les lymphocytes T (cellules T), qui assurent respectivement l'immunité humorale et l'immunité cellulaire.

Macrophages : grandes cellules phagocytaires mononucléaires importantes dans l'immunité innée, dans les phases précoces non adaptatives de la défense de l'hôte, comme cellules présentatrices d'antigène, ou comme cellules effectrices dans l'immunité humorale et cellulaire. Ce sont des cellules migrantes qui sont dérivées de précurseurs de la moelle osseuse et qui se trouvent dans la plupart des tissus de l'organisme. Ils jouent un rôle crucial dans la défense de l'hôte.

Mastocytes : grandes cellules présentes dans les tissus conjonctifs de tout l'organisme, plus abondantes dans les tissus sous-muqueux et les dermes. Les mastocytes jouent un rôle crucial dans les réactions allergiques. Le terme de mastocytose désigne une surproduction de mastocytes.

Maturation d'affinité : augmentation de l'affinité des anticorps pour leur antigène spécifique au cours d'une réponse immune humorale. Elle s'observe surtout lors d'une immunisation secondaire ou d'immunisations répétées.

Mécanismes effecteurs : processus par lesquels les pathogènes sont détruits et éliminés de l'organisme. Les réponses immunes innées et adaptatives utilisent très souvent les mêmes mécanismes effecteurs pour éliminer les pathogènes.

Moelle osseuse : tissu situé au centre des os, produit les différents types de cellules du sang : les globules blancs, les globules rouges et les plaquettes. Elle contient deux types de cellules spécifiques : les cellules souches hématopoïétiques (CSE) et les cellules du stroma médullaire qui sont les cellules hématopoïétiques. Chez l'homme, tous les os ont une activité

hématopoïétique jusqu'à l'âge de 5 ans. Par la suite, cette activité s'effectue essentiellement à la moelle osseuse des os courts et plats, plus précisément du sternum, des côtes, des vertèbres et des os iliaques.

Monocytes : leucocytes avec un noyau en forme de haricot, ce sont les précurseurs des macrophages.

Opsonisation : modification de la surface d'un pathogène ou d'une autre particule qui facilite leur ingestion par des phagocytes. Les anticorps et le complément opsonisent les bactéries extracellulaires pour qu'elles puissent être détruites par les neutrophiles et les macrophages.

Organes lymphoïdes : tissus organisés caractérisés par un très grand nombre de lymphocytes interagissant avec un stroma non lymphoïde. Les organes lymphoïdes centraux où les lymphocytes sont générés sont la moelle osseuse et le thymus. Les organes lymphoïdes périphériques principaux dans lesquels la réponse immune adaptative est induite sont les ganglions lymphatiques, la rate.

Organes lymphoïdes centraux ou primaires: sites de développement lymphocytaire. Chez l'homme, les lymphocytes **B** se développent dans la moelle osseuse, tandis que les lymphocytes **T** se développent dans le thymus à partir des progéniteurs dérivés de la moelle osseuse.

Organes lymphoïdes périphériques ou secondaires: les ganglions lymphatiques, la rate et les tissus lymphoïdes associés aux muqueuses, dans lesquels les réponses immunes sont induites, à l'opposé des organes lymphoïdes centraux, dans lesquels les lymphocytes se développent.

Parasites : organismes qui vivent aux dépens d'un hôte. En pratique médicale, le terme est restreint aux vers et aux protozoaires, sujets principaux de la parasitologie.

Pathogènes : microorganismes qui peuvent causer des maladies quand elles infectent un hôte.

Phagocytose : ingestion de particules par des cellules. Habituellement, les cellules phagocytaires ou phagocytes sont des macrophages ou des neutrophiles, et les particules sont des bactéries qui sont absorbées et détruites.

Plaquettes : ces fragments cellulaires que l'on trouve dans le sang jouent un rôle essentiel dans la coagulation sanguine.

Plasma : solution aqueuse d'électrolytes et de protéines plasmatiques qui constitue la phase liquide du sang.

Plasmocytes : lymphocytes **B** en phase terminale de différenciation. Ce sont les principales cellules sécrétrices d'anticorps. Ils se trouvent dans la médullaire des ganglions lymphatiques, dans la pulpe rouge de la rate et dans la moelle osseuse.

Pulpe blanche : partie de la rate constituée par des zones continues de tissu lymphoïde.

Pulpe rouge: partie non-lymphoïde de la rate dans laquelle les globules rouges âgés sont éliminés.

Rate : organe situé du côté supérieur gauche de la cavité péritonéale, comprenant la pulpe rouge, impliquée dans l'élimination des globules rouges sénescents, et la pulpe blanche constituée de cellules lymphoïdes qui répondent aux antigènes parvenant à la rate par le sang.

Réarrangement : l'expression du récepteur à l'antigène nécessite le réarrangement des segments géniques dans les lymphocytes en développement.

