

République algérienne démocratique et populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique

**UNIVERSITE HADJ LAKHDHAR BATNA**

Faculté de Technologie  
Département de Génie Industriel

---

**MEMOIRE**

Présenté  
En vue de l'obtention du diplôme de  
**Magister**  
**Spécialité Génie Industriel**  
Option Génie Industriel et Productique

**Par**

**HADRI Abdelkader**  
*Ingénieur d'état en Génie Industriel*  
*Université de Batna*

*Thème*

---

***L'ORDONNANCEMENT PAR INSERTION EN  
TEMPS REEL DE LA PRODUCTION DANS UN  
ATELIER FLEXIBLE***

---

<i>H. SMADI</i>	<i>MCA Université de Batna</i>	<i>Président</i>
<i>B. Azoui</i>	<i>Prof Université de Batna</i>	<i>Rapporteur</i>
<i>A.E. Bougloula</i>	<i>MCB Université de Batna</i>	<i>Co-rapporteur</i>
<i>H. KALLA</i>	<i>MCA Université de Batna</i>	<i>Examineur</i>
<i>A. DIB</i>	<i>MCA Université d'Oum-el-Bouaghi</i>	<i>Examineur</i>

*Année : 2011/2012*

# *REMERCIEMENT*

*Tout d'abord El-hamdoulillah, c'est lui qui m'a donné l'effort, le temps et (el-taofik) pour réaliser ce mémoire.*

*J'adresse mes plus vifs remerciements à Monsieur Prof. B. Azoui et Monsieur Dr. A.E. Bougloula pour avoir accepté de me suivre dans l'élaboration de ce travail et pour leurs conseils et leur disponibilité pendant toute la durée de la réalisation de ce mémoire.*

*Je souhaite remercier Dr. H. SMADI, Dr. A. DIB et Dr. H. KALLA pour nous avoir fait l'honneur d'être membres du jury. Ainsi que pour avoir consacré une partie de leur temps précieux pour lire et corriger ce mémoire.*

*Un remerciement très chaleureux à mes parents, ma femme et toute ma famille pour leur présence constante à mes côtés et leur soutien.*

*Je tiens à remercier tous les enseignants et les étudiants de Gini Industriel qui ont ma donnée des aides importantes.*

*« On a toujours assez du temps si on  
l'exploite bien »*

# Table des matières

---

<b>Table des matières</b>	i
<b>Liste des Figures</b>	iv
<b>Liste des Tableaux</b>	v
<b>Glossaire</b>	vi
<b>Introduction général</b>	1
<b>Chapitre I Problèmes d’ordonnancement dans les systèmes de production</b>	
I.1 Introduction	3
I.2 Présentation générale des problèmes d’ordonnancement	3
I.2.1 Définitions	3
I.2.2 Typologie des problèmes d’ordonnancement	6
I.2.2.1 Le problème à une machine	6
I.2.2.2 Les problèmes à machines parallèles	6
I.2.2.3 Les problèmes d’atelier multi-machines	7
I.2.3 Méthodes de résolution	8
I.2.3.1 Méthodes exactes	8
I.2.3.2 Les méthodes approchées ou méta-heuristiques	8
I.2.4 Représentations des solutions	8
I.3. Ordonnancement d’atelier en temps réel	10
I.3.1 Définition	10
I.3.2 Approches pour l’ordonnancement temps réel d’atelier	10
I.3.3 Méthode d’insertion de tâches	11
I.4 Conclusion	12
<b>Chapitre II Description de systèmes de production</b>	
I.1 Introduction	13
II.2 Systèmes de production	13
II.2.1 Définition	13
II.2.2 Classification des systèmes	13
II.2.2.1 Classification selon la nature et le volume des flux physiques	13

