

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingéniorat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Automatique
Spécialité : Automatique et
formatique industrielle

Mémoire

Présenté en vue de l'obtention du Diplôme de Master
Thème:

**OPTIMISATION D'UNE CELLULE DE PRODUCTION
FLEXIBLE PAR UNE METHODE HYBRIDE
SEQUENCIENTIELLE**

Présenté par : *Métiri Khouloud*

Encadrant : *Khaldouna Zahia*

MCB

UBMA

Jury de Soutenance :

Lakel Fateh	MCA	UBMA	Président
Khaldouna Zahia	MCB	UBMA	Encadrant
Debbache Nacer	Professeur	UBMA	Examineur

Année Universitaire : 2020/2021

Remerciements

Nous remercions en premier lieu Dieu, le tout puissant, pour nous avoir donné, le courage, la patience, la volonté et la force nécessaires, pour affronter toutes les difficultés et les obstacles, qui se sont hissés au travers de notre chemin, durant toutes nos années d'études..

Nous tenons à remercier tout particulièrement notre encadreur Mlle Z.Khaldouna pour l'assistance qu'elle nous a témoigné, et surtout pour sa gentillesse tout au long de ce travail, qu'elle trouve ici l'expression de notre gratitude pour ses conseils.

Nous adressons également nos plus vifs remerciements à Monsieur N. Debbache ainsi à Monsieur F.Lakel pour l'honneur qu'ils nous ont fait en acceptant de juger ce travail.

.

En fin, nous tenons à remercier vivement chère maman, son oublier tous ceux qui nous ont aidé de près ou de loin à élaborer ce projet, ainsi tous les collègues de notre promotion et tous notre amis pour nous avoir accompagné jusqu'à son aboutissement.

Dédicace

Je dédie ce mémoire à :

ma mère, et mes sœurs.

A mes amis, en particulier, à ceux que je connais depuis mon enfance, à ceux de la fac pour leur ambiance et leur fâcheuse habitude de ne jamais être sérieux.

Incontestablement, à notre encadreur qui a fait des grands efforts pour accomplir ce modeste travail,

A tous ceux qui me porte dans leur cœur.

Metiri Khouloud

Sommaire

Résumé

Abstract

ملخص

Liste des figures

Liste des tableaux

Introduction générale 1

Chapitre I : Le système de production flexible et leurs caractéristiques

I	Introduction.....	2
I.1	Définitions des systèmes de production.....	2
I.2	Qu'est-ce que la flexibilité.....	3
I.3	Classification des ateliers de production flexible.....	3
I.3.1	Flexibilité statique /dynamique.....	3
	a. La flexibilité statique.....	4
	b. La flexibilité dynamique.....	4
I.3.2	Flexibilité de réponse /d'initiative.....	4
I.3.3	flow-shop/jop-shop.....	4
I.4	Les différentes formes de flexibilité.....	4
I.4.1	Le travail ou les ressources humaines.....	5
I.4.2	Les technologies : la productique.....	4

I.5	L'automatisation.....	5
I.6	Les ateliers de production flexible.....	6
I.6.1	Les entités	6
I.6.2	Les ressource.....	6
I.6.3	Les tâches.....	6
I.6.4	Contrainte.....	6
I.6.5	Gamme d'opération.....	7
I.6.6	Contrôle.....	7
I.7	Conclusion.....	7

Chapitre II : Les réseaux de pétri

II.1	Introduction réseaux de Pétri.....	8
II.2	Notions de base.....	8
II.2.1	Places, Transitions et Arcs.....	8
II.2.2	Marquee.....	9
II.2.3	Franchissement d'une transition.....	10
II.2.4	Matrice d'entrée et matrice de sortie.....	11
II.3	Règles d'évolution d'un RdP	12
II.4	Principe de modélisation par réseau de Pétri.....	13
II.5	Structures particulières.....	14
II.5.1	Graphe d'état.....	14
II.5.2	Graphe d'événement.....	15

II.5.3	Réseau de pétri sans conflit.....	15
II.5.4	Réseau de pétri à libre choix.....	15
II.5.5	Réseau de pétri simple.....	15
II.5.6	Réseau de pétri pur.....	16
II.6	Méthode d'analyse d'un RdP.....	16
II.6.1	Analyse matricielle et Equation fondamentale.....	16
II.6.2	Analyse par reduction.....	17
II.7	Classification des réseaux de Pétri.....	17
II.7.1	Réseaux de Pétri autonomes.....	19
II.7.1.1	Réseau de Pétri ordinaire	19
II.7.1.2	Réseau de Pétri généralisé.....	19
II.7.1.3	Réseau de Pétri à capacité.....	20
II.7.1.4	Réseau de Pétri à prédicats.....	20
II.7.1.5	Réseau de Pétri coloré.....	21
II.7.1.6	Réseau de Pétri à arcs inhibiteur.....	21
II.7.1.7	Réseau de Pétri à priorité.....	22
II.8	Réseaux de pétri non autonomes.....	22
II.8.1	Réseaux de Pétri synchronisés.....	23
II.8.2	Réseaux de Pétri Temporisés.....	23
II.8.2.1	Réseaux de Pétri P-Temporisés.....	23
II.8.2.2	Réseaux de Pétri T-Temporisés.....	24
II.9	Les réseaux de Pétri stochastiques.....	24

II.10	Réseaux de Pétri continu et modèle hybride.....	24
II.11	Conclusion.....	27

Chapitre III : Les méthodes de résolution des problèmes d'optimisation

III.1	Introduction.....	28
III.2	Les méthodes exactes.....	28
III.2.1	Introduction.....	28
III.2.2	la mthod branch and bound.....	29
III.2.3	Recherche exhaustive.....	30
III.2.4	backtracking	30
III.3	Les méta-heuristiques.....	30
III.3.1	Introduction.....	30
III.3.2	Principe d'élaboration des métaheuristiques.....	31
III.3.2.1	Introduction.....	31
III.3.2.2	Mémoires.....	31
III.3.2.3	Construction de solutions.....	32
III.3.2.4	Voisinages.....	32
III.3.2.5	Stratégies.....	32

III.4	Les différentes métaheuristiques.....	33
III.4.1	La méthode du recuit simulé.....	33
III.4.2	Algorithmes gloutons.....	34
III.4.3	La méthode recherche tabou.....	36
III.4.4	Méthodes des algorithmes génétiques.....	37
III.5	Les méthodes hybrides.....	41
III.5.1	Introduction.....	41
III.5.2	La méthode grasp.....	42
III.5.2.1	Définition.....	42
III.5.2.2	Fonctionnements.....	42
III.5.2.3	Les avantages et les inconvénients.....	43
III.6	Méthode hybride séquentielle proposée.....	43
III.7	Conclusion.....	44

Chapitre IV : Résultats et conclusions

IV.1	Introduction.....	45
IV.2	Simulation du modèle de conduite.....	45
IV.2.1	Introduction.....	45
IV.2.2	Simulation par l'équation du marquage.....	46
IV.3	Algorithme de simulation.....	47
IV.3.1	Composition de la cellule de production flexible.....	48
IV.3.2	Plan des charges.....	49
IV.4	Conduite du système de production flexible.....	49
IV.4.1	Problématique.....	49
IV.4.2	Algorithme de conduite.....	51
IV.5	Résultats.....	55
Comparaison entre les deux méthodes.....		59
IV.6	Conclusion.....	59
	Conclusion générale	61
	Bibliographie.....	62

Résumé

Ces dernières années, la "science des produits" a occupé une place importante et de nombreux travaux de recherche sur la gestion flexible des systèmes de production ont été effectués. C'est dans cette recherche que nous avons mené ce travail. L'un des aspects les plus importants de la gestion de la production est "l'optimisation des coûts et du temps de production".

Le système de production flexible gère différents types de pièces (matières premières ou produits semi-finis), et chaque pièce passe par une série d'opérations appelées filières de production. Lors du démarrage de séquences de production en parallèle, différents périmètres peuvent se trouver en conflit avec l'accès à différentes ressources (postes de travail, véhicules de transport, stockage, etc.). Le rôle de la fonction de planification est d'organiser dans le temps, d'accéder aux ressources partagées et d'évaluer les séquences de conflits pour trouver la « meilleure » tâche à allouer aux ressources. Pour cela, nous avons besoin d'une fonction d'évaluation, d'un critère d'optimisation et d'un programme d'exploration de l'espace des solutions. Lorsque l'échelle du problème est suffisamment grande, des méthodes d'optimisation précises ne peuvent pas résoudre le problème dans un temps de réponse confortable. D'un autre côté, les méthodes heuristiques peuvent rapidement approcher les « meilleurs » résultats précis. Leur inconvénient est qu'il existe des programmes spécifiques développés séparément pour chaque type de problème qu'ils traitent. Afin de résoudre ces problèmes, les méta-heuristiques ont commencé à inverser ce processus. Étant donné un algorithme général connu (recherche tabou, algorithme génétique, recuit simulé, réseau de neurones, colonie de fourmis...). Ces méthodes sont désormais considérées comme le point de départ pour le développement de nouveaux algorithmes d'optimisation ; ce sont des méthodes hybrides ; parmi les méthodes qui ont émergé, nous utiliserons la méthode hybride séquentielle dans l'optimisation du système de production. Nous étudions le principe de la méthode et l'ajustons en fonction de notre problème, tout en respectant la méthodologie de base de l'algorithme. La norme d'optimisation que nous retenons est de minimiser le temps de réalisation de l'ensemble du procédé « MAKESPAN » sous la contrainte de maximiser la productivité. La fonction d'évaluation est basée sur l'équation d'évolution R.d.P, sachant que le système à l'étude est modélisé par la R.d.P temporisée.

Abstract

In recent years, "production" is an important, many research on the management of flexible manufacturing systems have been made. It is in this research we have conducted this work, one of the most important aspects in the management of production is: "The optimization of costs and production delays".

A system of flexible production process different types of parts (raw material or semi-finished product), each part undergoes a sequence of operation, called product range. When the sequences of execution are launched in parallel, different ranges may be in conflict over access to different resources (workstations, transportation, storage ...). The role of the scheduling function is to organize in time, access to shared resources, assess the conflict sequences in order to find the "best" task to be assigned to the resource. To do this we need an evaluation function, an optimization criterion and a procedure for exploring the solution space. When the problem size is large enough, the exact optimization methods are unable to resolve the problem in response time comfortable. Heuristics by cons approaching the exact result "Optimum" fairly quickly. Their disadvantage lies in the fact that specific procedures were developed individually for each type of problem addressed. To overcome these problems, metaheuristics have come back in again. Given a universal algorithm known (Tabu search, genetic algorithms, simulated annealing, neural networks, ant colonies ...). These methods are now seen as starting points for developing new optimization algorithms are hybrid methods, among the methods that we saw the day, the method hybride séquentielle , we will use in our optimization system production. We study the principles of the method and we adapt to our problem, while respecting the basic methodology of the algorithm. The optimization criterion we used is the minimization of completion time of the entire process, "Makespan" under the constraint of maximizing productivity.

The evaluation function is based on the equation of evolution of Petri nets, knowing that the studied system is modeled by timed Petri nets.

ملخص

في السنوات العشرين الأخيرة من القرن العشرين ظهرت في أوروبا, آليات جديدة لتطوير و تحسين أنظمة الإنتاج الصناعي. سميت هذه المنظومة المتكاملة من الآليات "Productique": وإذا بحثنا في مختلف المراجع لا نجد ترجمة حرفية أو مقابلا لهذه الكلمة فمثلا في الإنجليزية نجد " Production managment " أما في اللغة العربية فبإمكاننا أن نترجمها كما يلي: " آليات تحسين أنظمة الإنتاج الصناعي."

فهذا النوع من ورشات الانتاج, " F.M.S " يحتوي على آلات ذات تحكم رقمي, متعددة الوظائف تنجز أعمالا مختلفة على مواد خام متعددة و مختلفة. ولذلك تتطلب نظام تحكم متطور قادر على توزيع المهام بين مختلف وسائل الإنتاج بطريقة مثلى تحقق أقصى المعدلات في زمن مقبول و بأسعار معقولة. و لتحقيق هذا التوازن الثلاثي "إنتاجية- زمن- تكلفة" لا نستطيع تطبيق المناهج الرياضية الدقيقة لما تتطلبه من وقت طويل و حسابات معقدة. ولهذا عادة ما يلجأ الباحثون في الإنتاج الصناعي إلى طرق أقل دقة و تعقيدا و لكنها تعطي نتائج مقبولة في زمن معقول.

ومن مساوئ هذه الطرق أن لكل مشكلة طريقة حل خاصة بها لا يمكن تعميمها بل يجب تغييرها لو تغيرت معطيات المشكلة نفسها. وللتغلب على هذا المشكل, فكر الباحثون في إستعمال مناهج أكثر تعميما مستنبطة من ظواهر طبيعية, بيولوجية و فيزيائية معروفة مثل: " البحث بالطابوهات, نظام الوراثة, إعادة الإحياء المصطنع, مستعمرات النمل, شبكات العصبونات, قفزة الكنغر...". وينظر إلى هذه الطرق الآن كنقاط بداية لتطوير خوارزميات التحسين الجديدة ونحن بصدد الحديث هنا عن الطرق الهجينة.

من بين الأساليب التي شهدناها اليوم ، الطريقة hybride séquentielle حاولنا تطبيق هذه الطريقة في توخي التوزيع الأمثل للوظائف المطلوب إنجازها من طرف مجموعة من وسائل الإنتاج (آلات رقمية للتصنيع, رجل نقل آلي, مساحات التخزين...) علما أننا إعتدنا على شبكات بيثري كنموذج لتمثيل ورشة إنتاج صناعي مرن إختارناها كمثال للتطبيق.

وفي الأخير قمنا بدراسة النتائج المتحصل عليها باستعمال هذه الطريقة.

Liste des figures

Chapitre II

Figure 2-1 :	Un réseau de Pétri (place transition arc)	9
Figure 2-2 :	Réseau de Pétri à arcs multiples.	9
Figure 2-3 :	Exemple de marquages	10
Figure 2-4 :	Franchissement d'une transition	10
Figure 2-5 :	Les valeurs W_{ij}^- et W_{ij}^+	11
Figure 2-6 :	Exemple d'un RdP ordinaire.	11
Figure 2-7 :	Exemple d'évolution d'un Rdp.	13
Figure 2-8 :	Exemple de réseau de Pétri étiqueté.	14
Figure 2-9 :	Graphe d'état et graphe d'événement.	15
Figure 2-10 :	Réseau de pétri sans boucle	16
Figure 2-11 :	Exemple de simplification d'une séquence	17
Figure 2-12 :	Exemple de fusion des chemins parallèles	18
Figure 2-13 :	Exemple d'élimination des transitions identité	18
Figure 2-14 :	réseau de pétri autonome.	19
Figure 2-15 :	Exemple de réseau de pétri à capacité.	20
Figure 2-16 :	Exemple de RdP à prédicats.	20
Figure 2-17 :	Exemple de RdP à arcs inhibiteur.	21
Figure 2-18 :	Exemples de Réseaux de Pétri T-Temporisés	23
Figure 2-19 :	Principe du réseau de Pétri continu à vitesse variable	25

Chapitre III

Figure 3-1 :	Organigramme général de l'algorithme recherche tabou	36
Figure 3-2 :	Les principales étapes d'un algorithme génétique	38
Figure 3-3 :	Organigramme de l'algorithme génétique	39

Chapitre IV

Figure 4-1 :	Comparaison entre les deux méthodes par rapport au taux de qualité	56
Figure 4-2 :	le MAKESPAN pour les deux méthodes	57
Figure 4-3 :	nombre de pièce obtenue par l'application des deux méthodes	57
Figure 4-4 :	Fréquence d'exécution de l'algorithme pour les deux méthodes	58
Figure 4-5 :	temps moyen de production des deux méthodes	58

Liste des tableaux

Chapitre IV

Tableau 2-1 : La méthode hybride (l'algorithme génétique)

Tableau 2-2 : : La méthode de recherche tabou

Introduction générale :

L'intensification de la concurrence mondiale et la montée du progrès technologique sont des facteurs importants qui incitent l'industrie manufacturière à classer les outils de production comme ayant une productivité élevée et pouvant s'adapter à l'évolution permanente du marché. Cela a conduit à la recherche d'une plus grande flexibilité dans l'atelier de production. Un système de production flexible doit pouvoir s'adapter aux changements de politique et répondre aux défaillances des usines pour assurer la sécurité du personnel et des équipements, tout en continuant à répondre aux exigences de production. Par conséquent, la maîtrise de l'aspect temps réel est essentielle. Nous devons chercher à apporter des solutions capables de s'adapter aux nouveaux besoins de production. Pour cela, il est nécessaire d'utiliser des méthodes et des outils communs, et de concevoir le système de contrôle selon une approche globale et modulaire. Notre travail propose l'utilisation de MOP pour modéliser une unité de production flexible, puis nous proposons de découvrir et d'évaluer l'efficacité des méthodes d'optimisation appliquées au problème d'ordonnement de l'unité. Le but de cette méthode est de déterminer l'ordonnement optimal de la plage de fonctionnement. Nous nous intéressons donc à une classe d'algorithmes qui n'utilisent pas de formalisme mathématique, ce sont les méta-heuristiques. Nous aborderons principalement l'une d'entre elles, très importante dans le processus d'optimisation : les méthodes hybrides séquentielle

Notre mémoire est organisé de la manière suivante :

Le premier chapitre : donne un aperçu global sur les spécifications et caractéristiques des ateliers de production flexible.

Le 2^{ème} chapitre : on aborde les notions de bases sur les réseaux de pétri et leur extensions.

Le 3^{ème} chapitre: se focalise sur les méthodes de résolution des problèmes d'optimisation combinatoire.