Réponse immune: réponse de l'hôte pour se défendre lui-même contre un pathogène.

Réponse immune adaptative: réponses des lymphocytes spécifiques à un antigène, y compris le développement de la mémoire immunitaire. Les réponses immunes adaptatives sont générées par sélection clonale des lymphocytes. Les réponses immunes adaptatives de

l'immunité, qui ne sont pas produite par la sélection clonale de lymphocytes spécifiques d'antigène. Les réponses immunes adaptatives sont aussi appelées réponses immunes acquises.

Réponse immune primaire: réponse immune adaptative à une exposition initiale à l'antigène.

Sélection : une cellule est sélectionnée par l'antigène quand ses récepteurs se fixent à l'antigène. Si la cellule se met alors à proliférer, on parle de sélection clonale, et la cellule fondera un clone, si la cellule meurt, on parle de sélection négative ou de délétion clonale.

Sélection négative : processus par lequel les thymocytes qui reconnaissent les antigènes du soi sont éliminés du répertoire au cours de développement intra-thymique. Les cellules **B** auto réactives subissent un processus semblables dans la moelle osseuse.

Sélection positive : processus par lequel seules les cellules **T** dont les récepteurs peuvent reconnaître les antigènes présentés par les molécules du **CMH** du soi peuvent arriver à maturité au cours de leur développement dans le thymus. Toutes les autres cellules **T** meurent avant d'atteindre la maturité.

Système immunitaire : ensemble formé par les tissus, les cellules et les molécules qui sont impliqués dans l'immunité adaptative. Parfois, cette expression désigne la totalité des mécanismes de défense de l'hôte.

Système lymphatique : ensemble des canaux lymphatiques et des tissus lymphoïdes qui drainent le liquide extracellulaire de la périphérie vers le canal thoracique puis le sang. Il comprend les ganglions lymphatiques, et d'autres éléments lymphoïdes organisés en dehors de la rate, qui communique directement avec le sang.

Théorie de la sélection clonal : paradigme central de l'immunité adaptative, il postule que la réponse immune adaptative dérive de lymphocytes spécifiques de l'antigène qui sont tolérants au soi. Ces lymphocytes spécifiques prolifèrent en réponse à l'antigène et se différencient en cellules effectrices spécifiques des antigènes du pathogène qui pourra ainsi être éliminé, et en cellule mémoire qui maintiendront l'immunité vis-à-vis de ce pathogène. Cette théorie a été proposée par Sir **Macfarlane Burnet** et dans ses premières formulations par **Niels Jerne** et **David Talmage**.

Thymocytes : cellules lymphoïdes trouvées dans le thymus. Ce sont principalement des cellules **T** en développement, mais quelques thymocytes ont atteint une maturité fonctionnelle.

Thymus : organe lympho-épithélial situé dans la partie supérieure et médiane du thorax, juste derrière le sternum ; il constitue le site de développement des cellules **T**.

Vaccination : induction délibérée d'une immunité adaptative contre un pathogène par l'injection d'un vaccin, une forme inactivée ou atténuée (non pathogène) du pathogène.

Annexe B

Résultats de l'OR-Library

1. Introduction

La *OR-Library* [19] est une bibliothèque constituée d'un très grand nombre d'instances de différents problèmes largement utilisés dans la « communauté recherche opérationnelle ».

Ces instances sont disponibles sur le site <ftp://mscmga.ms.ic.ac.uk/pub> ou sur le site <http://mscmga.ms.ic.ac.uk/info.html>.

Nous ne présentons ici que les instances de *jobshop* simple. Ces instances sont réparties en deux catégories:

- la première est constituée d'instances proposées par différents auteurs à partir de problèmes plus ou moins réels;
- la seconde est constituée d'instances générées par le générateur d'instances proposé par Éric Taillard.

La *OR-Library* constitue une source d'informations très utiles lorsqu'il s'agit de comparer des méthodes entre-elles.

2. Première partie d'instances :

Cette partie est constituée de **82 instances de *jobshop*** (dernière mise à jour 23/02/96) rassemblées par Dirk C. Mattfeld et Rob J.M. Vaessens:

Instances	Instances issues de
abz5-abz9	J. Adams, E. Balas and D. Zawack (1988), The shifting bottleneck procedure for jobshop scheduling, Management Science 34, 391-401.
ft06, ft 10, ft20	H. Fisher, G.L. Thompson (1963), Probabilistic learning combinations of local jobshop scheduling rules, J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, 225-251.
la01-la40	S. Lawrence (1984), Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania.
orb01-orb10	D. Applegate, W. Cook (1991), A computational study of the jobshop scheduling instance, ORSA Journal on Computing 3, 149-156. (ils sont générées dans Borne en 1986)
swv01-swv20	RH. Storer, S.D. Wu, R. Vaccari (1992), New search spaces for sequencing instances with application to jobshop scheduling, Management Science 38, 1495-1509.

yn1-yn4	T. Yamada, R Nakano (1992), A genetic algorithm applicable to large-scale jobshop instances, R Manner, B. Manderick (eds.), Parallel instance solving from nature 2, North-Holland, Amsterdam, 281-290.
---------	--

Tableau B.1 : Les différentes instances de la 1^{ère} partie.