---

II.2.2.2 Classification selon le mode de pilotage	14
II.2.3 Décomposition et organisation du système de production	15
II.2.4 La gestion de systèmes de production	16
II.2.4.1 Définition et rôle de la gestion de production	16
II.2.4.2 Organisation hiérarchique de la gestion de production	16
II.3.4.3 Fonction ordonnancement dans la gestion de production	17
II.3 Flexibilité dans les systèmes de production	18
II.3.1 Définitions de La flexibilité	18
II.3.2 Dimensions de flexibilité	18
II.4 Système à étudier et hypothèses proposés	19
II.4.1 Organisation du système et modèle de transport	19
II.4.1.1 Les machines	20
II.4.1.2 Les commandes	20
II.4.1.3 Les tâches	21
II.4.2 Formulation du problème et notations utilisée	21
II.5 Conclusion	22
 <b>Chapitre III Méthode d’insertion des nouvelles commandes</b>	
III.1. Introduction	23
III.2. Première partie : cas sans délai de livraison	23
III.2.1 Définition d’une position	24
III.2.2 Notion d’admissibilité	25
III.2.3 Détermination des dates débuts des positions	26
III.2.4 L’insertion de la nouvelle commande	29
III.3 Approche qui prend en compte le délai de livraison	30
III.3.1 Définition d’une position admissible	31
III.3.2 Procédure de décalage	32
III.3.2.1 Déroulement de la procédure	32
III.3.2.2 Création d’une position admissible et détermination du nouvel ordre	32

---

III.3.3	Calcul de la nouvelle valeur de début des itérations	35
III.3.4	Choix du critère et définition de la bonne solution	36
III.3.4.1	Durée totale d'exécution « Maksepan »	36
III.3.4.2	Coût de décalage	36
III.4	Exemple d'application	38
III.4.1	Données concernant le plan prévisionnel	38
III.4.2	Application de l'approche d'insertion des tâches	40
III.4.3	Analyse par variation de la date d'apparition	44
III.4.3.1	Résultats selon le Makespan	44
III.4.3.2	Résultats selon le coût de décalage	45
III.4.4	Analyse par variation des commandes	47
III.4.3.1	Résultats selon le Makespan	47
III.4.3.2	Résultats selon le coût de décalage	48
III.5	Conclusion	49
	<b>Conclusion générale</b>	51
	<b>Bibliographie</b>	53
	<b>Annexes</b>	

# Liste des figures

<b>Figure I.1</b>	Caractéristique d'une tâche	4
<b>Figure I.2</b>	Exemple d'une organisation de type Flow Shop à 3 machines	7
<b>Figure I.3</b>	Exemple d'une organisation de type Job Shop à 5 machines	7
<b>Figure I.4</b>	Graphe Potentiel-Tâches d'un ordonnancement	9
<b>Figure I.5</b>	Exemple de Diagramme de Gantt	10
<b>Figure II.1</b>	Décomposition classique d'un système de production	16
<b>Figure II.2</b>	Système de production à quatre machines	20
<b>Figure II.3</b>	Diagramme de Gant d'un ordonnancement prévisionnel	22
<b>Figure III.1</b>	Organigramme de l'approche sans délai de livraison	24
<b>Figure III.2</b>	Les différents cas possible d'une position de la tâche $O_i^a$	25
<b>Figure III.3</b>	Calcule de la nouvelle valeur de $t_1^a$ « cas de la deuxième proposition »	27
<b>Figure III.4</b>	Exemple d'une position de la tâche $O_i^a$	28
<b>Figure III.5</b>	Calcule de la nouvelle valeur de $t_1^a$ « cas de la troisième proposition »	28
<b>Figure III.6</b>	Organigramme de la deuxième procédure	31
<b>Figure III.7</b>	Exemple d'une opération de décalage effectuée pour insérer la tâche $O_i^a$	32
<b>Figure III.8</b>	Un ordonnancement prévisionnel avec une position pour la nouvelle commande	33
<b>Figure III.9</b>	Diagramme de Gantt du plan prévisionnel	39
<b>Figure III.10</b>	Résultat de la première itération	42
<b>Figure III.11</b>	Résultat de la quatrième itération	43
<b>Figure III.12</b>	Variation du Makespan selon la date d'apparition	45
<b>Figure III.13</b>	Variation du coût de décalage	46
<b>Figure III.14</b>	Histogramme du Makespan suivant la variation des commandes	48
<b>Figure III.15</b>	Histogramme du coût suivant la variation des commandes	49