Le dernier chapitre: regroupe les résultats de simulation obtenus en associant la méthode de l'algorithme génétique et la méthode de Recherche Tabou à un exemple de simulation, en suite on les compare, et à la fin on donnera une critique sur la validité de la méthode hybride séquentielle

CHAPITRE I

Systeme de production flexible et
caractéristique

I. Introduction

Nous avons fait part dans l'introduction générale de notre intention de proposer un modèle de simulation permettant d'évaluer les performances d'un système de production a fin d'apporter des éléments aux prises de décision, des gestionnaires de ces derniers. Avant de proposer un tel outil il nous paraît important de présenter les systèmes de production, dans leur complexité et dans leur diversité afin de circonscrire notre cadre de travail.

Nous avons donc structuré ce chapitre de la manière suivante :

Au premier lieu nous présentons des généralités sur les systèmes de production (définition, typologie, classification)

Ensuite nous nous intéressons à la notion de flexibilité dans un système de Production

I.1. Définitions des systèmes de production

Un système de production est un ensemble de ressources qui concourent à la fabrication d'un bien. La fabrication d'un bien se fait par le biais de diverses opérations telles que : des opérations manuelles, d'usinage et/ou d'assemblage. L'assemblage est l'action de fabriquer des objets par agrégation d'objets plus simples. En plus de l'assemblage et de l'usinage, il faut aussi, noter les activités de transport et de stockage. Un système de production est présenté par comme une entité composée de :

- Un système physique : il comprend l'ensemble des ressources à savoir les personnes, les machines ou postes de travail, les produits en cours de production, les matières premières et les moyens de transport.
- Un système d'information : le système d'information prend en compte toutes les données permettant de décrire la structure physique (nombre de machines, types de ressources etc.) ,les processus de fabrication, la nature des produits. Le système d'information permet aussi de stocker des informations relatives au déroulement des activités de production et permet de faire une évaluation globale des systèmes de production. Dans les entreprises le système d'information est le plus souvent géré par les progiciels de gestion intégrée
- Un système décisionnel : le système décisionnel comprend l'ensemble des actions qui permettent de conduire les activités de production, de maintenance ou même de contrôle dans un système manufacturier. Ces actions peuvent être définies à travers des règles, des procédures et ont un impact très important sur les performances de ce dernier. Le système

décisionnel englobe les politiques de gestion des stocks et d'ordonnancement. Le système décisionnel s'appuie le plus souvent sur le système d'information pour fonctionner.

Dans notre domaine d'étude sur les systèmes de production Manufacturiers, il nous semble important de faire un rappel sur les définitions de certaines notions telles que : Un système de production, les postes de travail, les produits et les matières premières, les dispositifs de stockage, et les moyens de manutentions. Ces différents éléments rentrent dans la construction d'un système de production.[1]

I.2. Qu'est-ce que la flexibilité

- Définition

- La flexibilité est définie comme la capacité d'adaptation à tout moment à des variations de la demande en nature (produit ou service) et en volume.
- La flexibilité d'un système est son aptitude à se transformer pour améliorer son insertion dans l'environnement et accroître ainsi sa probabilité de survie. En cas de fortes turbulences dans l'environnement, les entreprises dotées d'une grande flexibilité s'adapteront de manière à maintenir des niveaux satisfaisants de performance alors que les plus rigides seront menacées de rupture fatale.
- Il faut que le décideur puisse à tout moment reconsidérer ses choix de manière à obtenir l'optimalité de sa décision.

Il existe deux manières de décider :

- Prendre une décision car on pense connaître tous les éléments pour le faire.
- Prendre une décision sans connaître toutes les informations. On est alors en situation de risque, en situation d'incertitude.
- Les notions liées à la flexibilité sont donc :
 - Capacité à changer rapidement le produit réalisé.
 - Capacité à produire efficacement, en petites quantités chacune des versions d'un produit.
 - Capacité à accepter un changement dans la conception d'un produit.

I.3 Classification des ateliers de production flexible

I.3.1 Flexibilité statique /dynamique

a. La flexibilité statique est instantanée, potentielle et se présente à tout moment. Elle concerne un environnement connu et se caractérise par l'emploi de ressources supplémentaires (heures supplémentaires, intérim, surcapacités) afin de répondre aux variations de l'environnement. [1]

Ainsi, dans les branches d'activité confrontées à la saisonnalité, cette surcapacité permet alors d'éviter des coûts de stockage excessifs. [1]

b. La flexibilité dynamique est séquentielle et la variable stratégique pour l'entreprise est le délai de réaction. Elle s'oriente davantage vers le développement de nouvelles formes d'organisation susceptibles de réduire les délais de réaction. La flexibilité dynamique correspond à l'aptitude qu'a l'entreprise de s'adapter à son environnement externe en maîtrisant au mieux les temps propres aux diverses opérations et à leurs enchaînements qui sont les éléments constituant la dimension temporelle de la production. [1]

I.3.2 Flexibilité de réponse /d'initiative

La flexibilité de réponse correspond à la capacité de réagir de façon continue dans le temps aux variations de l'environnement.

La flexibilité d'initiative repose sur la capacité de création et d'anticipation par rapport à l'environnement des produits et/ou des méthodes de fabrication et de commercialisation. C'est donc la capacité de l'entreprise à façonner son environnement. [1]

I.3.3 flow-shop/jop-shop

- **Le job shop** (atelier à cheminement multiple)
 - Ce sont des ateliers caractérisés par une main d'œuvre très qualifiée et polyvalente, utilisant des moyens de production peu coûteux, peu spécialisés et peu automatisés.
 - Ces ateliers répondent à une logique de production à flux tirés, ainsi, les produits sont très diversifiés. Marquées par un fort degré d'incertitude, les opérations de production sont longues, complexes et peu formalisées. [2]
- **Le flow-shop** (atelier à cheminement unique) : tous les produits à fabriquer dans cet atelier ont la même gamme de production (même séquence de machines). [2]

I.4. Les différentes formes de flexibilité

La flexibilité intervient dans tous les domaines de l'entreprise. Ceux-ci sont interdépendants :

I.4.1 Le travail ou les ressources humaines

Il faut libérer les contraintes qui pèsent sur l'embauche et le licenciement.

I.4.2 Les technologies : la productique

La productique est l'ensemble des techniques informatiques et automatiques visant à accroître la productivité :

I.5 L'automatisation

- C'est le développement de la science de l'automate. Les machines sont conçues pour contrôler, se réguler et s'arrêter en cas de panne afin d'éviter la diffusion du problème dans les autres ateliers.
- L'automatisation se traduit par l'apparition, à la fin des années 60, de machines-outils à commandes numériques (MOCN) et de robots capables de réaliser plusieurs tâches avec des temps d'adaptation réduits, par le développement de systèmes automatiques de convoyage des pièces afin de réduire les temps de transport et la main d'œuvre affectée à ce travail.
- Les MOCN et les machines-outils conventionnelles vont coexister. Puis, à la fin des années 70, arrivent des centres d'usinages capables de faire presque toutes les opérations de transformation de base, voire exécuter une pièce complète.
- L'automatisation est un bon outil pour acquérir une certaine flexibilité, mais il ne faut pas automatiser à outrance, au risque de perdre en productivité.
- Depuis Skinner, les politiques de production reposaient sur des choix entre 2 objectifs conflictuels. On privilégiait la minimisation des coûts obtenus généralement par réduction d'incertitude, de diversité et par automatisation, ou on choisissait de développer des capacités et aptitudes à maîtriser la diversité et l'incertitude. Productivité et flexibilité apparaissaient comme conflictuelles et exclusifs. Mais avec le développement des technologies de l'information et de l'informatisation, ces notions sont aujourd'hui dépassées.

- L'informatisation

- Elle est utilisée pour manipuler un grand nombre d'informations ou pour apporter une aide à la décision dans le système de gestion.

- Pour cela, l'informatique traite d'une part, des informations statiques dont l'évolution est lente, tels que les produits à fabriquer, les moyens de production, les gammes de fabrication et d'autre part, des informations dynamiques à savoir rapidement évolutives, tels que les commandes clients, l'approvisionnement, les encours de fabrication...

I.6 Les ateliers de production flexible

Les entreprises industrielles sont des systèmes complexes décomposables en plusieurs fonctions essentielles, toutes interdépendantes. Parmi toutes ces fonctions, la fonction production occupe, du moins dans les entreprises de transformation, une place essentielle. Elle conduit ainsi à la création du produit fini propre à la distribution vers le (les) clients (s) de l'entreprise.[12]

La fonction production peut être décomposable en :

I.6.1 Les entités :

L'entité est l'objet directement manipulé par la fonction production. Suivant les étapes de la production il peut donc s'agir de matière première, de composants, de produits semi-finis ou de produits finis. [12]

I.6.2 Les ressource :

Les ressources du système de production sont l'ensemble des moyens ou des composants mis en œuvre pour transformer les entités et passer de la matière première au produit fini. [12]

I.6.3 Les tâches :

Le concept de tâche est intimement lié aux deux notions précédentes. En effet, une tâche est l'application directe d'une ressource à une entité. [12]

I.6.4 Contrainte :

C'est une condition qui restreint les degrés de liberté disponibles pour l'ordonnancement des tâches. Elle peut être de nature et d'origine différente. [12]

On distingue les contraintes :

- **Organisationnelles** : échéances, en cours, coût, quantité de ressources, productivité
- **Physiques** : capacité de stock, temps de dépannage, temps de transport....

- **Causals** : relation de précédence, type de ressources propres à une tâche....
- **Disponibilité des ressources** : panne, maintenance....

I.6.5 Gamme d'opération

C'est une séquence de traitements qui définit un type de produit ou un type de tâches
Un atelier flexible peut produire plusieurs unités du même type de produit ou de types différents.[10]

I.6.6 Contrôle

Il est effectué suite a un échec issu, soit d'un ensemble de décisions ayant conduit à une incohérence globale, soit de l'autocritique d'une validation effectuée au paravent, dans un contexte différent de celui donné par l'état courant du système. Il conduit à une remise en cause des contraintes, dans le cadre d'une stratégie à suivre si plusieurs possibilités se présentent. [10]

I.7 Conclusion

Dans l'ensemble les modes de production sont indéniablement flexibles .Cependant des problèmes se posent au niveau de leur application et de leur organisation dans l'entreprise. Tous les modes de gestion de production ne sont pas applicables dans tous les secteurs ou même plus simplement dans toutes les entreprises. On a retrouvé souvent des problèmes d'adaptation à l'entreprise c'est-à-dire les clients, les fournisseurs ainsi que des problèmes purement informatiques.

Pouvons- nous penser qu'un développement des technologies de l'information puisse nous aider à mieux gérer la technique de production d'aujourd'hui ?

Chapitre II

Les réseaux de pétri

II.1. Introduction réseaux de Pétri

La théorie des réseaux de Pétri est apparue en 1962, dans la thèse du chercheur Carl Adam Pétri, intitulée : "Communication avec les automates", présentée à l'université de Darmstadt en Allemagne. Ensuite un groupe de chercheurs dirigés par Anatol W. Holt du Massachusetts Institute of technology MIT, fut séduit par ce travail. C'est ce groupe qui jettera les bases, entre 1968 et 1976 de ce qui devint les réseaux de Pétri. Ils constituent un puissant outil graphique et mathématique de représentation de phénomènes complexes et de mécanismes séquentiels. Ils sont largement utilisés pour l'analyse et la modélisation des systèmes à événements discrets. Ils permettent la modélisation des processus complexes mettant en jeu des phénomènes de synchronisme et de choix.

Parmi les applications des réseaux de pétri, on peut citer : l'évaluation des performances systèmes automatisés, les protocoles de communication, la commande des ateliers de production, organisation des entreprises et les interfaces homme-machine.

Depuis les années 70, les travaux sur les RdP ont connu un essor considérable. Aujourd'hui, la littérature sur les RdP est fort riche, et comprend en particulier de nombreux résultats analytiques.

II.2. Notions de base

II.2.1 Places, Transitions et Arcs

Un RdP est un graphe orienté R contenant deux types de nœuds : les places représentées par les cercles et les transitions représentées par des traits. Ces nœuds sont connectés par des arcs orientés. Mathématiquement un RDP peut être représenté par le triplet $R = \langle P ; T ; W \rangle$ où

- P : ensemble fini non vide des places.
- T : ensemble fini non vide des transitions.

Avec P et T sont des ensembles finis disjoints,

- W : matrice d'incidence ;

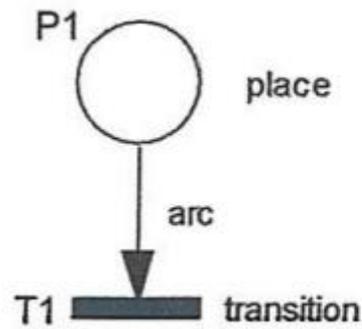


Figure 2.1 : Un réseau de Pétri (place transition arc)

On dira qu'une transition est en amont (entrée) ou en aval (sortie) d'une place. Une transition sans place d'entrée est une transition source. Une transition sans place de sortie est une transition puits. [10]

Une place peut être reliée par plusieurs arcs à une transition. Ces arcs multiples peuvent être représenté par un arc simple affecté d'une pondération égale au nombre d'arcs multiples (Figure 2.2). Par défaut les arcs non pondérés sont considérés comme ayant un poids égal à 1. [10]

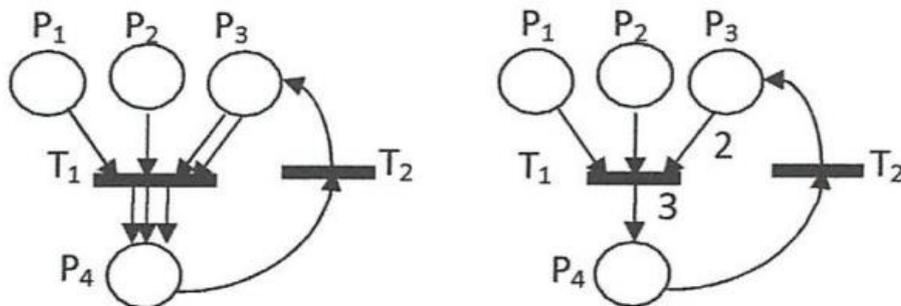


Figure 2.2 : Réseau de Pétri à arcs multiples.

II.2.2. Marquage

Un réseau de Pétri marqué est un couple $\langle R; M_0 \rangle$ tel que:

→ R est un réseau de Pétri, M_0 est un marquage dit marquage initial.

Chaque place contient un nombre entier positif ou nul de marques ou jetons. Le marquage M définit l'état du système décrit par le réseau à un instant donné. C'est un vecteur colonne de dimension égal au nombre de places dans le réseau. Le i ème élément du vecteur correspond au nombre de jetons contenus dans la place P_i . [10]

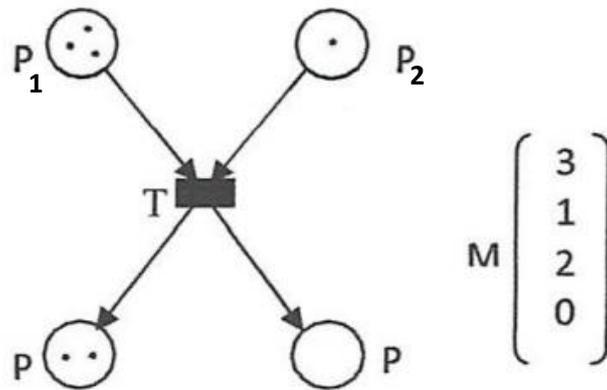


Figure 2.3: Exemple de marquages

II.2.3. Franchissement d'une transition

Une transition est franchissable (ou validée) lorsque toute place p_i située en amont de T_j contient un nombre de jetons au moins égal au poids attaché à l'arc allant de p_i vers T_j . Dans le cas d'un RdP ordinaire, il suffit que toutes les places d'entrée d'une transition contiennent au moins un jeton pour qu'elle soit franchissable. Une transition source et donc toujours validée.

Avant franchissement.[10]

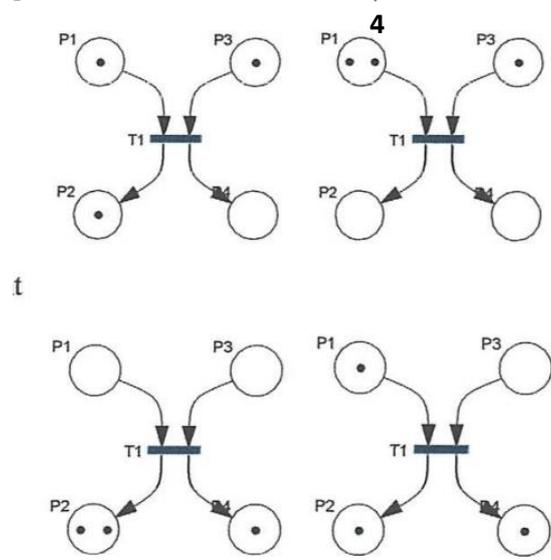
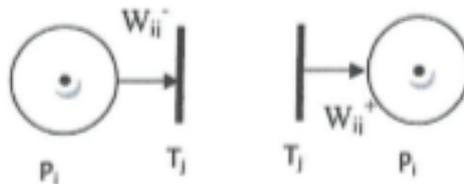


Figure 2.4 : Franchissement d'une transition**II.2.4 Matrice d'entrée et matrice de sortie**

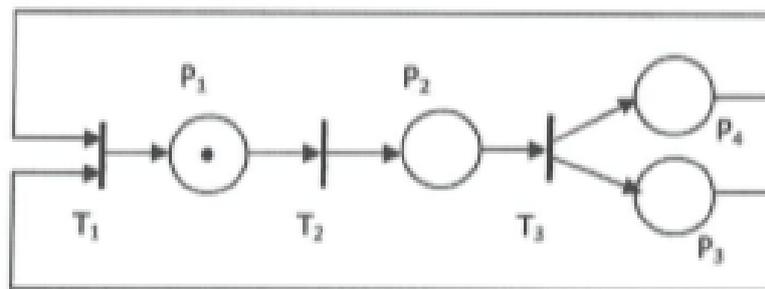
La structure du réseau de Pétri est décrite par deux matrices notées W_{ij}^- et W_{ij}^+ dans la Figure 2.5

**Figure 2.5** : Les valeurs W_{ij}^- et W_{ij}^+

La valeur W_{ij}^- (respectivement, W_{ij}^+) représente le poids de l'arc qui relie une transition T_j à une place p_i ; située en amont (respectivement, une transition T_j à une place p_i située en aval).