Pour chaque instance, les tableaux suivants donnent la valeur optimale (si elle est connue), ou les meilleures bornes inférieures et supérieures connues à ce jour. La taille de chaque instance est également précisée, ainsi qu'un code référant le premier auteur (ou groupe d'auteurs) qui ont prouvé la valeur de la borne inférieure, ou trouvé une solution dont la valeur est la borne supérieure.

La liste des codes utilisés est située sous les tableaux.

abz5-6 10x10	123 AC AC	943 AC ABZ				
abz7-9 20 x 15	656 MS MS	645- 669 MS BV	661- 679 MS BV			
ft06 6x6	55 LLR LLR					
ft 10 10x10	930 CPi La					
ft20 20 x 5	116 CPi CPi					
la01-05 10 x 5	666 ABZ ABZ	655 ABZ LAL	597 AC MSS	5 AC LAL	593 ABZ AB	
la06-10 15 x 5	926 ABZ AB	890 ABZ ABZ	863 ABZ ABZ	9 ABZ ABZ	958 ABZ LA	
la11-15 20 x 5	122 ABZ AB	103 ABZ ABZ	1150 ABZ ABZ	1292 ABZ ABZ	1207 ABZ AB	
la16-20 10x10	945 CP2 CP2	784 CP2 MSS	848 AC MSS	8 AC MSS	902 AC LA	
la21-25 15 x 10	104 VAL VA	927 AC MSS	1032 ABZ ABZ	9 AC AC	977 AC AC	
la26-30 20 x 10	121 ABZ LAL	123 ABZ CP3	1216 ABZ MSS	1142-1153 NR VAYN	1355 ABZ AB	
la31-35 30 x 10	178 ABZ AB	185 ABZ ABZ	1719 ABZ ABZ	1721 ABZ ABZ	1888 ABZ AB	
la36-40 15 x 15	126 CP2 CP2	139 CP2 AC	1196 VAL NS	1233 AC AC	1222 AC AC	
orb01-05 10x10	105 AC AC	888 AC AC	1005 AC AC	1005 AC AC	887 AC AC	
orb06-10 10x10	101 AC AC	397 AC AC	899 AC AC	9 AC AC	944 AC AC	
swv01-05 20 x 10	1392-1418 VA BV	1475 VA VABV	1369-1398 VA VAWe	1450-1483 VA BV	1421-1434 VA BV	
swv06-10 20 x 15	1591-1696 VA BV	1446-1620 VA VABV	1640-1770 VA VAAa	1604-1663 VA VABV	1631-1773 VA BV	

swv11-15 50 x 10	2983-3005 VA VA	2972-3038 VA BV	3104-3146 VA Aa	2968 VA BV	2885-2940 VA VAa
swv16-20 50 x 10	292 VA SW	2794 VA SWV	2852 VA SWV	2843 VA SWV	2823 VA SWV
yn 1-4 20 x 20	826- 888 CL We	861- 909 CL VABV	827- 894 VA VABV	918- 972 VA BV	

Tableau B.2: Les bornes ou la solution optimale de chaque instance.