# Liste des Tableaux

<b>Tableau III.1</b>	Caractéristiques des commandes prévisionnelles	38
<b>Tableau III.2</b>	Caractéristiques des marges libres existantes dans le plan prévisionnel	39
<b>Tableau III.3</b>	Caractéristiques de la nouvelle commande	40
<b>Tableau III.4</b>	Résultats obtenu pour chaque itération	40
<b>Tableau III.5</b>	Le choix de la bonne solution	41
<b>Tableau III.6</b>	Résultats des valeurs du Makespan selon la variation des dates d'apparition de la nouvelle commande	44
<b>Tableau III.7</b>	Les coût de décalage unitaire des machines	45
<b>Tableau III.8</b>	Résultats retenus pour les valeurs du coût de décalage	46
<b>Tableau III.9</b>	Caractéristique des commandes choisies	47
<b>Tableau III.10</b>	Résultats du Makespan obtenu par variation des commandes	47
<b>Tableau III.11</b>	Résultats du coût de décalage obtenu par variation des commandes	48

## Glossaire

$m$ :	Nombre des machines existantes dans le système
$m_i$ :	La $i^{\text{ème}}$ machine
$J$ :	Ensemble des jobs (travaux)
$n$ :	Nombre des jobs constituant l'ensemble $J$
$j$ :	Le $j^{\text{em}}$ job de l'ensemble $J$
$n_j$ :	Nombre des tâches constituant le job $j$
$O_{ij}^{uk}$ :	La tâche $i$ du job $j$ qui sera réalisée dans l'ordre $u$ sur la machine $k$
$p_{ij}^{uk}$ :	Durée de la tâche $O_{ij}^{uk}$
$t_{ij}^{uk}$ :	Date de début de la tâche $O_{ij}^{uk}$
$c_{ij}^{uk}$ :	Date de fin de la tâche $O_{ij}^{uk}$
$MO_k$ :	Ensembles des marges occupées concernant la machine $k$
$ML_k$ :	Ensembles des marges libres concernant la machine $k$
$mo_{uk}$ :	La marge occupée numéro $u$ de la machine $k$
$ml_{uk}$ :	La marge libre numéro $u$ de la machine $k$
$dd_{uk}$ :	Date début de la marge $ml_{uk}$
$df_{uk}$ :	Date fin de la marge $ml_{uk}$
$j^a$ :	Le job qui représente la nouvelle commande urgente
$T^a$ :	La date d'apparition de la nouvelle commande
$DI^a$ :	Le délai de livraison de la nouvelle commande
$M^a$ :	Ensembles des machines nécessaires pour réaliser la commande urgente
$n^a$ :	Nombre des machines utilisées par la commande urgente
$O_i^a$ :	La tâche $i$ de la commande urgente
$t_i^a$ :	Date début de la tâche $O_i^a$
$c_i^a$ :	Date fin de la tâche $O_i^a$
$p_i^a$ :	La durée de la tâche $O_i^a$
$dd_k(\text{secc } t_i^a)$ :	Date de début de la plus proche marge libre de la machine $k$ après la date $t_i^a$