Si $W_{ij}^- = 0$ (respectivement, $W_{ij}^+ = 0$), alors, il n'existe pas d'arc reliant une transition T_j à une place p_i située en amont (respectivement, une transition T_j à une place p_i située en aval). W^- est la matrice d'incidence dite "avant ou pré", et W^+ est la matrice d'incidence dite "arrière ou post".

On définit la matrice d'incidence d'un RdP comme suit : $W = W^+ - W^-$; où W est la matrice d'incidence.[10]

**Figure 2.6** : Exemple d'un RdP ordinaire.

Comme exemple, considérons le cas de la figure 2.6, la matrice d'incidence est :

$$W = W^+ - W^- = \begin{matrix} & T1 & T2 & T3 \\ \begin{matrix} P1 \\ P2 \\ P3 \\ P4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} - \begin{matrix} & T1 & T2 & T3 \\ \begin{matrix} P1 \\ P2 \\ P3 \\ P4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

II.3 Règles d'évolution d'un RdP :

L'évolution du réseau de Pétri est définie par le déplacement des jetons à partir de l'état initial. Les jetons, qui matérialisent l'état du réseau à un instant donné peuvent passer d'une place à une autre par franchissement (ou tir) d'une transition. [10]

Les règles d'évolution sont les suivantes :

- Une transition est franchissable ou sensibilisée, lorsque chacune de ses places d'entrée possède un nombre de jetons au moins égal au poids de l'arc qui la relie à cette transition.
- Le franchissement d'une transition est un phénomène instantané, qui se traduit par les opérations indivisibles suivantes :

* On enlève de chaque place d'entrée de la transition, un nombre de jetons égal au poids de l'arc qui la relie à celle-ci.

* On ajoute à chaque place de sortie de la transition, un nombre de jetons égal au poids de l'arc qui la relie à celle-ci.

Ainsi, dans les cas de la figure 2. 7, après le tir de T_1 ensuite T_2 , les deux transitions T_3 et T_4 sont franchissables, et il est donc possible de faire évoluer le réseau par franchissement "au hasard" de l'une d'elles. Pour revenir à l'état initial, nous avons tiré deux fois la transition T_4 . [10]

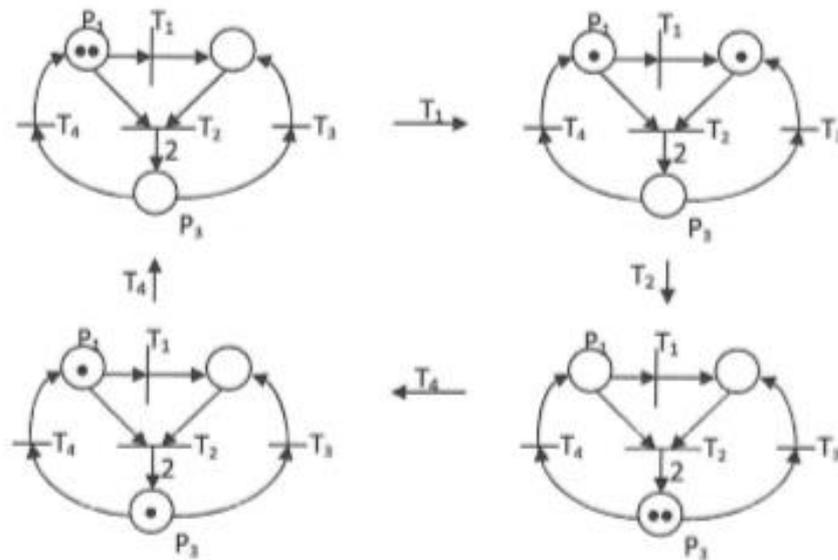


Figure 2.7 : Exemple d'évolution d'un Rdp.

II.4. Principe de modélisation par réseau de Pétri

L'évolution d'un système se traduit par des séquences d'actions qui sont effectuées lorsque certaines conditions internes ou externes sont réalisées.

Lorsqu'un système est modélisé par un réseau de Pétri, on adopte un certain nombre de conventions qui font correspondre les places, les transitions et les jetons aux fonctions et aux conditions du système, il existe de nombreuses façons différentes de modéliser un système par réseaux de Pétri. Ces variantes sont d'ailleurs justifiées par la nécessité d'arriver pour chaque problème à une représentation qui soit à la fois simple et fidèle. [10]

La méthode de modélisation la plus simple et la plus naturelle consiste à associer :

- les places aux actions (ou sorties) du système
- les transitions aux conditions (entrées) et/ou aux actions en exploitant les conditions normales de sensibilisation des transitions
- Les jetons aux conditions et/ou actions du système.

Compte tenu des propriétés des réseaux de Pétri, où le franchissement des transitions est supposé instantané, il est naturel de n'associer celles-ci que des actions de courtes durées. On aboutit ainsi à un réseau de Pétri étiqueté tel que celui de la figure (2.8), dans lequel chaque étiquette spécifie un état ou une action, mais sans qu'il y ait de conditions extérieures imposées pour le franchissement des transitions. [10]

Comme exemple, soit à modéliser l'exécution d'une tâche par un processeur. A première vue, on peut adopter le réseau de la figure (2.8) .a. Seulement, avec un tel modèle, nous n'avons pas pris en considération les deux contraintes suivantes :

- la transition est supposée être tirée instantanément,
- l'exécution de la tâche suppose que le processeur est préalablement libre.

Pour remédier à ces inconvénients, nous proposons la solution de la figure (2.8).



Figure 2.8 : Exemple de réseau de Pétri étiqueté.

Ce type de modélisation peut être considéré comme la caractérisation d'une suite d'événements qui peuvent se réaliser lorsqu'une série de pré conditions sont satisfaites, et qui produisent, par leur réalisation, une série de post conditions.

Comme autre type de modélisation, on peut associer les événements aux transitions et les conditions aux places, avec la convention que la condition associée à la place est réalisée s'il existe un jeton dans la place.

II.5. Structures particulières

II.5.1. Graphe d'état

Un réseau de pétri non marqué est un graphe d'état si et seulement si toute transition a exactement une place d'entrée et une place de sortie (voir figure 2.9)

Dans le cas d'un graphe d'état ordinaire marqué, le nombre de jetons est constant.

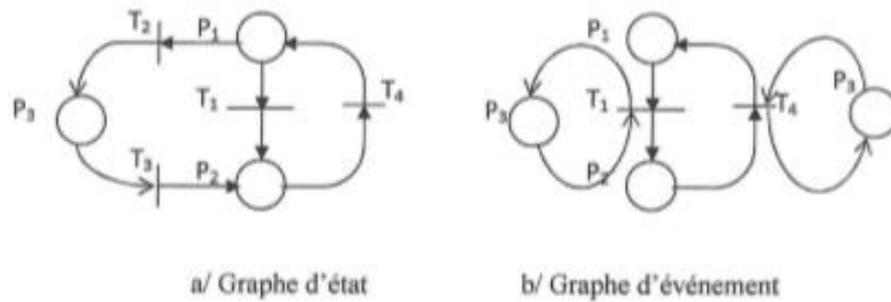


Figure 2.9 : Graphe d'état et graphe d'événement.

II.5.2. Graphe d'événement

Un réseau de pétri est un graphe d'événement si et seulement si toute place a exactement une transition d'entrée et une transition de sortie.

Un graphe d'événement est un dual d'un graphe d'état.

II.5.3. Réseau de pétri sans conflit

Un réseau de pétri sans conflit est un réseau de pétri dans lequel toute place a au plus une transition de sortie.

Un conflit (structurel) correspond à l'existence d'une place P_i , qui a au moins deux transitions de sortie T_j et T_k . On notera ce conflit par le doublet formé d'une place et d'un ensemble de transitions : $\langle P, T_j, T_k, \dots \rangle$. [12]

II.5.4. Réseau de pétri à libre choix

Un réseau de pétri à libre choix est un réseau de pétri dans lequel pour tout conflit $\langle P_i, \{T_j, T_k, \dots\} \rangle$ aucune des transitions T_j, T_k, \dots ne possède une autre place d'entrée que P_i .

Un réseau de pétri à libre choix étendu est un réseau de pétri dans lequel pour tout conflit $\langle P_i, \{T_j, T_k, \dots\} \rangle$ toutes les transitions T_j, T_k, \dots ont le même ensemble de places d'entrées. [12]

II.5.5. Réseau de pétri simple

Un réseau de pétri simple est un réseau de pétri dans lequel chaque transition ne peut être concernée que par un conflit au plus.

II.5.6. Réseau de pétri pur

Un réseau de pétri pur est un réseau de pétri dans lequel il n'existe pas de transition ayant une place d'entrée qui est également place de sortie de cette même transition.

Propriété :

Tout réseau de pétri impur peut être transformé en un réseau de pétri pur (figure 2. 10).

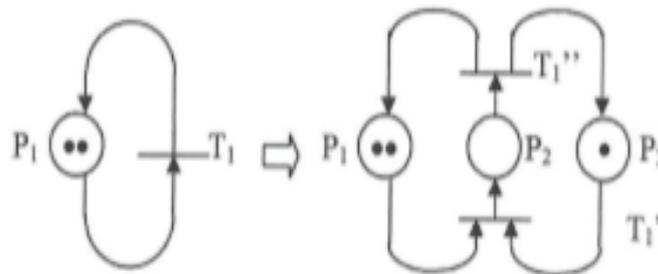


Figure 2.10 : Réseau de pétri sans boucle

Un réseau de pétri sans boucle est tel que s'il existe une place P ; qui est à la fois place d'entrée et place de sortie d'une transition T_j alors T_j a au moins une autre place d'entrée.

II.6. Méthode d'analyse d'un RdP

II.6.1. Analyse matricielle et Equation fondamentale

Une séquence de franchissement à partir d'un marquage M_0 est représentée par une suite de transitions. Si nous considérons le RdP de la figure II.6 et son marquage initial $M_0 = (1; 0; 0; 0)$, [12]

nous dirons que nous sommes passés de M_0 à $M_2 = (0; 0; 1; 1)$ en effectuant le tirage de la séquence $0 < T_2; T_3 >$ et nous écrivons $M_0 M_2$.

L'équation fondamentale qui caractérise l'évolution dynamique d'un RdP est:

$$M_k = M_0 + W \cdot D \cdot [12]$$

Où M_i est le marquage que l'on atteint à partir de M_0 après la séquence de franchissement D réalisable.

Une séquence de franchissement est une suite de transition $T_i; T_k$ qui peuvent être franchies successivement à partir d'un marquage donnée.

D est un vecteur qui représente cette séquence. Sa dimension est égale au nombre de transitions que contient le RdP.

Reprenons l'exemple de la figure 2.6. Le vecteur D est égal à (0; 1; 1)', ce que l'on peut interpréter de la manière suivante : T₂ et T₃ sont franchies chacune une fois pour passer de M₀ à M₂. La matrice d'incidence W est donnée par: [12]

$$W = W^+ - W^- = \begin{matrix} & \begin{matrix} T_1 & T_2 & T_3 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} - \begin{matrix} & \begin{matrix} T_1 & T_2 & T_3 \end{matrix} \\ \begin{matrix} P1 \\ P2 \\ P3 \\ P4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Le marquage M₂ peut être aussi calculé à partir des matrices M₀, D et W données ci-dessus et en appliquant l'équation M_k = M₀ + W.D, soit

$$M_2 = M_0 + W.S = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

II.6.2. Analyse par réduction

On est souvent amené à analyser un réseau pour déterminer ses propriétés et vérifier si elles sont bien conformes avec celles que doit présenter le système en cours de conception. Lorsque le réseau de pétri est complexe, il est en général intéressant de faire précéder l'analyse proprement dite par une phase de réduction qui conduit à un réseau plus simple et dont les propriétés sont inchangées.

La réduction peut être effectuée par la mise en œuvre systématique de trois règles simples qui portent sur la simplification d'une séquence, la fusion de chemins parallèles et l'élimination d'une transition identité. [12]

➤ La simplification d'une séquence

Si les arcs d'une séquence ont le même poids (figure 2.11) le bloc peut se réduire à une seule place comprise entre les deux transitions d'extrémités, sans que son évolution soit changée dans son principe

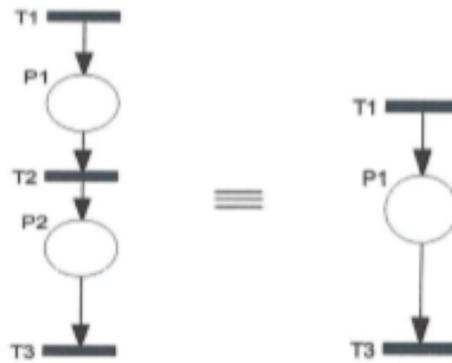


Figure 2.11 : Exemple de simplification d'une séquence

➤ **La fusion des chemins parallèles**

La fusion des chemins parallèles consiste à remplacer plusieurs chemins parallèles compris entre les deux transitions d'extrémités d'un bloc, par un seul chemin (figure 2.12).

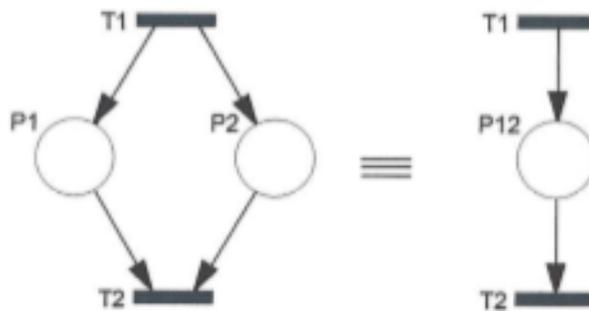


Figure 2.12 : Exemple de fusion des chemins parallèles

Cette fusion ne modifie pas l'évolution du réseau puisque les places correspondant aux différents chemins sont toutes dépendantes, de sorte que le marquage d'une des places, par exemple P1 dans la figure II.23, entraîne nécessairement le marquage des autres places, ici P2.

➤ **Elimination des transitions identité**

Une transition identité est telle qu'elle est sensibilisée si une autre transition l'est aussi et si son tir ne modifie pas le marquage.

Les transitions identité peuvent donc être supprimées sans modifier le marquage et l'évolution du réseau, comme indiqué sur l'exemple de la figure 2.13.



Figure 2.13 : Exemple d'élimination des transitions identité

II.7. Classification des réseaux de Pétri

II.7.1 Réseaux de Pétri autonomes

Un Réseau de Pétri autonome est un réseau de Pétri qui décrit le fonctionnement d'un système qui évolue de façon autonome, c'est à dire dont les instants de franchissement ne sont pas connus, ou pas indiqués. [12]

Exemple :

Le cycle des saisons peut être modélisée par un réseau de pétri autonome.

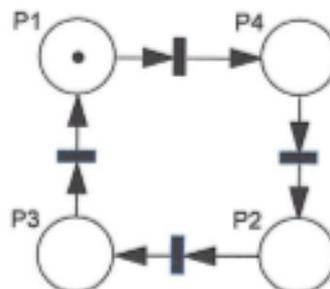


Figure 2.14 : réseau de pétri autonome.

II.7.1.1. Réseau de Pétri ordinaire

Un Réseau de Pétri ordinaire est un réseau dont les poids des arcs sont tous égaux à 1.

II.7.1.2. Réseau de Pétri généralisé

Un Réseau de Pétri généralisé est un réseau dans lequel des poids non nuls (nombres entiers strictement positifs) sont associés aux arcs.

Propriété :

Tout Réseau de Pétri généralisé peut être transformé en un Réseau de Pétri ordinaire.

II.7.1.3. Réseau de Pétri à capacité

Un Réseau de Pétri à capacité est un réseau dans lequel des capacités (nombres entiers strictement positifs) sont associés aux places (voir figure 2.15).

Le franchissement d'une transition d'entrée d'une place P , dont la capacité $\text{cap}(P)$ n'est possible que si le franchissement de cette transition ne conduit pas à un nombre de marques dans P_i supérieur à $\text{cap}(P_0)$.

Propriété :

Tout Réseau de Pétri à capacité peut être transformé en un Réseau de Pétri ordinaire.