Aa	B. Aarts (1996), Personal communication. (a parallel taboo search algorithm).
ABZ	J. Adams, E. Balas and D. Zawack (1988), The shifting bottleneck procedure for jobshop scheduling. <i>Management Sci.</i> 34, 391-401.
AC	D. Applegate, W. Cook (1991), A computational study of the jobshop scheduling problem. <i>ORSA J. Comput.</i> 3, 149-156.
BV	E. Balas, A. Vazacopoulos (1995), Guided local search with shifting bottleneck for jobshop scheduling. Management Science Research Report #MSRR-609(R) [revised version], Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania.
CL	y. Caseau, F. Laburthe (1995), Disjunctive scheduling with task intervals. Report LIENS 95-25, Département de Mathématiques et Informatique, École Normale Supérieure, Paris, France.
CP1	J. Carlier, E. Pinson (1989), An algorithm for solving the jobshop problem. <i>Management Science</i> 35, 164-176.
CP2	J. Carlier, E. Pinson (1990), A practical use of Jackson's preemptive schedule for solving the jobshop problem. <i>Ann. Oper. Res.</i> 26, 269-287.
CP3	J. Carlier, E. Pinson (1994), Adjustments of heads and tas for the jobshop problem. <i>European J. Oper. Res.</i> 78, 146-161.
La	B.J. Lageweg (1984), Personal communication.
LAL	P.J.M. Van Laarhoven, E.H.L. Aarts, J.K. Lenstra (1992), Jobshop scheduling by simulated annealing. <i>Oper. Res.</i> 40, 113-125.
LLR	B.J. Lageweg, J.K. Lenstra, A.H.G Rinnooy Kan (1977), jobshop scheduling by implicit enumeration. <i>Management Science</i> 24, 441-450.
MS	P. Martin, D.B. Shmoys (1995), A new approach to computing optimal schedules for the jobs hop scheduling problem. Working paper, School of Operations Research an Industrial Engineering, Cornell University, Ithaca, New York.
MSS	H. Matsuo, C.J. Suh, R.S. Sullivan (1988), A controlled search simulated annealing method for the general jobshop scheduling problem. Working paper 03-04-88, Graduate School of Business, University of Texas, Austin.
NR	W. Nuijten, J. Rogerie (1996), Communicated by post on netnews.sci.op-research on Mon, 22 Jan 96.
NS	E. Nowicki, C. Smutnicki (1993), A fast taboo search algorithm for the jobshop problem. Preprinty nr 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw, Wroclaw, Poland.
SWV	RH. Storer, S.D. Wu, R Vaccari (1992), New search spaces for sequencing problems with application to jobshop scheduling. <i>Management Sei.</i> 38, 1495-1509.
VA	R.J.M. Vaessens (1995), Personal communication. (lower bounds found using Applegate & Cook's algorithm "edge-finder").
VAAa	R.J.M. Vaessens (1996), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of B. Aarts (1996)).
VABV	R.J.M. Vaessens (1995), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of E. Balas, A. Vazacopoulos (1994)).
VABV	R.J.M. Vaessens (1996), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of E. Balas, A. Vazacopoulos (1996)).
VAL	R.J.M. Vaessens, E.H.L. Aarts, J.K. Lenstra (1995), Jobshop scheduling by local search. Memorandum COSOR 94-05 (second revised version) Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

VAWe	R.J.M. Vaessens (1995), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of M. Wennink (1995)).
VAYN	R.J.M. Vaessens (1995), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of T. Yamada, R. Nakano (1995)).
We	M. Wennink (1995), Personal communication. (upper bounds found by a taboo search algorithm).
YN	T. Yamada, R Nakano (1995), Jobshop scheduling by simulated annealing combined with deterministic local search, MIC-95.

Tableau B.3 : Les auteurs des épreuves d'instances.

3. Deuxième partie d'instances :

Cette partie est constituée de **80 instances de jobshop** générées par le générateur d'instances proposé par Éric Taillard [20] (dernière mise à jour 30/06/99 sur la *OR-Library*, et 27/01/99 sur http://www.idsia.sh/~eric/problemes.dir / ordonnancement.dir /jobshop.dir /best _lb _up.txt).

Pour chaque instance, les tableaux suivants donnent la valeur optimale (si elle est connue), ou les meilleures bornes inférieures et supérieures connues à ce jour. La taille de chaque instance est également précisée, ainsi qu'un code référant le premier auteur (ou groupe d'auteurs) qui ont prouvé la valeur de la borne inférieure, ou trouvé une solution dont la valeur est la borne supérieure.

La liste des codes utilisés est située sous les tableaux.

ta01-05 15 x 15	1231 Tai Tai	1244 VA NS	121 B BV	117 B We	122 B B
ta06-10 15 x 15	1238 B B	1227 B B	121 B BV	127 B BV	1241 VA BV
ta11-15 20 x 15	1321-1364 VA BV	1321-1367 VA BV	1271-1350 VA BV	1345 VA NS	1293-1342 VA VAAa
ta16-20 20 x 15	1300-1362 VA AH	1458-1464 VA BV'	1369-1396 VA BV	1276-1341 VA BV	1316-1353 VA We
ta21-25 20 x 20	1539-1645 VA AH	1511-1601 VA Aa	1472-1558 VA BV	1602-1651 VA Aa	1504-1597 VA AHLS
ta26-30 20 x 20	1539-1651 VA AH	1616-1687 VA AHLS	1591-1615 VA BV	1514-1625 VA Aa	1473-1585 VA AHLS
ta31-35 30 x 15	1764 Tai AHLS	1774-1803 Tai BV	1778-1796 VA BV	1828-1832 Tai BV	200 VA Tai
ta36-40 30 x 15	1819 VA AH	1771-1784 Tai BV	1673-1677 Tai AHLS	179 VA AHLS	1631-1686 VA BV
ta41-45 30 x 20	1859-2023 VA BV'	1867-1961 VA BV	1809-1879 VA BV	1927-1998 VA AHLS	1997-2005 VA BV
ta46-50 30 x 20	1940-2029 Tai AHLS	1789-1913 VA AHLS	1912-1971 VA AHLS	1915-1984 VA AHLS	1807-1937 VA AHLS