---

$Ra_i$ :	Le rapport d'admissibilité concernant la tâche $O_i^a$
$Ra_i^{It}$ :	Le rapport d'admissibilité concernant la tâche $O_i^a$ calculé dans l'itération $It$
$Nd_{It}$ :	L'ensemble des tâches qui n'ont pas des positions admissibles dans l'itération $It$
$Rp_i$ :	Le rapport de passage concernant la tâche $O_i^a$
$Rp_i^{It}$ :	Le rapport de passage concernant la tâche $O_i^a$ calculé dans l'itération $It$
$Rp_{max}$ :	Le rapport de passage maximum
$Rp_{min}$ :	Le rapport de passage minimum
$Rp_w$ :	Le rapport de passage concernant la tâche $O_w^a$
$Rd_i(O_{ij}^{uk})$ :	Le temps de décalage de la tâche $O_{ij}^{uk}$ généré par l'insertion de la tâche $O_i^a$
$C_{max}$	La durée totale d'exécution (Makespan)
$C_{max}^{It}$	La durée totale d'exécution (Makespan) calculé dans l'itération $It$
$CG$ :	Le coût global de décalage
$CG^{It}$ :	Le coût global de décalage calculé dans l'itération $It$
$CU_{m_i}$ :	Le coût unitaire de décalage d'une tâche dans la machine $m_i$
$n_{m_i}$ :	Le nombre des tâches décalées concernant la machine $m_i$
$O_{déc}^{um_i}$ :	La $u^{\text{ème}}$ tâche de la machine $m_i$ qui a été décalée
$Rd_i(O_{déc}^{um_i})$ :	Le temps de décalage de la tâche $O_{déc}^{um_i}$

# Introduction Générale

## Introduction générale

La réactivité des ateliers de production et le taux de satisfaction clients en termes de délais repose pour toute ou partie sur la capacité de ces ateliers à faire face aux événements imprévus aux quels ils sont soumis.

La flexibilité en gestion de production confère aux ateliers cette capacité potentielle en utilisant des méthodes adéquates d'affectation et d'ordonnement des tâches. Les industriels doivent donc maîtriser leur système de production au niveau opérationnel et être capables de réagir sur le très court terme aux événements imprévus tels qu'une modification ou une annulation d'un ordre de fabrication, l'arrivée d'une commande urgente.

Un problème d'ordonnement est défini par un ensemble de travaux à réaliser sur un ensemble de ressources ; de sorte qu'une fonction objectif soit optimisée.

Le problème d'ordonnement en temps réel consiste à adapter en permanence les modalités d'exécutions d'ensemble des tâches par un ensemble des ressources à la situation réel du système considéré. On rencontre souvent ce type de problème dans les ateliers de fabrication travaillant à la commande où le délai de livraison représente une des difficultés majeur [1].

Le problème de type Job Shop est l'un des problèmes de la théorie de l'ordonnement le plus étudié et le plus difficile à résoudre. Il est indispensable de savoir, dans la résolution de ce type de problème, si l'on doit privilégier la qualité de la solution recherchée, la rapidité du temps de calcul ou trouver un compromis. La résolution de manière optimale s'avère donc dans la plupart des cas impossible à cause de son caractère fortement combinatoire. Alors on fait le recours généralement aux méthodes dite méthodes approchées qui donnent des solutions approchées dans un temps raisonnable [2].

Dans ce mémoire, nous nous intéressons plus particulièrement aux problèmes d'ordonnement en temps réel pour un system de type Job Shop à quatre machines où la réalisation des travaux s'effectue suivant un plan prédéfini (prévisionnel). Ce système est soumis à une perturbation de l'environnement représentée par l'occurrence des nouvelles commandes urgentes, qu'il doit les exécuter.

Le problème imposé ici est comment le système doit il réagir lors de l'apparition aléatoire d'une nouvelle commande et quelle est la bonne méthode à utiliser pour l'absorber ?

Pour la résolution de ce problème nous avons proposé une approche approximative qui consiste à générer un ordonnancement en temps réel et qui peut nous donner la possibilité d'introduire la commande aléatoire dans le plan prévisionnel.