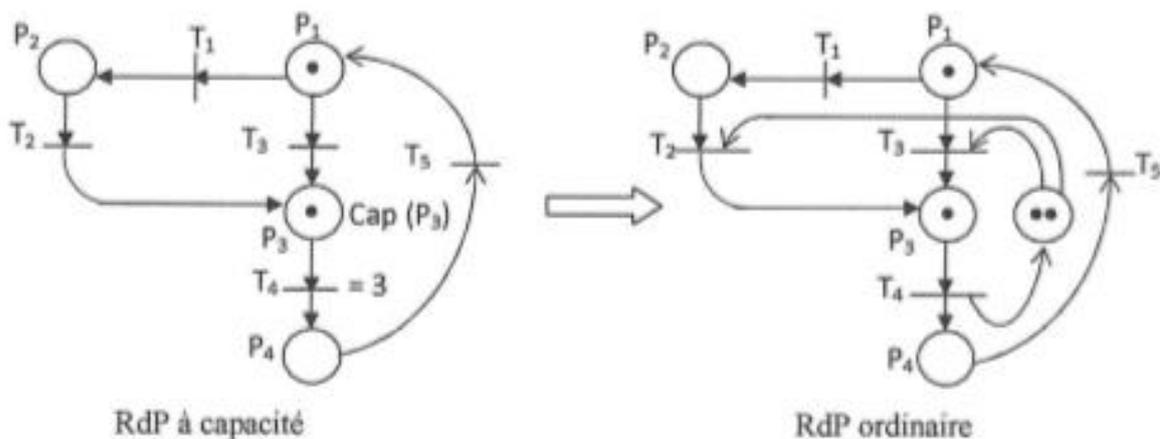


Figure 2.15 : Exemple de réseau de pétri à capacité.

II.7.1.4. Réseau de Pétri à prédicats

Un Réseau de Pétri à prédicats comporte des marques auxquelles sont associés des paramètres.

On utilise ce type de Réseaux lorsque les marques d'une même place ne sont pas du même type.

Pour mettre en évidence ce concept, considérons le cas du réseau de Pétri de la figure 2.16 qui permet de modéliser le lien de parenté entre père et enfant dans un ensemble de quatre personnes parmi lesquels : Mohamed est père de Ali et Hassen père de Kamel. [12]

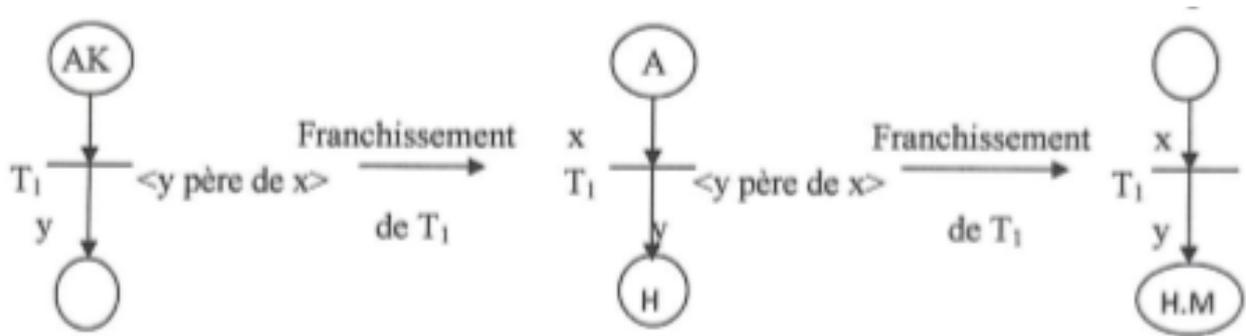


Figure 2. 16 : Exemple de Rdp à prédicats.

La structure du réseau de Pétri qui caractérise "des relations" est indépendante du nombre de paramètres qui caractérisent "un marquage".

Propriété :

Tout réseau de Pétri à prédicats, dont le nombre de prédicats est fini, peut être transformé en un réseau de Pétri ordinaire.

II.7.1.5. Réseau de Pétri coloré

Un Réseau de Pétri coloré comporte des marques auxquelles sont attribuées des couleurs.

Il existe plusieurs types de Réseaux de Pétri, tel que le Rdp à prédicats, qui peuvent être considéré comme colorés.

Propriété :

Tout réseau de Pétri coloré, dont le nombre de couleurs est fini, peut être transformé en un réseau de Pétri ordinaire.

II.7.1.6. Réseau de Pétri à arcs inhibiteur

Un arc inhibiteur est un arc orienté qui part d'une place P_i pour aboutir à une transition T ; et dont l'extrémité d'arrivée est marquée par un petit cercle. Il signifie que la transition T ; n'est validée que si la place P_i ne contient aucune marque (c'est un test à 0) (voir figure 2.17). [10]

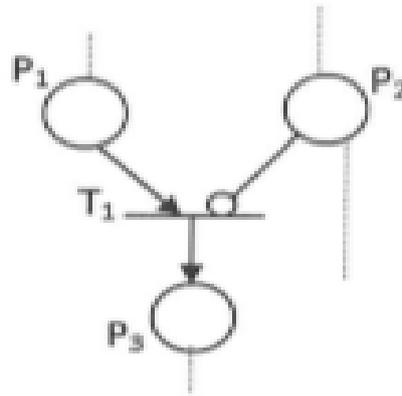


Figure 2.17 : Exemple de RdP à arcs inhibiteur.

Dans ce cas, le franchissement de la transition T , n'est possible que si P_2 ne contient aucun jeton et consiste à retirer une marque dans P_1 , et ajouter une marque dans P_3 .

Propriété :

Dans le cas général, un réseau de Pétri à arcs inhibiteurs ne peut pas être transformé en un réseau de Pétri ordinaire sauf s'il est borné. [12]

II.7.1.7 Réseau de Pétri à priorité

Un tel réseau est utilisé lorsque l'on veut imposer un choix entre plusieurs transitions validées. Il est composé d'un réseau de Pétri et d'une relation d'ordre partielle sur les transitions du réseau.

II.8. Réseaux de pétri non autonomes

Dans un réseau de Pétri autonome, on sait qu'une transition validée est franchissable, mais on ne sait pas quand elle sera franchie. C'est pourquoi, on pouvait qualifier une transition indifféremment par validée ou franchissable.

Dans le cas des réseaux de Pétri non autonomes, la situation est différente, il décrit le fonctionnement d'un système dont l'évolution est coordonnée par des événements extrêmes ou par le temps.

Il existe deux types de réseaux de Pétri non autonomes : Synchronisés et Temporisés.

II.8.1 Réseaux de Pétri synchronisés

➤ Définitions

Un réseau de Pétri synchronisé « RdPS » est un réseau de Pétri autonome auquel on a associé un ensemble d'événements externes à l'ensemble de ses transitions. Cet ensemble peut comporter l'événement certain.

Un événement est tout prédicat dont le résultat est vrai ou faux.

➤ Règles de marquage ou d'évolutions

Le franchissement d'une transition d'un réseau de Pétri s'effectue :

Si la transition est validée,

Et Quand l'événement associé se produira.

Les règles de déplacement des jetons sont identiques à celles d'un réseau de Pétri autonome.

II.8.2. Réseaux de Pétri Temporisés

Un réseau de pétri temporisé « RdPT » permet de décrire un système dont le fonctionnement dépend du temps et d'évaluer ces performances. Il permet de rendre compte de la durée nécessaire à l'exécution d'une opération.

Il y a principalement deux façons de modéliser les temporisations. Elles sont soit associées aux places et on dit que l'on a un réseau de Pétri P-Temporisé soit associées aux transitions et on dit que l'on a un réseau de Pétri T-Temporisé. [10]

II.8.2.1 Réseaux de Pétri P-Temporisés

Dans ce cas, une transition ne peut être tirée que si elle est sensibilisée et si les jetons ont séjourné dans les places précédentes pendant une durée au moins égale au temps de séjour minimum. En d'autres termes, lorsqu'une marque est déposée dans la place P_i , cette marque doit rester dans cette place pendant un temps d_j . On dit que la marque est indisponible pendant cette durée. Quand d_j est écoulée, la marque devient disponible. [12]

A l'instant initial, le marquage M_0 est constitué de marques disponibles.

II.8.2.2 Réseaux de Pétri T-Temporisés

Les réseaux de Pétri T-Temporisés sont équivalents aux réseaux de Pétri P-Temporisés.

Dans ce cas, une marque peut avoir deux états : elle peut être réservée pour le franchissement d'une transition T ; ou elle peut être non réservée.

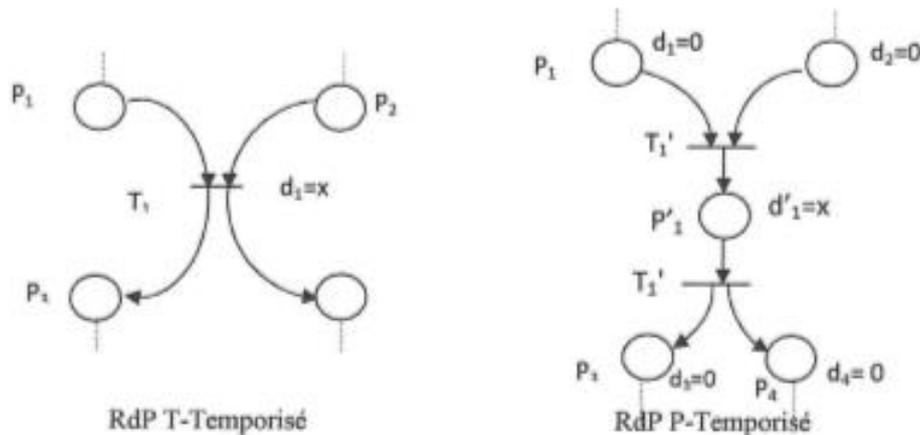


Figure 2.18 : Exemples de Réseaux de Pétri T-Temporisés

II.9. Les réseaux de Pétri stochastiques

Dans un RdP temporisé, des durées sont associées aux places ou aux transitions du réseau. Ces modèles sont bien adaptés pour étudier les systèmes pour lesquels les aspects quantitatifs sont bien définis. En revanche, certains phénomènes (tels que l'apparition de panne dans une machine ou la connaissance imprécise de certains paramètres tels qu'un débit) sont imprévisibles et ne peuvent alors plus être traités par ce type de modèle. Dans ce contexte, les réseaux de Pétri stochastiques ont été introduits par Florin et Natkin, afin de modéliser efficacement les phénomènes aléatoires. Ainsi, dans un RDP stochastique, les durées de sensibilisation associées aux transitions résultent d'un tirage aléatoire. L'hypothèse la plus couramment employée est que cette durée est distribuée selon des lois exponentielles. [10]

II.10. Réseaux de Pétri continu et modèle hybride

Vu les exigences et la complexité des processus industriels, plusieurs extensions de RdP ont été développées. Notons par exemple, la prise en compte de la contrainte du temps

pour l'étude des processus apparaît dans les réseaux de Pétri temporisés. De même, la notion du marquage réel est introduite à travers les RDP continus. Un réseau de pétri continu est un modèle dans lequel les nombres de marques dans les places sont des nombres réels. Ce modèle peut être introduit en considérant que le franchissement d'une transition se fait de façon continue avec une certaine vitesse. Il s'apparente aux réseaux de Pétri temporisés sur les transitions avec l'évolution technologique et l'orientation vers la description plus réaliste possible des processus, d'autres développements ont été effectués en particulier dans le cas du couplage d'un modèle RdP avec un comportement continu du système. La première notion de RdP hybride est apparue.

Les réseaux de Pétri continus étendent le marquage dans l'espace d'état des réels en fonction du taux de franchissement des transitions. Ainsi, le processus du franchissement des transitions obéi aux conditions suivantes :

- Une transition est considérée sensibilisée dès que le marquage de la place en amont est strictement différent de zéro.

Le franchissement d'une transition n'est plus instantané puisque le jeton va franchir la transition par une quantité infinitésimale. Ceci a nécessité une association des vitesses de franchissement aux transitions.

Avec ces principes, la structure est formée d'une transition continue avec une place continue d'entrée et une place continue de sortie. L'ensemble est l'image exacte d'un sablier, où le jeton s'écoule continument de la place d'entrée vers la place de sortie. De ce fait, l'évolution du marquage devient :

$$\frac{dM}{dt} = W * v(t)$$

Avec $v(t)$ la vitesse de franchissement des transitions à l'instant t .

Dans la littérature, on dénombre, trois configurations de réseaux de Pétri continus qui ont été formellement définis : RDPC à vitesse asymptotique RdPCC à vitesse constante et RdPCV à vitesse variable. Ils sont différenciés par la façon de calculer les vitesses de franchissement des transitions. [12]

Bien que, le modèle constitue une approximation des systèmes à événements discrets, les travaux de Julvez et montrent que le RdPCV à vitesse variable peut être utilisé pour la représentation des SDH. Les RdPCV à vitesse variable sont bien adaptés à l'étude des

systèmes dynamiques hybrides vu qu'ils combinent des aspects structurels et discrets avec des comportements continus (équations différentielles

du premier ordre qui déterminent l'évolution temporelle du marquage). Ainsi, chaque transition est franchie avec une fréquence de franchissement effective inférieure à la fréquence maximale. Soit $v = V_{j=1...p} \in \mathbb{R}^p$ le vecteur de vitesse de franchissements de la transition T ; à l'instant t . Les composantes de ce dernier dépendent continûment du marquage des places de la forme : [12]

$$v_j(t) = v_{maxj} * \mu_j(t)$$

$$\text{Avec } \mu_j(t) = \min_{p_i \in {}^{-}T_j}(M_i(t))$$

T ; L'ensemble des places d'entrée de la transition T ;

En fonction du paramètre minimum, l'évolution du marquage présente des commutations. De ce fait, une représentation par morceaux peut être utilisée. D'autre part, à partir d'un ensemble de transformations mathématiques, le modèle est décrit sous forme de représentation d'état.

Ainsi, le vecteur marquage correspond au vecteur d'état, le vecteur sorti est donné par des sommes pondérées des sous-ensembles de places et les taux de franchissement des transitions forment le vecteur d'entrée. Comme le montre la Figure II. 26, chaque transition est bouclée sur elle-même afin de réduire le nombre de franchissements simultanés. La limitation du nombre de franchissements des transitions apparait dans la représentation d'état du modèle comme une commutation d'une structure à une autre. Ce qui rend le modèle apte à décrire les phénomènes à transitions. [12]

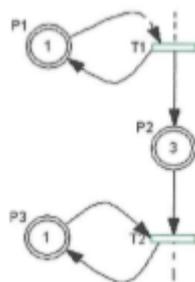


Figure 2.19 : Principe du réseau de Pétri continu à vitesse variable

Dans ce contexte, les arcs d'un réseau de Pétri continu portent des débits qui limitent le passage de jetons continus (Figure 2.19) et créent un marquage décrit par une équation différentielle définissant l'état du système.

Afin d'aboutir aux expressions exprimant l'évolution du marquage de réseaux, considérons la dérivée de la dynamique du marquage discret.

Donnée par :

$$\dot{M}(t) = W\sigma(t)$$

Où σ décrit le flux des transitions. [13]

Ainsi, le flux de transition est proportionnel à son degré de franchissement. Ce dernier dépend du marquage de la place d'entrée et du poids connecté à la transition considérée. Le taux de franchissement des transitions est calculé en considérant la division du marquage de chaque place d'entrée par le poids de l'arc. Le résultat donnant le minimum de la division correspondra à la transition franchit.

En conséquent, l'évolution du marquage est limitée par le comportement du flot de ces transitions de sortie. De ce fait, le processus sera décrit par une structure de sous réseau formé par un ensemble de transitions et places qui change en fonction du marquage minimal.

II.11 Conclusion :

Dans ce chapitre nous avons donné une vue sur le marquage permet de représenter l'état d'un système dynamique discret ou continu. Généralement, on associe les places aux différentes variables d'état du système, tandis que les transitions sont associées aux événements susceptibles de modifier l'état du système.

C'est le seul formalisme qui permet de modéliser, les contraintes liées au temps, celles liées aux ressources, ainsi que les contraintes de précedence entre les tâches. L'évolution du marquage permet de simuler la dynamique du système, le réseau de pétri temporisé permet d'étudier les problèmes d'ordonnancement répétitifs.

Chapitre III

Les méthodes de résolution des problèmes d'optimisation

III.1 Introduction

Dans ce chapitre nous présentons des techniques d'optimisations éprouvées, qui nous permet de réaliser rapidement des logiciels d'optimisation, notamment dans le domaine d'ordonnancement de production.

Nos logiciels permettent de créer, à partir d'une liste d'ordre de fabrication et de gammes, des plans de charges occupant chaque machine et chaque opérateur avec le moins possible de temps mort, et permettent de prendre en compte toutes les contraintes qui peuvent être énoncées logique ou temporelle (valeur minimum, pas deux tâches sur le même poste au même moment, respect de la séquence des opérations, etc.).

Ces logiciels sont souples et rapides et permettent de faire face aux aléas techniques et humains.

Ce qui suit est une représentation sur les méthodes de résolution des problèmes d'optimisation.

Parmi les méthodes d'optimisation on trouve les méthodes exactes, qui ont trouvé leurs limites face aux problèmes d'optimisation de taille importante, ce problème a été résolu à l'apparition des méthodes métaheuristiques. Une métaheuristique ne garantit pas une solution, mais elle offre une solution proche de l'optimum. Des méthodes hybrides ont vu le jour, qui se basent à l'hybridation de deux métaheuristiques afin d'aboutir à une meilleure optimisation possible. Dans notre problème d'optimisation on se base sur une méthode hybride sur deux métaheuristiques (l'algorithme génétique et la recherche tabou).

III.2 les méthodes exactes

III.2.1 introduction

Les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classées problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Etant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées. Parmi ces méthodes on trouve les méthodes exactes.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Les méthodes exactes ont permis de trouver des solutions optimales pour des problèmes de taille raisonnable. Parmi les méthodes exactes, on trouve la méthode Branch and Bound, la Recherche Exhaustive, Backtracking.

III.2.2 la method branch and bound

Par convenance, on représente l'exécution de la méthode de branch-and-bound à travers une arborescence. La racine de cette arborescence représente l'ensemble de toutes les solutions du problème considéré. Dans ce qui suit, nous résumons la méthode de branch-and-bound sur des problèmes de minimisation. Pour appliquer la méthode de branch-and-bound, nous devons être en possession : [3]

1. d'un moyen de calcul d'une borne inférieure d'une solution partielle
2. d'une stratégie de subdiviser l'espace de recherche pour créer des espaces de recherche de plus en plus petits.
3. d'un moyen de calcul d'une borne supérieure pour au moins une solution.