ta51-55 50 x 15	2760 Tai Tai	2756 Tai Tai	271 Tai Tai	2839 Tai Tai	2679 Tai NS
ta56-60 50 x 15	2781 Tai Tai	2943 Tai Tai	288 Tai Tai	2655 Tai Tai	2723 Tai Tai
ta61-65 50 x 20	2868 Tai NS	2869-2872 VA AHLS	275 Tai NS	2702 BV NS	2725 Tai NS
ta66-70 50 x 20	2845 Tai NS	2825 VA AHLS	278 BV NS	3071 Tai NS	2995 Tai NS
ta71-75 100 x 20	5464 Tai Tai	5181 Tai Tai	556 Tai Tai	5339 Tai Tai	5392 Tai Tai
ta76-80 100 x 20	5342 Tai Tai	5436 Tai Tai	539 Tai Tai	5358 Tai Tai	5183 Tai NS

Tableau B.4 : Les bornes ou la solution optimale de chaque instance.

Aa	B. Aarts (1996), Personal communication. (a parallel taboo search algorithm)
AHLS	E. Aarts, Huub ten Eikelder, J.K. Lenstra, R. Schilham (23.6.1999), Personal communication. (An adaptive memory programm embedding a taboo search algorithm with NS neighbourhood and constant time Ta)
B	Wolfgang Brinkkoetter (27.1.1999), Personal communication. (The method combines ideas of Carlier and Pinson with an efficient way of propagating release/queue-time adjustments in a simulation of parallel preemptive machines.)
BV	E. Balas, A. Vazacopoulos (1995), Guided local search with shifting bottleneck for jobshop scheduling. Management Science Research Report #MSRR-609(R) [revised version], Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania.
BV'	E. Balas, A. Vazacopoulos (1996), Personal communication.
NS	E. Nowicki, C. Smutnicki (1993), A fast taboo search algorithm for the jobshop problem. Preprinty nr 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw, - Wroclaw, Poland.
TaI	E. Taillard (1993), Benchmarks for basic scheduling problems. EJOR 64, 278-285.
VA	R.J.M. Vaessens (1995), Personal communication. (lower bounds found using Applegate & Cook's algorithm "edge-finder").
VAAa	R.J.M. Vaessens (1996), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of B. Aarts (1996)).
VABV	R.J.M. Vaessens (1996), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of E. Balas and A. Vazacopoulos (1995)).
VAWe	R.J.M. Vaessens (1995), Personal communication. (upper bounds found using Applegate & Cook's algorithm "shuffie" with initial solutions of M. Wennink (1995)).
We	M. Wennink (1995), Personal communication. (upper bounds found by a taboo search algorithm).

Tableau B.5 : Les auteurs des épreuves d'instances.

Annexe C

Résultats numériques

Dans cette annexe, nous présentons les résultats de l'AIS-JSS appliquée pour **82** benchmarks qui sont classées en cinq classes :

Classe (1): la01-la40, 40 instances de huit tailles différentes ($n \times m = 10 \times 5$; 15×5 ; 20×5 ; 10×10 ; 15×10 ; 20×10 ; 30×10 ; 15×15). **[21]**

Classe (2): swv01-swv20, 20 instances de trois tailles différentes (20×10 ; 20×15 ; 50×10). **[22]**

Classe (3) : orb1-orb10, 10 instances de la même taille (10×10). **[23]**

Classe(4) : ft06, ft10, ft20 et abz5-abz9, 8 instances de tailles différentes (6×6 ; 10×10 ; 20×5 ; 10×10 ; 20×15). **[24,25]**

Classe(5) : yn1-yn4, 4 instances de la même taille 20×20 . **[26]**

Pour chaque classe, nous représentons les résultats obtenus par un tableau suivi de deux représentations graphiques (Histogramme et Courbe)

Instance	n	m	Sol Opt	SIA
la1	10	5	666	676
la2	10	5	655	661
la3	10	5	597	606
la4	10	5	590	598
la5	10	5	593	594
la6	15	5	926	934
la7	15	5	890	890
la8	15	5	863	870
la9	15	5	951	951
la10	15	5	958	960
la11	20	5	1222	1232
la12	20	5	1039	1041
la13	20	5	1150	1154
la14	20	5	1292	1293
la15	20	5	1207	1217
la16	10	10	945	946
la17	10	10	784	788
la18	10	10	848	854
la19	10	10	842	849
la20	10	10	902	904

Instance	n	m	Sol Opt	SIA
la21	15	10	1046	1054
la22	15	10	927	929
la23	15	10	1032	1035
la24	15	10	935	938
la25	15	10	977	982
la26	20	10	1218	1226
la27	20	10	1235	1237
la28	20	10	1216	1221
la29	20	10	1152	1159
la30	20	10	1355	1356
la31	20	10	1784	1792
la32	20	10	1850	1850
la33	20	10	1719	1729
la34	20	10	1721	1723
la35	20	10	1888	1898
la36	15	15	1268	1276
la37	15	15	1397	1404
la38	15	15	1196	1196
la39	15	15	1233	1237
la40	15	15	1222	1226

Tableau C.1 : Résultats du SIA comparés avec la solution optimale des instances de classe 1.