L'objectif de cette approche est de fournir au système considéré une capacité non seulement d'absorber une commande supplémentaire, mais aussi de la traiter le plus vite possible pour avoir la possibilité d'absorber une autre commande qui peut apparaître dans le futur. Pour cette raison notre approche se base essentiellement sur l'exploitation des marges libres existantes dans l'ordonnancement prévisionnel. L'idée est d'essayer d'insérer les tâches de la nouvelle commande dans ces marges libres.

Alors, après une introduction générale notre mémoire est divisé en trois chapitres. Les deux premiers chapitres sont conçus essentiellement pour déterminer le cadre générale du travail et pour définir et positionner le problème d'ordonnancement à étudier. Par contre le troisième chapitre a pour objectif de décrire l'approche proposée.

Dans **Le premier chapitre** nous donnons une généralité sur les problèmes d'ordonnancement (ses notions de base, leurs différentes classes et les méthodes utilisées pour les résoudre). Nous abordons ensuite le problème d'ordonnancement en temps réel et les deux types d'approche trouvés dans la littérature pour résoudre ce type de problème. Nous définissons à la fin du chapitre la position de notre problème par rapport aux travaux existants, cela après une présentation d'un état de l'art sur la méthode d'insertion de tâches. **Le deuxième chapitre** consiste à présenter en premier temps les systèmes de production d'une manière générale en définissant leurs principaux composants, leurs différentes classes et leurs manières de gestion et en deuxième temps la plate forme de l'étude et quelques propriétés spécifiques. À la fin de ce chapitre nous donnons une description détaillée du problème traité. L'objectif du **troisième chapitre** est la description détaillée de l'approche de résolution proposée qui sera suivie par une application numérique pour valider l'approche. Les résultats obtenus seront exposés avec des explications. Enfin, ce travail sera complété par une conclusion générale à travers laquelle on exposera les principaux points retirés et on donnera les perspectives à envisager comme suite à ce travail.

# CHAPITRE I

## Problèmes D'ordonnancement Dans Les Systèmes De Production

## I.1 Introduction

L'objectif de ce premier chapitre est de définir et de situer le problème d'ordonnancement visé par ce travail. Alors dans ce chapitre nous exposons les problèmes d'ordonnancement et leurs notions de base d'une manière générale. Puis nous présentons leurs classifications et les méthodes de résolution utilisées. L'ordonnancement en temps réel, cas particulier de ces problèmes, est aussi abordé. Nous présentons sa définition et les deux types d'approches de résolution destinées à ce type de problème. Et afin de situer notre problème d'ordonnancement par rapport à l'ensemble traité dans la littérature, un état de l'art rapide sur la méthode d'insertion de tâches est exposé.

## I.2 Présentation générale des problèmes d'ordonnancement

### I.2.1 Définitions

#### ➤ L'ordonnancement

De nombreuses définitions du problème d'ordonnancement sont proposées dans la littérature. Parmi les définitions les plus souvent rencontrées sont :

« Un problème d'ordonnancement est défini par un ensemble de jobs à réaliser sur un ensemble de ressources; de sorte qu'une fonction objectif soit optimisée » [3].

« Le problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement,...) et de contraintes portant sur l'utilisation et la disponibilité de ressources requises » [1, 4, 5].

La solution d'un problème d'ordonnancement est un "ordonnancement". Il définit pour chaque tâche du problème les dates du début (ou de la fin) de leur exécution et les ressources auxquelles elle sera affectée. Un problème d'ordonnancement peut être traité en fonction d'un ou plusieurs objectifs ou critères de performance (durée totale d'exécution, somme des retards, équilibrage de la charge,...) [6].