La méthode commence par considérer le problème de départ avec son ensemble de solutions, appelé la racine. Des procédures de bornes inférieures et supérieures sont appliquées à la racine. Si ces deux bornes sont égales, alors une solution optimale est trouvée, et on arrête là. Sinon, l'ensemble des solutions est divisé en deux ou plusieurs sous-problèmes, devenant ainsi des enfants de la racine. La méthode est ensuite appliquée récursivement à ces sous-problèmes, engendrant ainsi une arborescence. Si une solution optimale est trouvée pour un sous-problème, elle est réalisable, mais pas nécessairement optimale, pour le problème départ. Comme elle est réalisable, elle peut être utilisée pour éliminer toute sa descendance : si la borne inférieure d'un nœud dépasse la valeur d'une solution déjà connue, alors on peut affirmer que la solution optimale globale ne peut être contenue dans le sous-ensemble de solution représenté par ce nœud. La recherche continue jusqu'à ce que tous les nœuds sont soit explorés ou éliminés. [3]

III.2.3 recherche exhaustive

C'est une méthode exacte complète (la plus simple), elle consiste à énumérer toutes les solutions réalisables possibles, jusqu'à trouver une solution optimale du problème considéré.

-Utilisable pour traiter des problèmes d'optimisation combinatoire de petites tailles.

-Inefficace dans le cas où la taille de l'espace des solutions réalisables est exponentielle (ce qui est le cas pour tous les problèmes d'optimisation combinatoire).

III.2.4 backtracking

La procédure Backtracking français le retour sur trace consiste à revenir légèrement en arrière sur des décisions prises afin de sortir d'un blocage. La méthode des essais et erreurs (anglais : trial and error) constitue un exemple simple de Backtracking.

Backtracking est un outil important pour résoudre les problèmes d'optimisation combinatoire. C'est souvent le plus pratique technique pour l'analyse, pour le problème du sac à dos et d'autres problèmes d'optimisation combinatoire. Cette méthode est garantie pour trouver toutes les solutions à un problème fini dans un temps limité.

Malgré les progrès réalisés, comme le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème, les méthodes exactes rencontrent généralement des difficultés face aux applications de taille importante. Dans ce qui suit nous adaptons les méta heuristiques qui permettent de résoudre ce problème, et peuvent proposer une solution pour des problèmes d'optimisation de plus grande taille.

III.3 les méta-heuristiques

III.3.1 introduction

Depuis une dizaine d'années, des progrès importants ont été réalisés avec l'apparition d'une nouvelle génération de méthodes approchées puissantes et générales, souvent appelées métaheuristiques. On constate, depuis ces dernières années, que l'intérêt porté aux métaheuristiques augmente continuellement en recherche opérationnelle et en intelligence artificielle. Une métaheuristique est constituée d'un ensemble de concepts fondamentaux qui permettent d'aider à la résolution d'un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes.

Elles se distinguent classiquement en deux groupes : les méthodes de recherche locales (les approches **trajectoire**) et les méthodes de recherche globale (les approches **population**). A partir d'un point de départ quelconque, les méthodes locales permettent d'obtenir le minimum local le plus proche du point de départ et sont incapables de s'extraire d'un minimum local. Les méthodes globales possèdent un mécanisme leur permettant de s'extraire des minima locaux.

Ces métaheuristiques sont représentées essentiellement par les méthodes de voisinage, (Recherche Tabou, recuit simulé) et les algorithmes évolutifs (algorithmes génétiques, algorithmes de colonie de fourmis,). Grâce à ces méthodes, on peut proposer aujourd'hui des solutions approchées pour des problèmes d'optimisation de plus grande taille et pour de très nombreuses applications qu'il était impossible de traiter auparavant.

III.3.2-principe d'élaboration des métaheuristiques

III.3.2.1 introduction

Les métaheuristiques ont changé radicalement l'élaboration d'heuristiques : alors que l'on commençait par s'interroger sur les caractéristiques et les particularités du problème à résoudre avant de commencer à programmer une méthode spécifique, les métaheuristiques ont en quelque sorte inversé le processus, la trame de la méthode de résolution étant fournie par la métaphore qui a inspiré la métaheuristique. Ayant une première heuristique, on cherche ensuite à l'améliorer en observant les faiblesses qu'elle présente pour le problème à résoudre.

III.3.2.2 mémoires

On peut dire que toutes les métaheuristiques font appel à une structure de voisinage ou à une mémoire, et généralement aux deux à la fois. En effet, pour simplifier et en ne considérant que les versions les plus primitives de ces métaheuristiques, il n'y a que les algorithmes génétiques et les colonies de fourmis qui n'utilisent pas de structure de voisinage. Dans leurs versions plus élaborées, ces techniques se dotent cependant d'une procédure de recherche locale, et donc d'une structure de voisinage. De manière analogue, seuls le recuit simulé, les algorithmes de bruitage et d'acceptation à seuils ne reposent que sur un générateur pseudo aléatoire et ne font pas appel à une forme de mémoire pour s'échapper des optima locaux relativement à la structure de voisinage choisie.

III.3.2.3 Construction de solutions

La reconstruction de nouvelles solutions à l'aide d'une mémoire fait partie de nombreuses métaheuristiques. Les algorithmes génétiques ou les colonies de fourmis artificielles sont basés essentiellement sur le principe de la construction périodique de solutions entièrement nouvelles, contrairement au recuit simulé ou des algorithmes de bruitage qui s'appuient seulement sur une structure de voisinage pour modifier légèrement une solution existante. Le cas de la recherche avec tabous est intermédiaire. Dans sa forme la plus élémentaire, elle aussi n'a qu'une structure de voisinage, modulée en fonction des solutions précédemment énumérées. Cependant, certains principes plus évolués de la recherche avec tabous font également appel à la construction de nouvelles solutions.

III.3.2.4 voisinages

Le principe général le plus largement utilisé dans l'élaboration d'heuristiques est celui de voisinage. À chaque solution s du problème, on associe un sous-ensemble $V(s)$ de solutions. Ce sous-ensemble peut être statique, comme dans le cas du recuit simulé ou des méthodes de bruitage, mais aussi dynamique et dépendre du temps t . [4]

Même dans le cas d'une structure de voisinage statique et très simple, où le seul but est de trouver un optimum local, on ne connaît que très peu de résultats théoriques utiles. [4]

Pour un problème d'optimisation combinatoire, pour lequel il est possible de définir un ensemble de voisinages a priori intéressants, la situation se complexifie. Il devient alors difficile de se décider pour l'un ou l'autre des voisinages autrement que par des essais. Comme le voisinage n'est qu'une partie des principes, tout interdépendants, utilisés dans l'heuristique, le choix d'un (ou de plusieurs) voisinages devient réellement problématique car on ne dispose que de très peu de résultats théoriques sur la qualité d'un voisinage pour un problème particulier. [4]

III.3.2.5 stratégies

Un principe très largement utilisé dans les métaheuristiques (et pas seulement pour la recherche avec Tabou) est connu sous le nom d'oscillations stratégiques.

Son principe nécessite qu'il faut modifier périodiquement les paramètres d'une recherche itérative pour être mieux à même de l'intensifier, c'est-à-dire d'examiner en détail un sous ensemble très restreint des solutions, et de la diversifier en changeant fondamentalement la structure des solutions.[4]

Dans une recherche évoluée, ces phases seront alternées. Une autre possibilité consiste à retourner en arrière et à redémarrer la recherche à partir d'une bonne solution trouvée précédemment, ou à partir d'une de ses voisines. Une façon très simple d'implanter une diversification dans une recherche avec Tabou et d'effectuer un certain nombre de mouvements aléatoires.[4]

III.4 les différentes métaheuristiques

III.4.1 la méthode du recuit simulé

Cette méthode de recherche consiste à reproduire le phénomène de recuit obtenu lors du refroidissement d'un matériau métallique préalablement chauffé. La structure du matériau (taille et arrangement des cristaux), varie en fonction de la vitesse de refroidissement. Le recuit simulé reproduit les variations d'énergie observées. De même que la recherche Tabou, le recuit simulé est doté d'un mécanisme qui lui permet de s'échapper des optima locaux.[5]

L'algorithme du recuit simulé, représenté de manière simplifiée sur la figure ci-dessous, repose sur la démarche suivante : partant d'un état initial aléatoire, l'algorithme va de façon itérative sélectionner puis évaluer un état voisin (variation d'énergie). Le nouvel état est accepté si la variation d'énergie est positive ou, si le nouvel état est plus mauvais, selon une probabilité où E correspond à l'énergie d'un état et T le paramètre température (critère de l'algorithme de Metropolis). Un mauvais déplacement a donc plus de chance d'être accepté si T est élevé. La température initiale est élevée pour permettre au recuit d'explorer l'espace de recherche et sortir des optima locaux. Plus la température baisse, plus l'algorithme ne va se concentrer sur une zone particulière. Elle est généralement abaissée lentement par seuil jusqu'à un critère de fin selon la performance des solutions proposées ou selon le nombre d'itérations. [5]

Qui peut être décrit comme suit.

Choisir une solution $s \in S$ ainsi qu'une température initiale T ;

Tant que aucun critère d'arrêt n'est satisfait faire

Choisir aléatoirement une solution $s_0 \in N(s)$;

Générer un nombre réel aléatoire $r \in [0, 1]$;

Si $r < p(T, s, s_0)$ alors poser $s \leftarrow s_0$;

Mettre à jour T ;

Fin du tant que

En général, on définit $p(T, s, s_0) = e^{-\frac{f(s) - f(s_0)}{T}}$ (qui est la distribution de Boltzmann).
Ainsi :

- si $f(s_0) \leq f(s)$ alors $p(T, s, s_0) \geq 1 - r$, ce qui signifie qu'on accepte s_0
- si T a une très grande valeur, alors $p(T, s, s_0) \approx 1$ et s_0 est donc très probablement acceptée
- si $T \approx 0$ et $f(s) < f(s_0)$, alors $p(T, s, s_0) \approx 0$ et s_0 est donc très probablement refusée. [5]

La méthode du Recuit Simulé est une méthode sans mémoire. On peut facilement l'améliorer en rajoutant une mémoire à long terme qui stocke la meilleure solution rencontrée. En effet, l'algorithme du Recuit Simulé décrit ci-dessus peut converger vers une solution s^* en ayant visité auparavant une solution s de valeur $f(s) < f(s^*)$. Il est donc naturel de mémoriser la meilleure solution rencontrée durant le processus de recherche. Le critère d'arrêt peut être une limite sur le temps ou sur le nombre d'itérations sans modification de la solution courante. Si on stocke la meilleure solution rencontrée s^* , on peut décider de stopper la recherche dès qu'un certain nombre d'itérations a été effectué sans amélioration. [5]

L'un des principaux défauts de la méthode du Recuit Simulé est le choix aléatoire du voisin s_0 dans $N(s)$. On peut donc être proche de l'optimum et passer juste à côté sans le voir.

III.4.2 algorithmes gloutons

Les algorithmes gloutons (greedy algorithms en anglais) sont des algorithmes pour lesquels, à chaque itération, on fixe la valeur d'une (ou plusieurs) des variables décrivant le problème sans remettre en cause les choix antérieurs.[5]

Le principe est donc de partir d'une solution incomplète (éventuellement totalement indéterminée) que l'on complète de proche en proche en effectuant des choix définitifs : à chaque étape, on traite une partie des variables sur lesquelles on ne revient plus.

Stratégie :

- Tester tous les candidats $C[i]$ exactement une fois.
- Si $C[i]$ peut être ajouté à S , garder le.
- Si $C[i]$ ne peut pas être ajouté à S , jeter le.

➤ Ne retester jamais un candidat déjà testé.

Ces algorithmes se résument à la forme suivante

- $S \leftarrow \emptyset$
- $i \leftarrow 1$
- **Tant que** la solution n'est pas complète **Et** \exists des candidats à testés **Faire**
- **Si** $C[i]$ Réalisable **Alors**
- $S \leftarrow S \cup \{C[i]\}$
- $i \leftarrow i+1$
- **Retourner** S
- **Fin Tant que**

Si les algorithmes de ce type sont souvent rapides, en revanche la solution qu'ils déterminent peut-être arbitrairement loin de la solution. On les utilise néanmoins fréquemment pour obtenir rapidement une solution réalisable. Par exemple, elles servent à initialiser une méthode itérative. Mais dans certains types d'instances réelles, la solution gloutonne est parfois très bonne. [5]

III.4.3 la méthode recherche tabou

Le principe de la Recherche Tabou est de choisir à chaque itération la meilleure solution $s_0 \in N(s)$, même si $f(s_0) > f(s)$. Lorsqu'on atteint un minimum local s par rapport au voisinage N , la Recherche Tabou va donc se déplacer vers une solution s_0 de valeur $f(s_0) \geq f(s)$. Le danger est alors de revenir à s puisque, en général, $s \in N(s_0)$ et $f(s) \leq f(s_0)$. Pour éviter de tourner ainsi en rond, on crée une liste T qui mémorise les dernières solutions visitées et interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste Tabou. Les solutions ne demeurent dans T que pour un nombre limité d'itérations. La liste T est donc une mémoire à court terme. Si une solution s_0 est dans T on dit que s_0 est une solution taboue. De même, tout mouvement qui nous mène de la solution courante à une solution de T est appelé mouvement tabou. La mémorisation de solutions dans T demande typiquement beaucoup de place mémoire, et il peut être difficile de vérifier si une solution donnée est dans T . C'est la principale raison pour laquelle certains préfèrent mémoriser des attributs ou des modifications de solutions. Ainsi, on peut interdire de visiter une solution présentant un attribut appartenant à T . Ou alors, on peut interdire une solution s_0 si le mouvement qui permet de se rendre de la solution courante s vers la solution voisine s_0 est un mouvement appartenant à la liste T . [5]

L'algorithme de Recherche Tabou peut être décrit comme suit

Choisir une solution $s \in S$,

Poser $T \leftarrow \emptyset$ et $s^* \leftarrow s$;

Tant qu'aucun critère d'arrêt n'est satisfait faire

Déterminer une solution s_0 qui minimise f dans $NT(S)$;

Si $f(s_0) < f(s^*)$ alors

Poser $s^* \leftarrow s_0$; Poser $s \leftarrow s_0$ et mettre à jour T ;

Fin du tant que

- La liste Tabou

Comme critère d'arrêt on peut par exemple fixer un nombre maximum d'itérations sans amélioration de s^* , ou on peut fixer un temps limite après lequel la recherche doit s'arrêter. Certains préconisent l'utilisation d'une liste Tabou de longueur variable d'autres préfèrent les listes Tabou réactives dont la taille varie selon les résultats de la recherche. Ainsi, si des solutions sont visitées de manière répétée (à intervalle $> |T|$) alors on peut augmenter la longueur de la liste, ce qui aura pour effet de diversifier la recherche. Aussi, si la solution courante n'est que rarement améliorée, on peut décider d'intensifier la recherche autour de la meilleure solution rencontrée s^* , ce qui peut se faire en diminuant la longueur de la liste Tabou. [5]

La liste Tabou est une mémoire à court terme. On peut rajouter une mémoire à plus long terme. Quatre types de mémoire sont alors envisageables :

- Les mémoires basées sur la récence mémorisent pour chaque solution (attribut ou mouvement) l'itération la plus récente qui l'impliquait. On peut par exemple favoriser les mouvements vers des solutions contenant des attributs peu récents, afin d'éviter de rester bloquer dans une même région de l'espace de recherche.
- Les mémoires basées sur la fréquence mémorisent le nombre de fois que les solutions (attributs ou mouvements) ont été visitées. Cette information permet d'identifier les régions

de S qui ont été le plus visitées. On peut exploiter cette information pour diversifier la recherche.