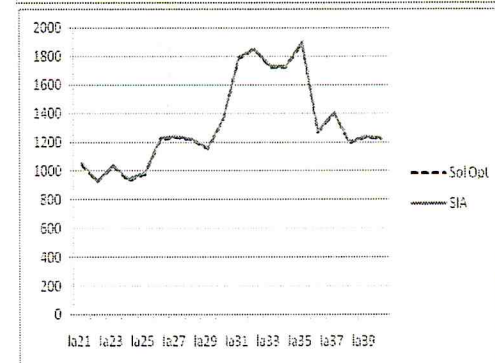
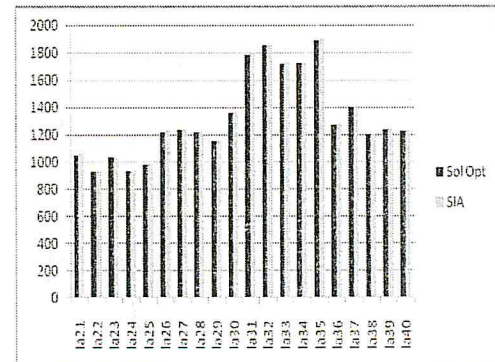
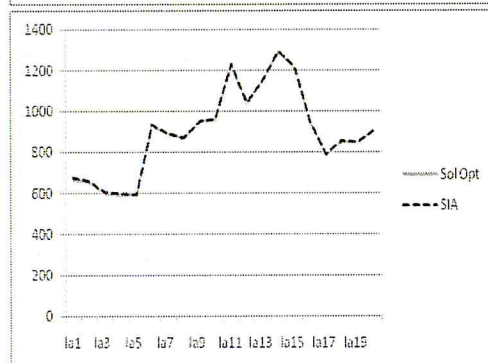
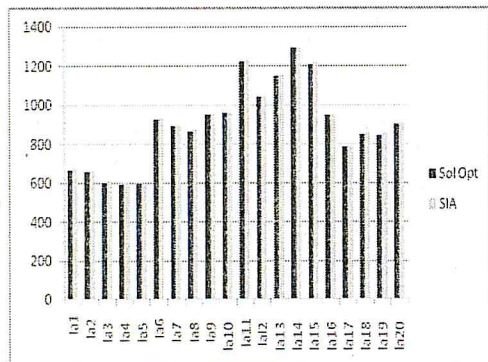


Figure C.1: Graphiques des instances de classe 1 (LA1-LA20).

Figure C.2: Graphiques des instances de classe 1 (LA20-LA40).

Instance	N	m	Sol Opt	SIA
swv 1	20	10	1407	1414
swv2	20	10	1475	1482
swv 3	20	10	1369	1371
swv4	20	10	1450	1454
swv 5	20	10	1424	1432
swv6	20	15	1591	1596
swv7	20	15	1446	1451
swv 8	20	15	1640	1645
swv9	20	15	1604	1609
swv 10	20	15	1631	1634
swv II	50	10	2983	2990
swv 12	50	10	2972	2973
swv 13	50	10	3104	3112
swv 14	50	10	2968	2972
swv 15	50	10	2885	2887
swv 16	50	10	2924	2930
swv 17	50	10	2794	2801
swv 18	50	10	2852	2856
swv 19	50	10	2843	2846
swv20	50	10	2823	2831

Tableau C.2: Résultats du SIA comparés avec la solution optimale des instances de classe 2.

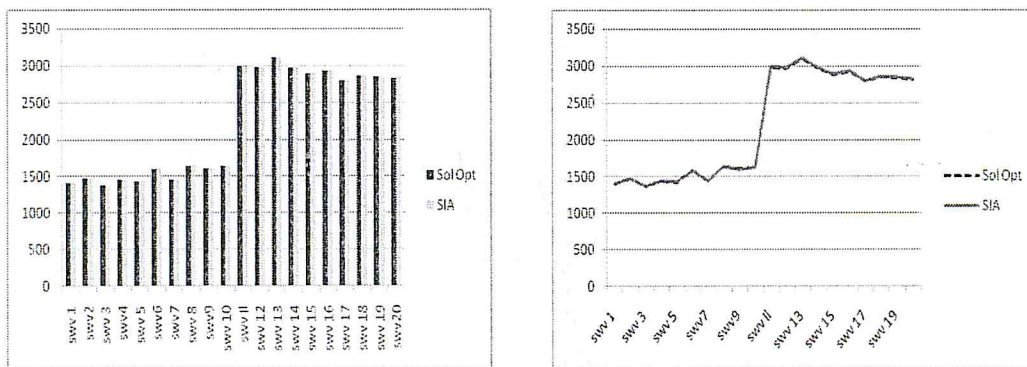


Figure C.3: Graphiques des instances de classe 2.