#### ➤ Les tâches

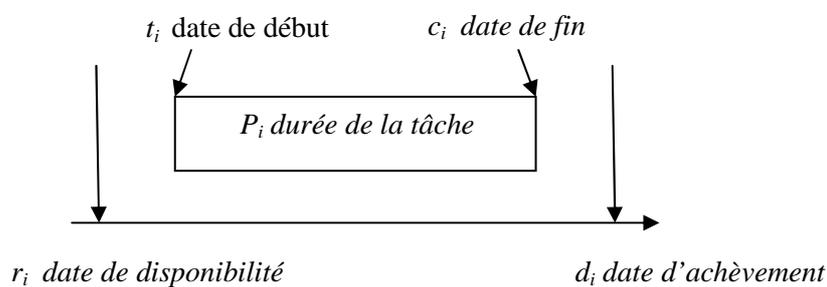
Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début  $t_i$  ou de fin  $c_i$ , dont la réalisation est caractérisée par une durée  $P_i$  (on a  $c_i = t_i + P_i$ ) et par l'intensité  $a_i^k$  avec laquelle elle consomme certains moyens  $k$  ou ressources. Cette intensité est supposée comme constante durant l'exécution de la tâche [1].

La tâche ou « l'opération » peut accepter une interruption de son exécution dans certains problèmes, on dit qu'elle est préemptive. Dans d'autres problèmes cette interruption est non permise, la tâche donc est non préemptive.

Généralement une tâche est aussi caractérisé par :

- une date de disponibilité  $r_i$  qui correspond à la date avant laquelle la tâche ne peut pas être commencée
- et une date d'achèvement souhaitée  $d_i$

La Figure suivante donne une représentation de la tâche en désignant ses principales caractéristiques



**Figure I.1 :** Caractéristique d'une tâche [7]

Dans ce mémoire nous nous sommes intéressés uniquement par la durée, la date de début et la date de fin de la tâche et nous considérons toujours que  $t_i$  et  $c_i$  sont coincidés respectivement avec  $r_i$  et  $d_i$  et nous considérons aussi que ces dates sont flexibles c'est-à-dire qu'on peut le déplacer dans le temps.

### ➤ Les ressources

« Une ressource  $k$  est un moyen technique ou humain requis pour la réalisation d'une tâche et disponible en quantité limitée, sa capacité  $c_k$  est supposé constante » [4].

Une ressource peut être renouvelable, comme c'est le cas pour les hommes, les machines, l'outillage,..., c'est à dire qu'elle peut être utilisée et qu'une fois la tâche terminée, elle est à nouveau disponible. Mais elle peut aussi ne pas l'être, on parle alors de ressource consommable telle que la matière première, les produits d'entretien ou encore les budgets.

Une ressource est de plus doublement contrainte si son utilisation instantanée et sa consommation globale sont limitées. Si une ressource ne peut exécuter qu'une seule tâche à la fois elle est dite disjonctive (ou non partageable) comme c'est le cas pour une machine outil ou un robot manipulateur. Dans le cas où une ressource pourrait être utilisée dans

le traitement de plusieurs tâches simultanément, comme dans le cas où plusieurs ressources soit utilisées pour la même tâche, on parle de ressource cumulative (ou partageable) [6].

### ➤ Les contraintes

Les contraintes expriment des restrictions sur les valeurs que peuvent prendre certaines variables. Dans les problèmes d'ordonnancement, deux types de contraintes sont distinguées : les contraintes *temporelles* et les contraintes *de ressources*.

- Les contraintes temporelles comprennent les contraintes de temps alloué, qui correspondent généralement aux impératifs liés aux tâches (délai de livraison par exemple) ou encore à la durée totale d'un ordonnancement. Elles comprennent les *contraintes d'antériorité* ou de *précédence* qui correspondent à des contraintes de cohérence technologique qui positionnent les tâches les unes par rapport aux autres. Elles comprennent aussi *les contraintes de calendrier* qui correspondent, par exemple, aux plages horaires de travail, etc.
- Les contraintes de ressources, quant à elles, traduisent la disponibilité des ressources et le fait qu'elle soit en quantité limitée. Deux types de contraintes de ressource, liées à la nature cumulative ou disjonctive des ressources, peuvent alors être distingués. Les ressources disjonctives ne peuvent être utilisées que par une tâche à la fois. Les ressources cumulatives, quant à elles, peuvent être utilisées par plusieurs tâches simultanément, comme dans le cas d'un ensemble de ressources, par exemple [4, 8].