- Les mémoires basées sur la qualité mémorisent les meilleures solutions rencontrées afin d'identifier les composantes communes de ces solutions. On peut ensuite faire usage de cette information pour créer de nouvelles solutions qui contiennent autant que possible ces apparemment bonnes composantes.
- Les mémoires basées sur l'influence mémorisent les choix qui se sont avérés les plus judicieux ou les plus critiques durant le processus de recherche. Ceci permet d'essayer de répéter les bons choix et d'éviter de répéter les mêmes erreurs. [5]

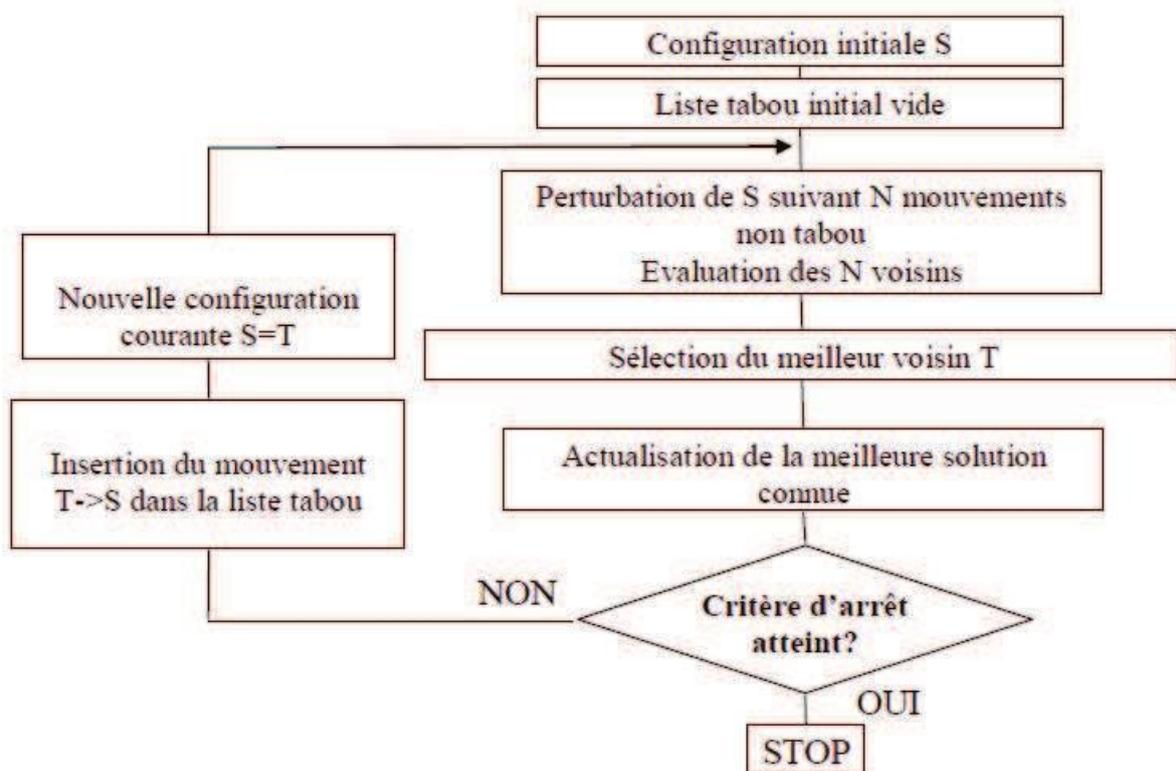


Figure 3.1 Organigramme général de l'algorithme recherche tabou

III.4.4 la méthode des algorithmes génétiques

Les algorithmes génétiques (AG) sont des algorithmes de recherche inspirés des mécanismes de l'évolution naturelle des êtres vivants et de la génétique. Les premiers travaux ont été menés par Holland en 1975 dans l'ouvrage "Adaptation of Natural and Artificiel System" qui

formalise les algorithmes génétiques dans le cadre de l'optimisation mathématique. La parution de l'ouvrage de David Goldberg en 1989 décrivant l'utilisation de ces algorithmes pour la résolution des problèmes concrets, a permis de mieux faire connaître ces derniers dans la communauté scientifique et a marqué le début d'un nouvel intérêt pour cette technique d'optimisation. Ces méthodes s'attachent à simuler le processus de sélection naturelle dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par Darwin en 1859 " Journal of Research in to the Geology and Natural History". Selon ces concepts, lorsqu'une population est soumise aux contraintes d'un milieu naturel, seuls les individus les mieux adaptés survivent et génèrent une descendance. Au cours des générations, la sélection naturelle permet l'apparition d'individus de mieux en mieux adaptés au milieu naturel, au détriment de ceux se montrant notoirement inadaptés, assurant ainsi la pérennité de l'espèce. Cette particularité de l'évolution naturelle : la capacité d'une population à explorer son environnement en parallèle et à recombinaison les meilleurs individus entre eux, est empruntée par les algorithmes génétiques. [6]

Principes

Par analogie à la génétique, ces algorithmes cherchent à partir d'une population initiale les meilleurs individus. Pour constituer les parents appropriés donnant naissance à des meilleures descendance Enfants, parmi les quelle seront tiré les solutions acceptables du problème traité. Chaque problème d'optimisation est caractérisé par des variables de décision qui conditionnent les décisions à prendre, un objectif à satisfaire et des contraintes à respecter. Contrairement aux méthodes d'optimisation classiques, les algorithmes génétiques n'utilisent pas les variables mais leur associe un codage particulier, les variables de décision sont prises en compte à l'aide de caractères représentant une séquence de codes. Chaque variable de décisions est traduite sous forme d'un gène qui peut contenir un ou plusieurs codes, pouvant exprimer des caractères différents. La séquence de code représente un individu. En d'autres termes une solution potentielle connue sous le nom de chromosome. [6]

Afin d'explicitier le fonctionnement des algorithmes génétiques, nous présentons dans la suite de cette section les différentes étapes d'un algorithme génétique simple illustrées sur la Figure 3.2

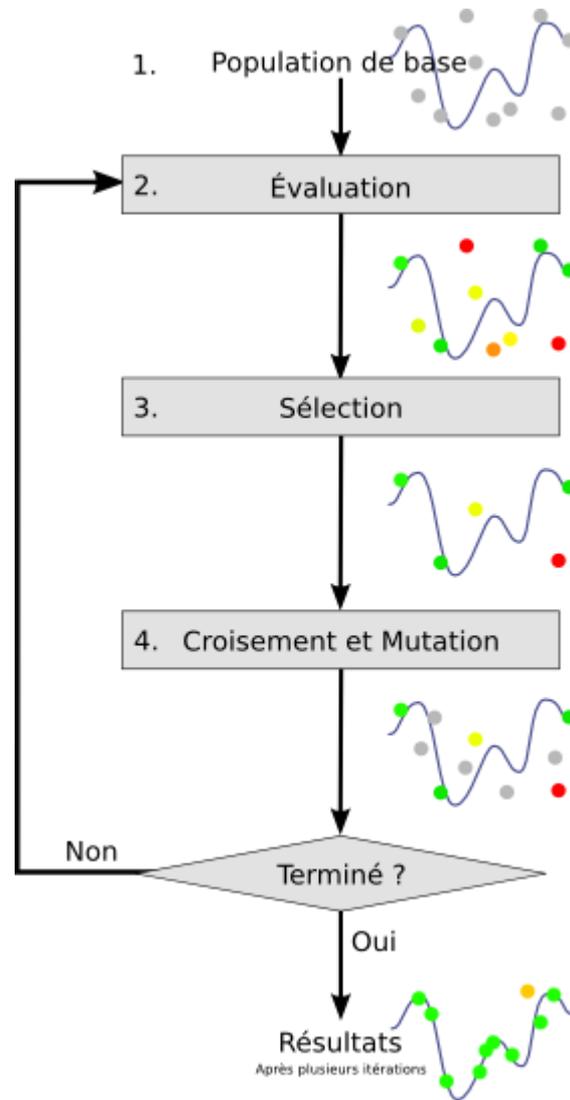


Figure3.2 Les principales étapes d'un algorithme génétique

L'algorithme génétique ne fait que transposer ce que fait la nature, à des systèmes artificiels. Il simule les processus évolutifs darwiniens et génétiques s'appliquant aux chromosomes. Il transforme un ensemble d'objets mathématiques. Une population d'individus souvent représentés par des chaînes de caractères pour imiter les chaînes d'ADN, chacun ayant une valeur d'adaptation, en une nouvelle population. [6]

L'algorithme fait donc appel à quatre opérateurs de base :

- **L'évaluation** : du niveau d'adaptation d'un individu.
- **La sélection** : processus où les individus sont copiés selon la valeur de leur fonction objective. On peut décrire la fonction f comme une mesure de profit, utilité ou qualité que l'on veut maximiser (minimiser). Si on copie des individus selon leur valeur f ceci implique que les individus ayant des valeurs plus élevées ont une plus grande probabilité de contribuer des rejetons à la prochaine génération. Ceci correspond à

une version artificielle de la « survie des plus forts » de Darwin. L'implantation de la sélection peut se faire de plusieurs façons. La plus facile est peut-être la roue de roulette biaisée, où la probabilité de reproduction d'un individu dépend de sa valeur par rapport au total des valeurs de la population.[7]

- **Le croisement** : Si on commence avec une population finie, le croisement entre paramètres ne permet d'atteindre qu'un nombre fini de points dans l'espace de recherche, soit ceux où les paramètres sont choisis parmi ceux de la population. En fait, si la population initiale est de taille n , on peut atteindre au plus $n \cdot m$ vecteurs de paramètres avec le croisement où m est le nombre de variables. Donc un des buts de la mutation est de permettre d'atteindre un point arbitraire dans l'espace.[7]
- **La mutation** : processus aléatoire où un bit change de valeur. Ce processus joue un rôle secondaire dans l'algorithme génétique, mais il est quand même important. La mutation assure qu'aucun point dans l'espace de recherche n'a une probabilité nulle d'être atteint. [7]

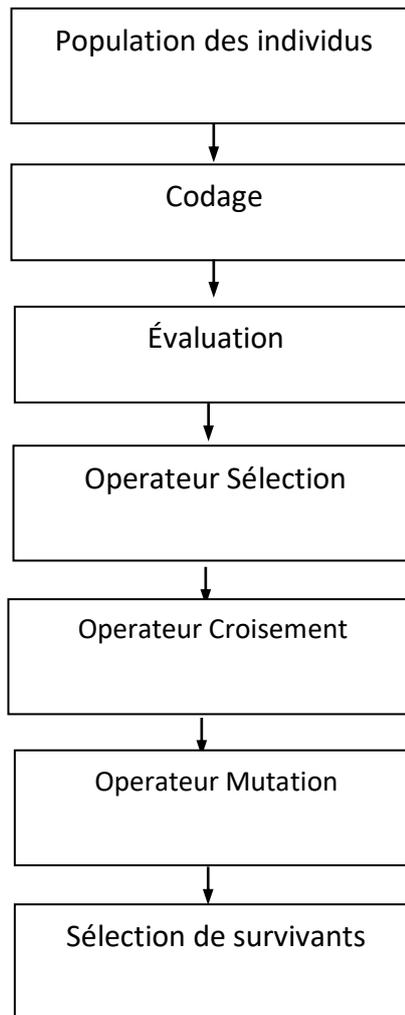


Figure3.3 Organigramme de l'algorithme génétique

III.5 les méthodes hybrides

III.5.1 introduction

L'hybridation est une tendance observée dans de nombreux travaux réalisés sur les métaheuristiques ces dix dernières années. Elle permet de tirer profit des avantages cumulés des différentes métaheuristiques, à tel point que les métaheuristiques que nous avons vues jusqu'à présent ne sont plus que des canevas, des points de départ, pour commencer à résoudre un problème d'optimisation.

Une des techniques les plus populaires d'hybridation concerne l'utilisation de métaheuristique de type trajectoire locale avec des métaheuristiques à population.

Une autre manière d'hybrider consiste à exécuter en parallèle plusieurs fois la même métaheuristique, mais avec des paramètres différents. Ces processus parallèles communiquent entre eux régulièrement pour échanger de l'information sur leurs résultats partiels. Cette

technique est d'autant plus utilisée que s'est approfondie la recherche sur les architectures parallèles. Toutes les métaheuristiques classiques ont fait l'objet d'une implémentation parallèle, y compris l'algorithme du recuit simulé qui, bien que de nature séquentielle, a pu être parallélisé en le divisant en processus élémentaires.

III.5.2 la méthode grasp

III.5.2.1 définition

GRASP (Greedy Randomized Adaptive Search Procédure) est un algorithme métaheuristique et multi-start utilisé pour la résolution des problèmes d'optimisation combinatoire, chaque itération de cet algorithme se passe sur deux étapes, la première – étape de construction – essaie de construire une solution selon l'algorithme semi-glouton (semi-Greedy) et la deuxième – étape de recherche locale – essaie d'améliorer la solution construite dans la première étape, par une recherche locale. La solution finale sera la meilleure solution obtenue. [9]

III.5.2.2 fonctionnements

GRASP est un algorithme métaheuristique basant dans son fonctionnement sur deux techniques d'optimisation largement connues, il s'agit de l'algorithme glouton (Greedy) et la recherche locale. Pour un nombre donné d'itérations, on effectue la construction d'une solution suivant une heuristique semi-glouton (Randomize Greedy), puis on applique une recherche locale sur le voisinage de cette solution afin d'obtenir un optimum local, la solution finale sera la meilleure solution obtenue parmi toutes les solutions de chaque itération. On remarque l'indépendance des itérations dans GRASP. [9]

- L'algorithme GRASP

Peut-être décrit comme suit.

Poser $f^* \leftarrow \infty$ (valeur de la meilleure solution rencontrée)

Tant qu'aucun critère d'arrêt n'est satisfait faire

Poser $s \leftarrow \emptyset$;

Tant que la solution courante s est partielle (et donc non complète) faire

Déterminer la liste RCL des R meilleures composantes pouvant être rajoutées à s ;

Choisir aléatoirement une composante dans RCL et la rajouter à s ;

Fin du tant que

Appliquer une Recherche Locale, en partant de s , pour obtenir une solution s_0 ; Si $f(s_0) < f^*$ alors poser $s^* \leftarrow s_0$ et $f^* \leftarrow f(s_0)$;

Fin du tant que

III.5.2.3 les avantages et les inconvénients

GRASP est l'une des métaheuristiques qui possède des avantages prometteurs pour les problèmes d'optimisation combinatoires ; parmi lesquels on note :

- Temps de calcul : par rapport à d'autres métaheuristiques, l'algorithme consomme moins de temps CPU, et il est prédictible.

Néanmoins, comme toute métaheuristique, GRASP est loin d'être une méthode parfaite :

- 1 Incertitude sur la qualité de la solution : la solution finale dépend toujours des les paramètres de l'algorithme.
- 2. Construction sans mémoire : Informations non sauvegardées d'une itération à l'autre

III.6 Méthode hybride séquentielle proposée

Notre méthode hybride séquentielle proposée se base sur deux étapes :

la première étape est la construction d'une population de solution selon l'algorithme génétique (population enfants), dont on applique les opérateurs de sélection et de croisement. Dans l'opérateur mutation il sera remplacé par une deuxième méthode métaheuristique (on a choisi la méthode recherche tabou) comme deuxième étape de recherche sur cette nouvelle population. Cette méthode essaie de trouver la meilleure solution parmi une population enfants. La solution finale obtenue sera la meilleure.

III.7 Conclusion

L'hybridation des métaheuristiques est devenue un enjeu important, à tel point que toutes les différentes métaheuristiques que nous avons présentés, recuit simulé, la Recherche Tabou, algorithmes génétiques, sont maintenant perçues comme des points de départ au développement de nouveaux algorithmes d'optimisation. Les différentes caractéristiques de ces métaheuristiques (mémoires, listes de candidats, populations, la fonction objectif, etc.) sont recomposées entre elles pour former de nouvelles méthodes, dont la liste s'allonge.

Chapitre IV

Résultats et conclusions

IV.1. Introduction :

Un F.M.S possède un environnement dynamique et compliqué qui est à l'origine de plusieurs problèmes. Notamment l'allocation, à chaque opération, des ressources requises pour la réalisation du produit.

Lorsque l'on aborde le problème d'ordonnement de fabrication, on doit considérer le caractère dynamique de son fonctionnement. En effet, les caractéristiques du problème sont dépendantes du contexte courant. Donc il est primordial de tenir compte de tous les paramètres qui interviennent au cours du temps, afin d'assurer la flexibilité et l'adaptabilité des traitements. On peut facilement identifier les quatre principaux éléments intervenant dans les problèmes d'ordonnement : les tâches, les ressources les travaux et les contraintes.

Le problème est alors d'identifier les plus facilement quantifiables et les plus pertinentes pour le problème à résoudre. Après cette identification, il s'agit de déterminer, en fonction du modèle de représentation construit, une méthode d'ordonnement adaptée au type de problème à résoudre.

Dans cette optique, nous avons proposé comme méthodes d'ordonnement :

La méthode hybride séquentielle pour la résolution des conflits d'accès aux ressources partagées. Le modèle de conduite, étant basée sur les réseaux de pétri temporisés, la métaheuristique développée est basée sur l'équation d'évolution et la notion du marquage à un instant donnée. L'évaluation du coût des séquences en conflits est exprimée en unités de temps.

L'objectif de l'algorithme de conduite est la résolution des conflits d'accès aux ressources partagées, sous la contrainte de minimisation du MAKESPAN (temps de réalisation global du processus composé de gammes différentes).

IV.2. Simulation du modèle de conduite

IV.2.1. Introduction

La simulation permet d'analyser le comportement du système spécifié et de réaliser une première vérification de la cohérence du modèle. Ainsi l'on prouvera que le comportement spécifié correspond à celui attendu. De point de vu des réseaux de pétri, un modèle peut être simulé selon deux types d'abstraction :

- Un système de production est généralement modélisé par un R.d.P temporisé.
- Le réseau de pétri est remplacé par l'équation d'évolution du marquage.