Instance	n	m	Sol Opt	SIA
orb 1	10	10	1059	1070
orb 2	10	10	888	888
orb 3	10	10	1005	1016
orb 4	10	10	1005	1015
orb 5	10	10	887	895
orb 6	10	10	1010	1011
orb 7	10	10	397	399
orb 8	10	10	899	907
orb 9	10	10	934	940
orb 10	10	10	944	947

Tableau C.3 : Résultats du SIA comparés avec la solution optimale des instances de classe 3.

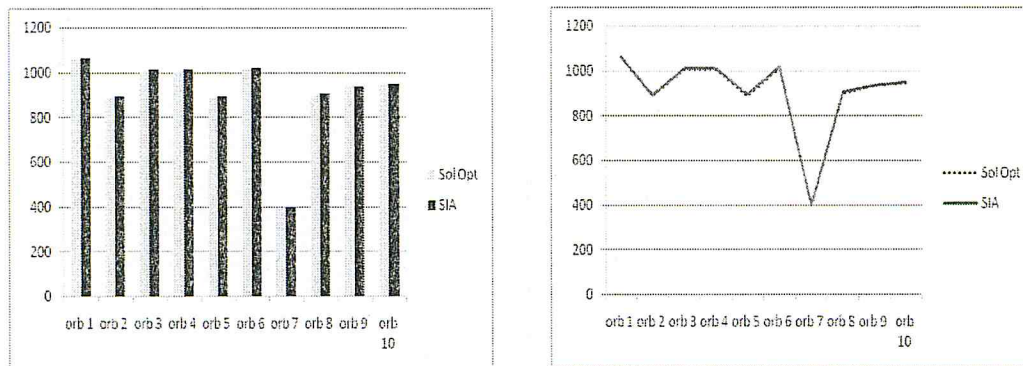


Figure C.4 : Graphiques des instances de classe 3.

Instance	n	m	Sol Opt	SIA
ft06	6	6	55	57
ft10	10	10	930	940
ft20	20	5	1165	1173
abz5	10	10	1234	1237
abz6	10	10	943	947
abz7	20	15	656	659
abz8	20	15	645	650
abz9	20	15	661	666

Tableau C.4: Résultats du SIA comparés avec la solution optimale des instances de classe 4.

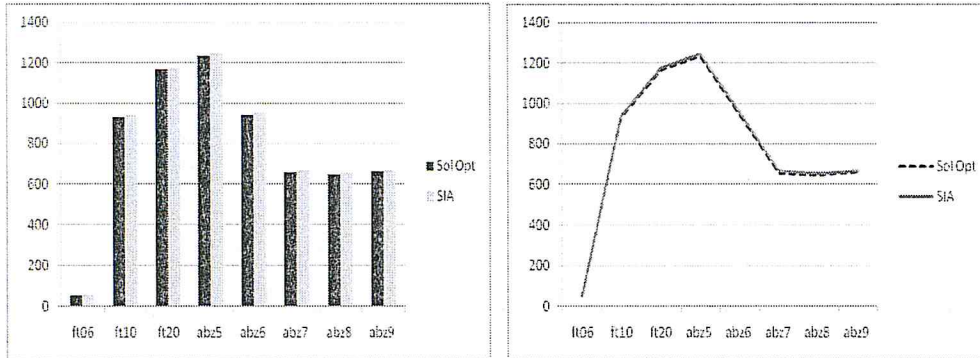


Figure C.5: Graphiques des instances de classe 4.

Instance	n	m	Sol Opt	SIA
yn 1	20	20	826	830
yn2	20	20	861	871
yn3	20	20	827	832
yn4	20	20	918	921

Tableau C.5: Résultats du SIA comparés avec la solution optimale des instances de classe 5.

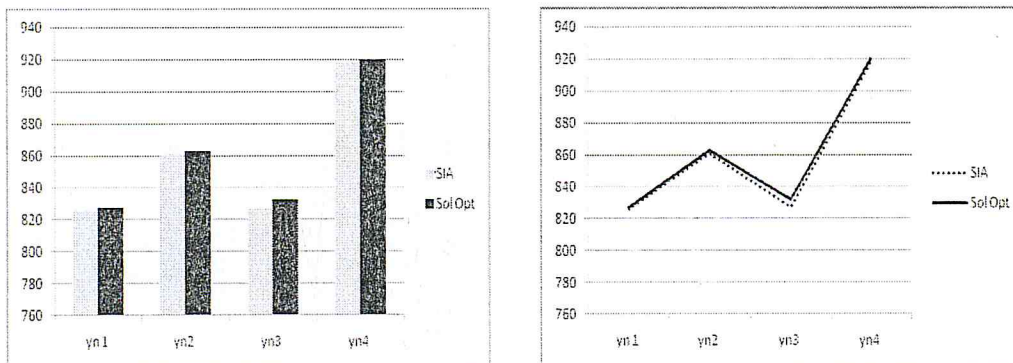


Figure C.6: Graphiques des instances de classe 5.