IV.2.2. Simulation par l'équation du marquage :

Un modèle basé sur les réseaux de pétri a l'avantage d'être exécutable. L'évolution du R.d.P depuis son marquage initial jusqu'à son marquage final simule l'évolution du système modélisé depuis son état initial jusqu'à son état final. Cette évolution est régie par l'équation fondamentale des R.d.P :

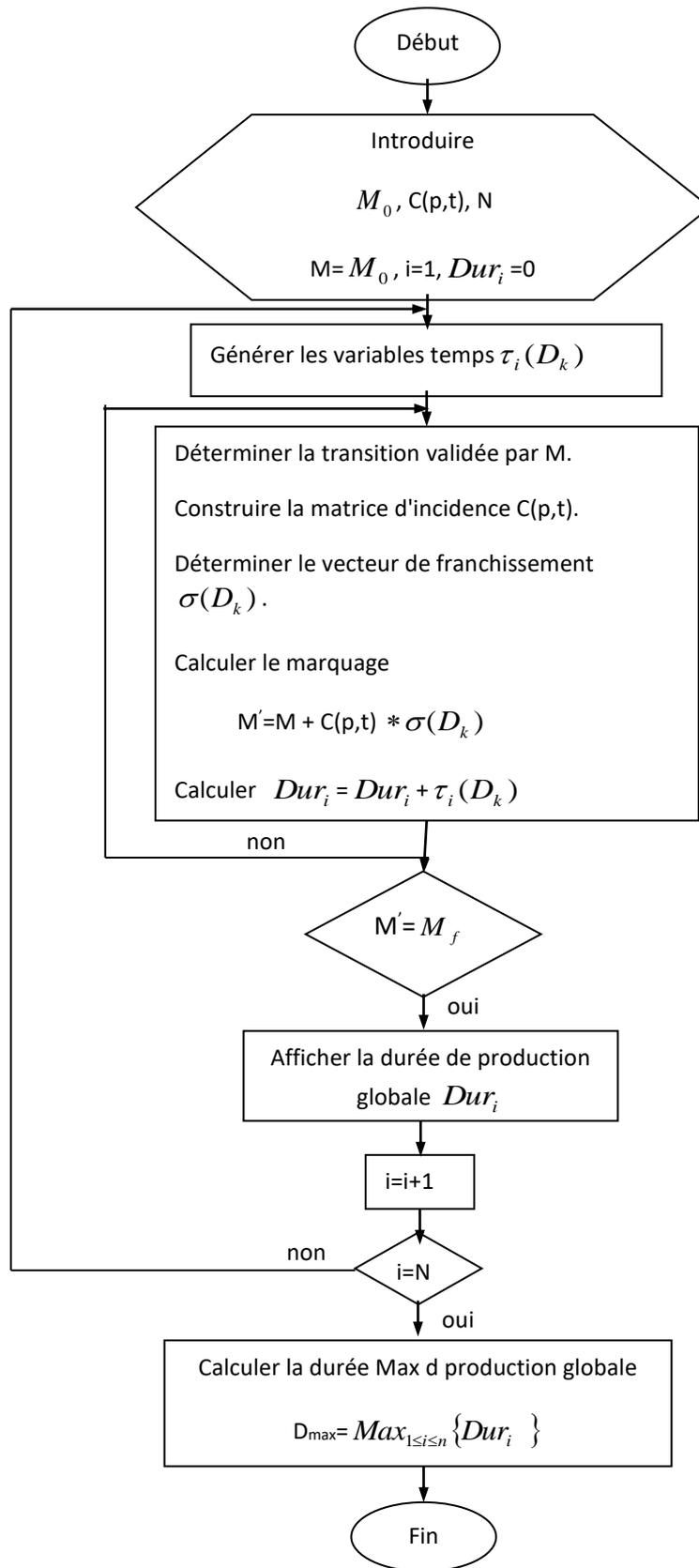
$$M_{k+1} = M_k + C(p,t) * D_k$$

Avec $C(p,t)$, la matrice d'incidence globale du réseau et D_k est le vecteur de franchissement

donné par :

$$D_k(j) = \begin{cases} 1 & \text{si } t_j \text{ est franchie après validation par le marquage } M_k \\ 0 & \text{si } t_j \text{ n'est pas franchis} \end{cases}$$

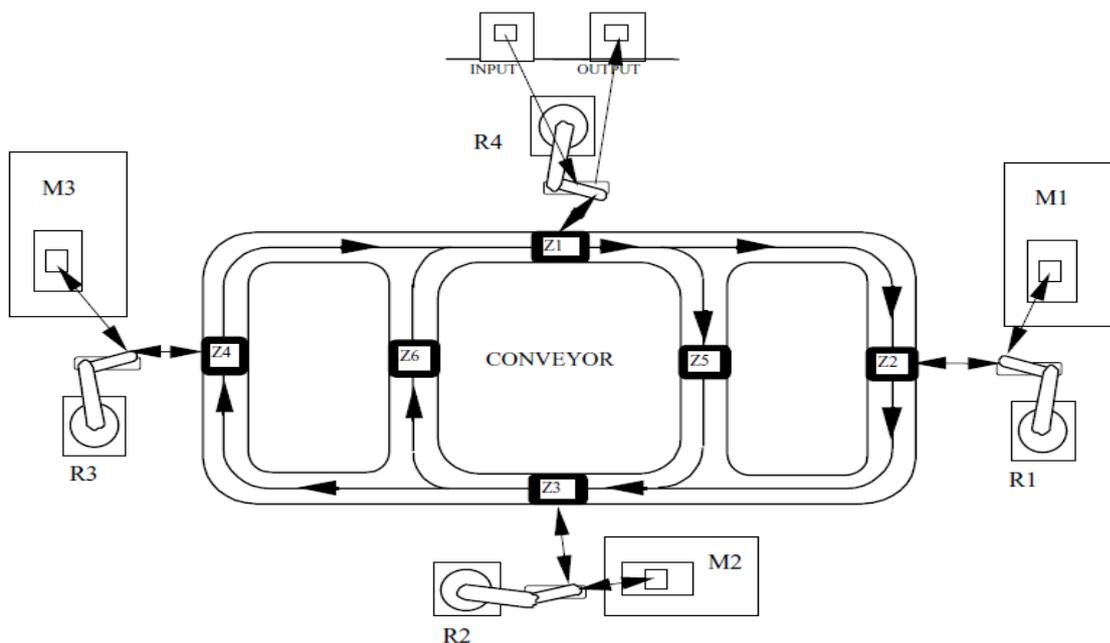
IV.3 Algorithme de simulation :



IV.3.1. Composition de la cellule de production flexible :

Dans les systèmes industriels, les opérations appartiennent à des catégories différentes : opérations de transferts, d'usinage, et de stockage. Elles sont définies au niveau des ressources. Plusieurs opérations peuvent rendre effectif les mêmes fonctions dans une FMS (cas de ressources redondantes). Une ressource peut s'exécuter aussi sur plusieurs opérations (cas de ressources polyvalentes). Soit la cellule de production flexible pilote, développée au LAGIS (LILLE). Les ressources principales de ce processus sont trois machines appelées M_1 , M_2 , M_3 . Elles exécutent respectivement les fonctions d'usinage : M_1 , M_2 , et M_3 . R_{14} est un robot qui exécute des opérations du transfert de FIFO IN à Z_1 , de Z_1 à FIFO OUT, de Z_1 à Z_2 , de Z_2 à Z_1 , de Z_1 à Z_4 , et de Z_4 à Z_1 . R_{11} est un robot de chargement /déchargement affecté à M_1 , R_{12} est un robot du chargement /déchargement affecté à M_2 , R_{13} est un robot de chargement /déchargement affecté à M_3 . CV est un convoyeur, permettant le transfert de Z_1 à Z_2 ou Z_5 , de Z_2 ou Z_5 à Z_3 , de Z_3 à Z_4 ou Z_6 , de Z_4 ou Z_6 à Z_1 .

Le convoyeur est une ressource de transport complexe. Des chemins différents peuvent être suivis pour atteindre le même emplacement. R_{14} peut exécuter le même transfert comme le convoyeur, de Z_1 à Z_2 . C'est un exemple de flexibilité du transport.



IV.3.2. Plan des charges :

On a utilisé 6 gammes opératoires, ci-dessous un exemple d'une gamme opératoire utilisé

- **Gamme d'opération :**

A: IN⇒Z1⇒Z2⇒M1⇒Z2⇒Z3⇒Z4⇒M2⇒Z4⇒Z1⇒OUT

⇒ : La pièce est transportée par le robot R14.

⇒: La pièce est transportée par le robot R11.

⇒: La pièce est transportée par le robot R13.

⇒ : La pièce est transportée par le convoyeur.

IV.4 Conduite du système de production flexible

IV.4.1 Problématique :

Nous voulons gérer les ressources partagées dans l'atelier de façon à minimiser le temps de fabrication global (MAKESPAN). Notre approche consiste à gérer les conflits d'accès aux différents moyens de production, en se basant sur le modèle établi à base des R.d.P et en lui associant la méthode hybride séquentielle dans l'ensemble des séquences en conflit. La meilleure solution est évaluée par rapport au critère retenu.

Le type de système étudié est un système de production flexible composé d'un ensemble de machines de capacité unitaire. (Exécute une opération à la fois).

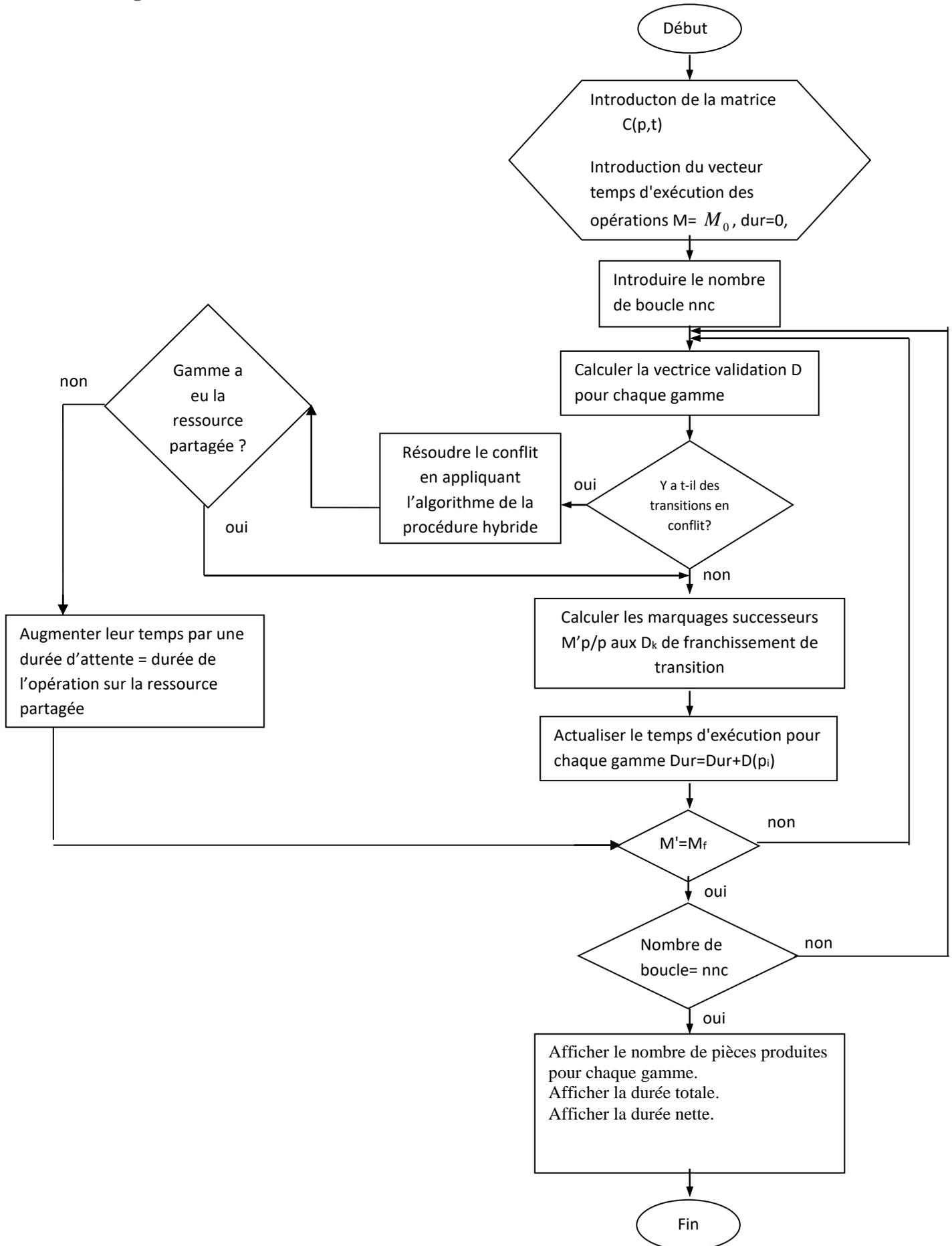
Chaque produit est défini par un ensemble d'opérations, qui peuvent être exécutées par différentes machines. Le temps requis par opération peut être différent d'une machine à une autre. Les stocks d'entrées et de sortie ont des capacités infinies. Il existe des contraintes de précédence dans un même produit mais pas entre les produits (les produits peuvent être exécutés en parallèle).

Notre problème est de minimiser le temps de réalisation global (MAKESPAN) dont la fonction d'évaluation est donnée par :

$$f_i = \sum \text{dur}_i + \sum \text{temps d'attente}.$$

Temps de réalisation = $\text{Max}_{1 \leq i \leq N} \{f_i\}$, N : nombre de gamme de production.

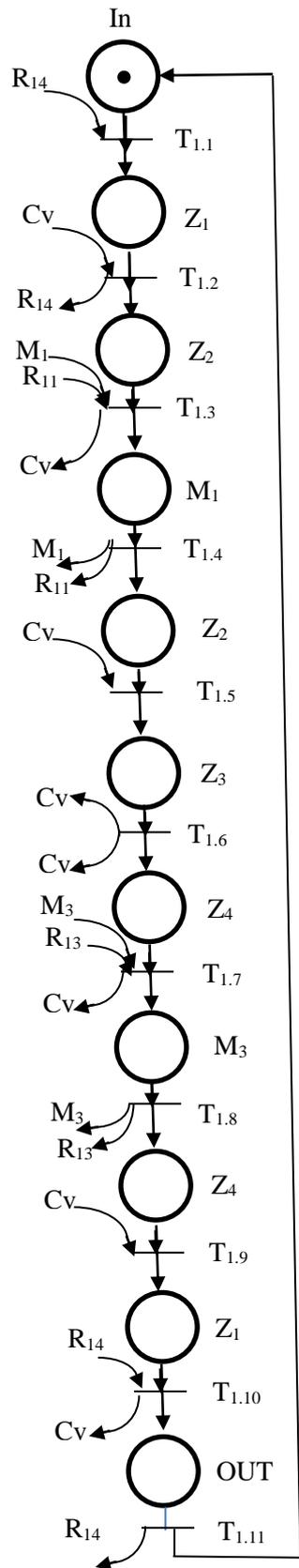
IV.4.2 Algorithme de conduite :



• **Modélisation à base des Rdp :**

On a modélisé les gammes avec les Rdp, on a associé les places à des actions et les transitions à des évènements, ensuite on va donner un exemple sur le calcul des marquages en utilisant l'équation fondamentale suivante : $M_{k+1} = M_k + C(p, t) * D_k$

$D_k = E * M_0$



- Association des transitions à des évènements :

Gamme A :

T1.1 : La durée de temps nécessaire pour le parcours INPUT vers Z₁ effectué par le R₁₄

T1.2 : Disponibilité R₁₄

T1.3 : Disponibilité de la machine M₁ et R₁₁

T1.4 : La durée de temps nécessaire pour la production d'une pièce avec la machine M₁

T1.5 : La durée de temps nécessaire pour le parcours Z₂ vers Z₃ effectué par le convoyeur

T1.6 : Disponibilité convoyeur

T1.7 : La durée de temps nécessaire pour le parcours Z₄ vers M₃ effectué par le R₁₃

T1.8 : Disponibilité de la machine M₃ et R₁₃

T1.9 : La durée de temps nécessaire pour le parcours Z₄ vers Z₁ effectué par le convoyeur

T1.10 : La durée de temps nécessaire pour le parcours Z₁ vers OUT effectué par le R₁₄

T1.11 : Disponibilité d'une dans le stock

On a pris la gamme A comme exemple :

$$E = \begin{matrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \\ P_1 & | & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & | \\ P_2 & | & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & | \\ P_3 & | & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & | \\ P_4 & | & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & | \\ P_5 & | & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & | \\ P_6 & | & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & | \\ P_7 & | & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & | \\ P_8 & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & | \\ P_9 & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & | \\ P_{10} & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & | \\ P_{11} & | & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & | \end{matrix} \quad M_0 = \begin{matrix} | & 1 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \\ | & 0 & | \end{matrix}$$

$$D_0 = E * M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$C(p,t) = \begin{matrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \\ P_1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ P_2 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_3 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_4 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_5 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_6 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ P_7 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ P_8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ P_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ P_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ P_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{matrix}$$

$$M_1 = M_0 + c(p,t) * D_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Fonction ou critère d'évaluation $f_i = \sum_{j=1}^n dur_j + \sum \text{temps d'attente}$

n : nbre d'opération par gamme.

IV.5 Résultats

Pour pouvoir mieux illustrer l'efficacité de la méthode hybride séquentielle (composé du séquencement de deux métaheuristiques AG + RT) des comparaisons aura lieu entre la méthode hybride et la méthode recherche tabou

Pour comparer entre les deux méthodes nous devons calculer les trois critères à optimiser selon :

$$\text{Tauxdequalité} = \frac{\text{temps net}}{\text{MAKESPAN}},$$

$$\text{Fréquence d'exécution} = \frac{\text{nombredepiècesproduites}}{\text{MAKESPAN}},$$

$$\text{Tempsmoyenpourproduireunepièce} = \frac{\text{MAKESPAN}}{\text{nombredepiècesproduites}},$$

Dans les tableaux suivants nous avons effectué plusieurs tests d'optimisations en changeant le critère à optimiser. Après plusieurs tests nous avons obtenu les statistiques montrant l'évolution de chaque méthode d'optimisation.

- La méthode hybride séquentielle

Nbre de cycle	5	55	105	155	205	255	305
Makespan	147	1594	3040	4479	5904	7341	8782
Nbre de pièces	14	155	298	439	580	722	864
Taux de qualité (%)	85,73	86,90	87,03	87,05	86,97	86,99	86,99
Freq d'exécution de l'algorithme	0,095	0,097	0,098	0,098	0,098	0,098	0,098

Temps moy de production	10,53	10,28	10,20	10,20	10,17	10,16	10,16
-------------------------	-------	-------	-------	-------	-------	-------	-------

Tableau 4-1 : La méthode hybride séquentielle

• **La méthode de recherche tabou**

Nbre de cycle	5	55	105	155	205	255	305
Makespan	164	1682	3206	4700	6221	7739	9233
Nbre de pièces	14	149	281	411	543	675	805
Taux de qualité (%)	76,82	78,59	78,97	79,14	79,24	79,33	79,36
Freq d'exécution de l'algorithme	0,085	0,088	0,087	0,087	0,087	0,087	0,087
Temps moy de production	11,71	11,28	11,40	11,43	11,45	11,46	11,46

Tableau 4-2 : La méthode de recherche tabou

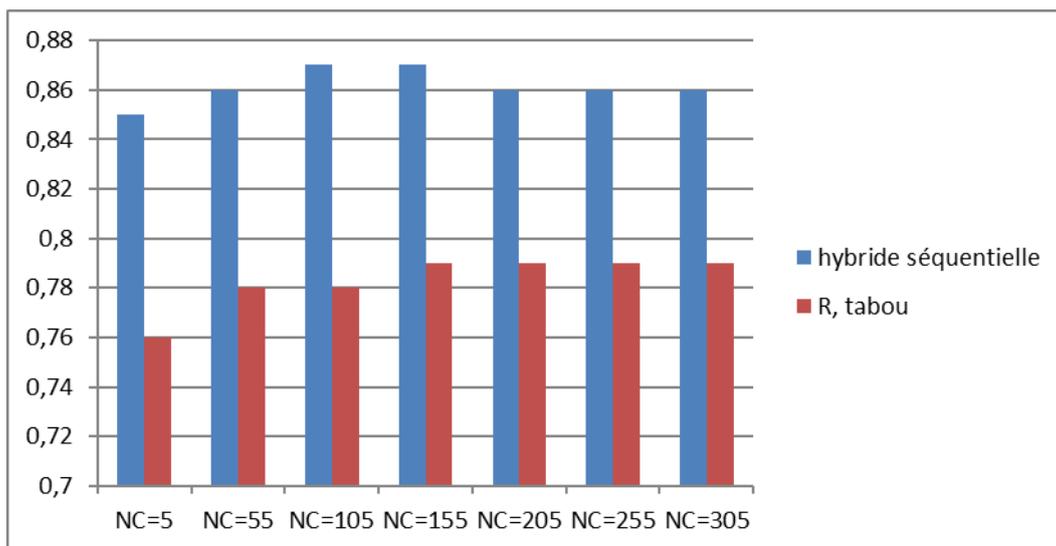


Figure 4-1 : Comparaison entre les deux méthodes par rapport au taux de qualité

On remarque que le taux de qualité en utilisant la méthode hybride séquentielle donne des résultats plus performants que le taux de qualité en utilisant la méthode R. Tabou.

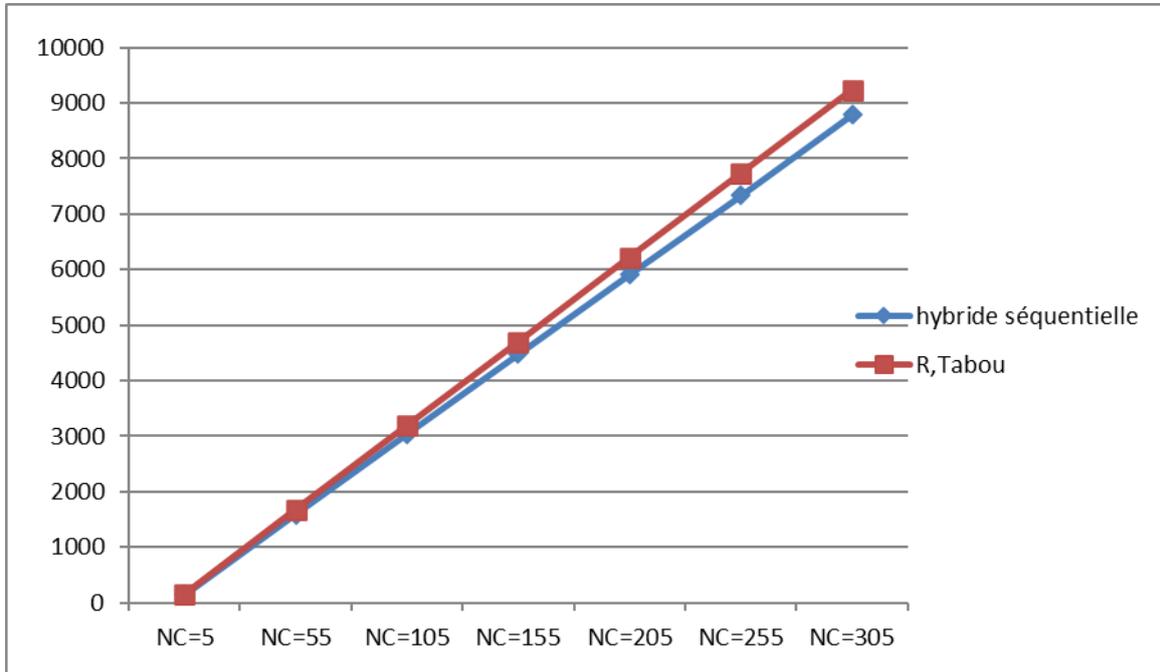


Figure 4-2: le MAKESPAN pour les deux méthodes

On remarque que dans les deux méthodes, le MAKESPAN augmente en augmentant le nombre de cycle mais d'une façon inférieure pour la méthode hybride séquentielle .

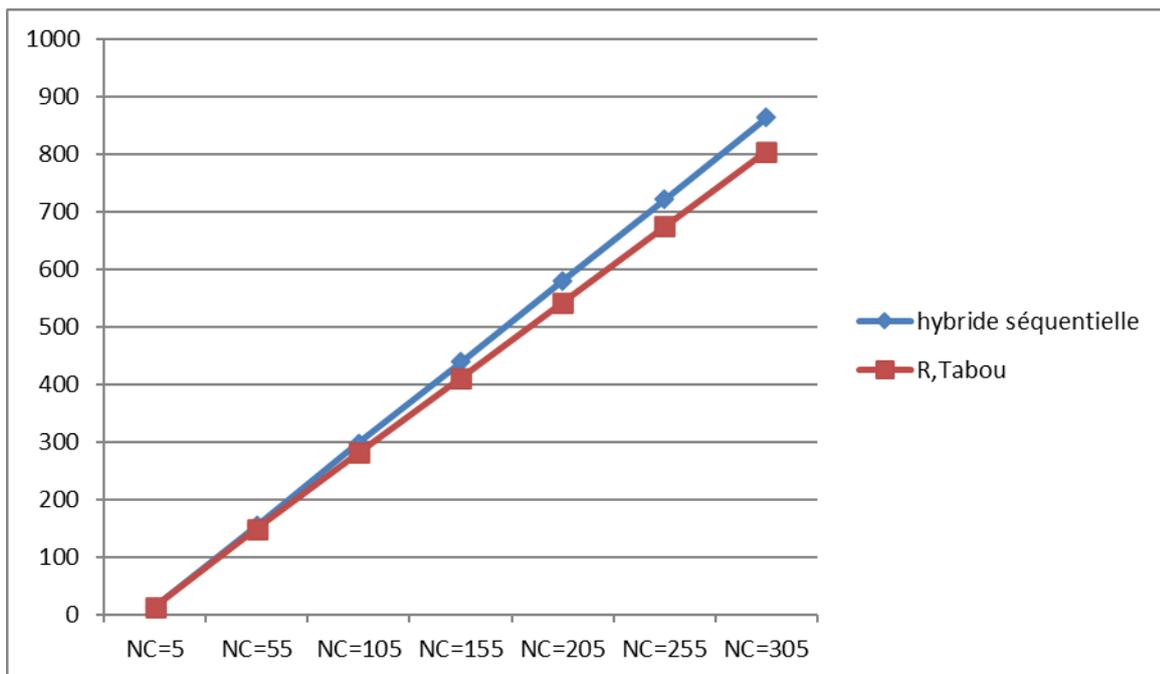
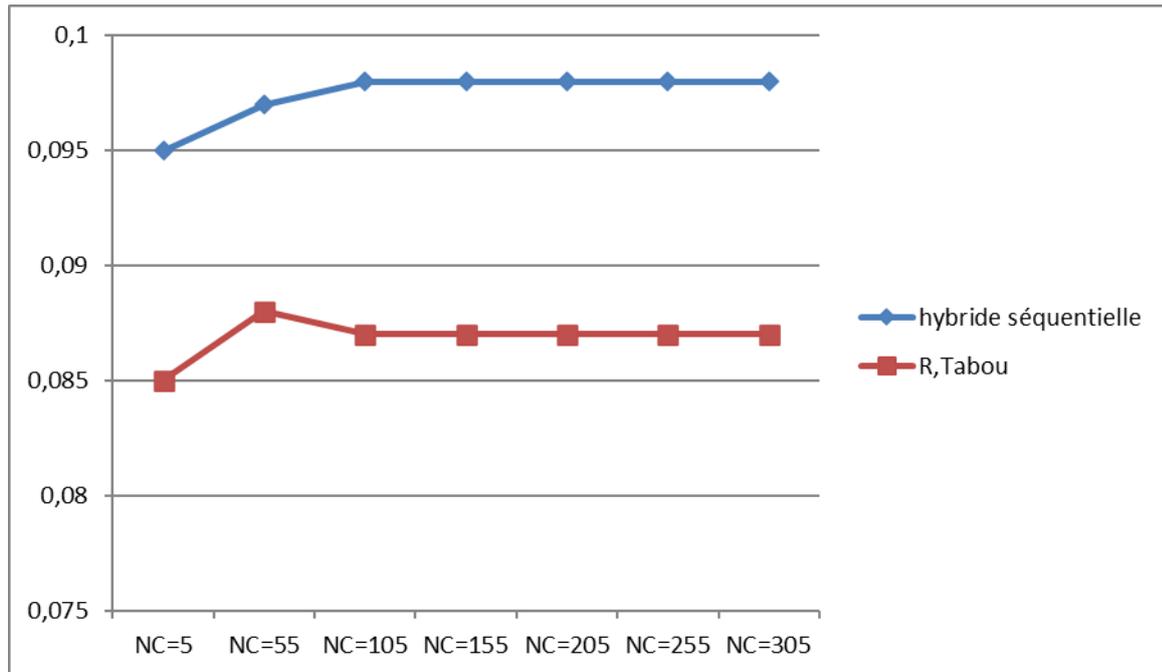


Figure 4-3 : nombre de pièce obtenue par l'application des deux méthodes

D'après le graphe, le nombre de pièce produite en utilisant la méthode hybride séquentielle est plus grand que celle utilisé par la méthode Recherche Tabou.

**Figure 4-4 : Fréquence d'exécution de l'algorithme pour les deux méthodes**

La fréquence d'exécution en utilisant la méthode hybride séquentielle est supérieure que la fréquence en utilisant l'autre méthode. Ce qui nous informe que l'hybride séquentielle rend le système plus rapide.

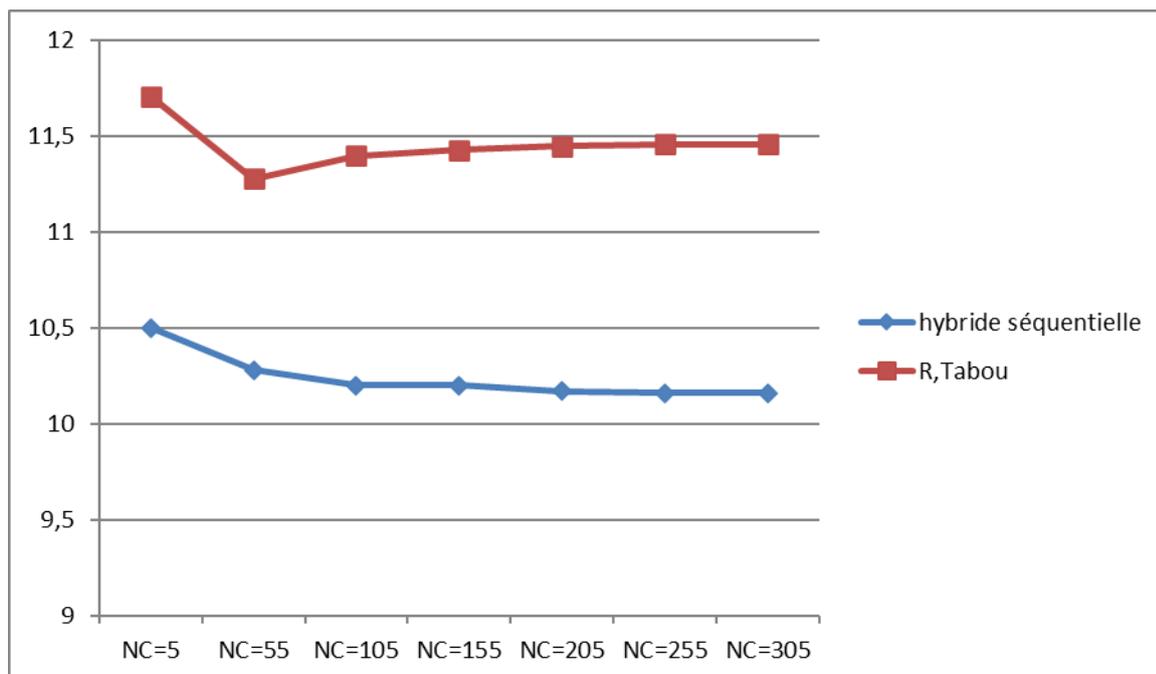


Figure 4-5: temps moyen de production des deux méthodes

Cette figure confirme notre dernière remarque ; en utilisant la méthode hybride séquentielle celle-ci donne des temps moyens inférieurs que celle utilisé par la méthode recherche tabou.

– Comparaison entre les deux méthodes

D'après les résultats de simulation obtenus nous pouvons faire les remarques suivantes :

1. Du point de vue productivité la méthode hybride séquentielle à donner de résultats satisfaisants que celle de la recherche tabou.
2. Du point de vue facilité d'élaboration la méthode tabou est plus simple à programmer.
3. Du point de vue richesse de paramètres et possibilités d'adaptation de l'algorithme hybride à donner une panoplie de résultats intéressantes.

De manière globale, nous pouvons conclure que la méthode hybride séquentielle a fournis de meilleurs résultats.

IV.6 Conclusion

Une comparaison entre les deux méthodes développées a été réalisée sur le problème.

Elles sont efficaces si on se réfère au nombre d'évaluations pour atteindre l'optimum et au respect des contraintes. Nos résultats obtenus montrent que la méthode hybride séquentielle fournit de meilleures solutions que la méthode de recherche Tabou. La qualité des solutions trouvées par les deux méthodes dépend de leur paramétrage, les algorithmes convergent vers un optimum global selon l'efficacité des paramètres choisis de la méthode utilisée.

Le choix d'une bonne représentation de configuration (solution) est un facteur influençant sur l'efficacité de la méthode choisie.

Dans certains cas il arrive que le statut Tabou empêche d'atteindre une solution intéressante, car il peut être restrictif. Il faut éviter de parcourir une branche dans la recherche d'une solution

Conclusion générale

A travers ce travail, nous essayons d'apporter quelques contributions à l'étude des problèmes d'ordonnement dans des systèmes complexes et flexibles à événements discrets. L'originalité de ces technologies tient à l'utilisation d'outils de modélisation dédiés à la description de systèmes à événements discrets (réseaux de Pétri) d'une part, et à l'adaptation de méta-heuristiques aux problèmes d'optimisation d'autre part. Système de production. Les algorithmes et méthode hybride séquentielle développés autour de la recherche sur les tabous ont donné des résultats tout à fait satisfaisants. Le critère d'optimisation est conservé comme MAKESPAN, exprimé en unités de temps, et nous sélectionnons la mise à l'échelle en calculant la productivité de chaque type d'opération. Ce paramètre nous permet de comparer différents algorithmes développés. La variété des hypothèses et la variété des paramètres spécifiques de chaque méta-heuristique se reflètent dans les résultats obtenus, l'utilisation intensive d'algorithmes et l'exploration approfondie des espaces de solutions acceptables. En effet, lorsque l'algorithme se termine, un ensemble de tâches correctement planifiées est obtenu sur des ressources partagées. La mise en œuvre de cet algorithme permet d'observer la nécessité de réduire la convergence rapide du temps total de production. On peut donc dire que le choix de l'un ou de l'autre dépend des conditions de fonctionnement du système, des buts recherchés et des contraintes imposées.

Bibliographie

- [1] : Ghoul.R.Hadiby, "Modélisation et conduite des systèmes de production flexible par les réseaux de pétri", thèse de doctorat d'état de l'université de Annaba, 8 septembre 2003
- [2] : <http://dictionnaire.sensagent.leparisien.fr/Flow-shop/fr-fr/>
- [3] : <http://www.uqac.ca/rebaine/8INF806/techniquedebranchandboundcourshiver2005.pdf>
- [4] : http://mistic.heigvd.ch/taillard/codes.dir/comparaison.dir/implantation_meta_heuristiques.pdf
- [5] : <https://www.gerad.ca/~alainh/Metaheuristiques.pdf>
- [6] : <https://tel.archives-ouvertes.fr/tel-00905604/document>
- [7] : http://www8.umoncton.ca/umcmcmormier_gabriel/systemesintelligents/AG.pdf
- [8] : K. Tamani, "Développement d'une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production," Thèse de Doctorat, 2008.
- [9] : Namane Med Redha «optimisation d'une cellule de production flexible par la methode hybride gras» .master 2 d'état de l'université de Annaba 2019
- [10] : Rene david et hassan alla 'du grafcet aux réseaux de pétri ' traite des nouvelles technologique « série automatique » ED, hermès 2eme édition 1998
- [11] : Thèses d'Internet de types fichier PS
- [12] : Yousfi soufi « Modélisation d'un processus hybride par réseaux de pétri cas d'un trafic urbain » , Université 8 Mai 1945- Guelma, juin 2011.