Bibliographie

- [1] Dasgupta D., Forrest S., « Novelty detection in time series data using ideas from immunology », Proceedings of the 5th International Conference on Intelligent Systems, Reno, Nevada, 1999.
- [2] Dasgupta D., Ji Z., Gonzalez F., « Artificial Immune System (AIS) research in the last five years », The 2003 Congress on Evolutionary Computation, CEC '03, vol.1, 2003, p. 123-130.
- [3] Derbala.Ali .Ordonnancement des ateliers. Université de Blida,2005.
- [4] Janeway.Travers.Walport.Shlomchik.2003. 2^{ème} édition.deboek. 2003. chapitre1, chapitre2.
- [5] Jonathan Timmis and Leandro N. de Castro. Artificial Immune Systems : A New Computational Intelligence Approach. Springer-Verlag, September 2002.7,13,14,16,19
- [6] Leandro Nunes de Castro and Fernando J. Von Zuben. The clonal selection algorithm with engineering applications. In *Artificial Immune Systems*, pages 36.39, Las Vegas, Nevada, USA, 8 2000.18, 19
- [7] Darmoul S., Pierreval H., Gabouj S.H., « Scheduling using Artificial Immune System metaphors: A review », IEEE International Conference on Service Systems and Service Management, Troyes, 25-27 Oct. 2006, France, p. 1150-1155.
- [8] Chandrasekaran . P. Asokan .S. Kumanan . T. Balamurugan . S. Nickolas Solving job shop scheduling problemsusing artificial immune system. Int J Adv Manuf Technol. 2006.
- [9] Pezzella F, Merelli E (2000) A tabu search method guided by shifting bottleneck for the job shop scheduling problem. Eur J Oper Res 120:297–310
- [10] Engin O, Doyen A (2004) A new approach to solve hybrid flow shop scheduling problems by artificial immune system. Future Generation Comput Syst 20: I OR3-1 095

- [11] Zheng H, Zhang J, Nahavandi S (2004) Learning to detect texture objects by artificial immune approaches. *Future Generation Comput Systems* 20:1197–1208
- [12] Huang S (1999) Enhancement of thermal unit commitment using immune algorithms based optimization approaches. *Electr Power Energy Syst* 21:245–252
- [13] Attux RRF, et al (2003) A paradigm for blind IIR equalization using the constant modulus criterion and an artificial immune network. *IEEE XIII Workshop on Neural Networks for Signal Processing*, pp 839–849
- [14] May P, Mander K, Timmis J (2003) Mutation Testing: An Artificial Immune System Approach. Computing Laboratory. University of Kent, Canterbury, Kent, CT2 7NF.
- [15] De Castro LN, Von Zuben FJ (1999) Artificial immune systems, Part I. basic theory and applications. Technical Report, TR-DCA 0 1/99.
- [16] Messaoudi N, Bataoui H. Etude et resolution d'un problème d'ordonnancement de type JOB SHOP. 2001, p 50,65-66.
- [17] Jensen Mikkel T., Robust and Flexible Scheduling with Evolutionary Computation, Ph.D. Thesis, University of Aarhus, Denmark, 2001.
- [18] Balas E, Vazacopoulos A (1994) Guided local search with shifting bottleneck for job shop scheduling. Tech Rep Management Science Research Report MSRR-609, GSIA Carnegie Mellon University, Pittsburgh
- [19] J.E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069-1072,1990. Disponible par l'anonyme ftp dans [mscmga.ms.ic.ac.uk/pub](ftp://mscmga.ms.ic.ac.uk/pub).
- [20] E.Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64 :278-285,1993.

[21] Lawrence S (1984) Supplement to resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques. Tech Rep, GSIA, Carnegie Mellon Univ

[22] RH. Storer, S.D. Wu, R. Vaccari (1992), New search spaces for sequencing instances with application to jobshop scheduling, *Management Science* 38, 1495-1509.

[23] Applegate D, Cook W (1991) A computational study of the job shop scheduling problem. *ORSA J Comput* 3:149-156

[24] H. Fisher, G.L. Thompson (1963), Probabilistic learning combinations of local jobshop scheduling rules, J.F. Muth, G.L. Thompson (eds.), *Industrial Scheduling*, Prentice Hall, Englewood Cliffs, New Jersey, 225-251

[25] Adams J, Balas E, Zawack D (1988) The shifting bottleneck procedure for job shop scheduling. *Manage Sci* 34:391-401

[26] Takeshi Yamada et Ryohei Nakamo. Scheduling by genetic local search with multi-step crossover. In *vers96*, pages 960969,1996.