### ➤ Les objectifs ou les critères d'évaluation

Dans la résolution des problèmes on cherche souvent, soit à minimiser, soit à maximiser un critère correspondant à une amélioration suivant au moins l'un des trois facteurs: coût, qualité ou délais. Alors on distingue plusieurs classes d'objectifs concernant un ordonnancement [4] :

- les objectifs liés au temps : on trouve par exemple la minimisation du temps total d'exécution, du temps moyen d'achèvement, des durées totales de réglage ou des retards par rapport aux dates de livraison,
- les objectifs liés aux ressources : maximiser la charge d'une ressource ou minimiser le nombre de ressources nécessaires pour réaliser un ensemble de tâches sont des objectifs de ce type,
- les objectifs liés au coût : ces objectifs sont généralement de minimiser les coûts de lancement, de production, de stockage, de transport, etc.,
- les objectifs liés à une énergie ou un débit.

## **I.2.2 Typologie des problèmes d'ordonnancement**

Plusieurs classifications ont été proposées dans la littérature. On trouve par exemple :

- Une classification standard proposée par Graham et al. en 1979 [9]. Dans cette classification les problèmes sont décrits par la notation  $\alpha/\beta/\gamma$ , où le premier paramètre ( $\alpha$ ) décrit l'organisation de l'atelier et le nombre de machines de quelles il est constitué. Le deuxième paramètre ( $\beta$ ) définit les caractéristiques des travaux à réaliser et les contraintes associées. Le dernier paramètre ( $\gamma$ ) représente le critère d'optimisation sur lequel l'ordonnancement sera évalué.
- Une classification simple basée sur l'organisation de l'atelier classifie les problèmes d'ordonnancement en trois classes (le problème à une machine, les problèmes à machines parallèles et les problèmes d'atelier multi-machines)

### **I.2.2.1 Le problème à une machine**

Un problème est dit problème à une machine (ou ressource unique) si on ne dispose qu'une seule ressource pour réaliser un ensemble de travaux. Ces derniers sont constitués d'une seule tâche pour chacun d'eux. Dans ce type de problème la résolution consiste à trouver l'ordre optimal d'exécution de ces tâches vis-à-vis d'un critère donné.

### **I.2.2.2 Les problèmes à machines parallèles**

Les problèmes d'ordonnancement à machines parallèles se caractérisent par l'existence de plus d'une ressource. Elles représentent un cas particulier de problèmes multi-machines où les machines sont disposées en parallèle. On peut distinguer trois types des problèmes à machines parallèles:

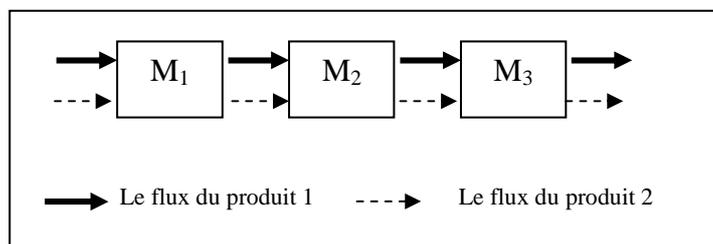
- Les problèmes à machines identiques : les durées opératoires sont égales et ne dépendent donc pas des machines,
- Les problèmes à machines uniformes : la durée d'une opération varie uniformément en fonction de la performance de la machine choisie,
- Les problèmes à machines indépendantes (non liées) : les durées opératoires dépendent complètement des machines utilisées.

### I.2.2.3 Les problèmes d'atelier multi-machines

Pour ce type de problème les ressources nécessaires pour réaliser l'ensemble des travaux sont multiples. L'organisation de ces ressources et le type de passage des produits entre eux génèrent trois sous types de problèmes :

#### ✓ Problème de type Flow Shop

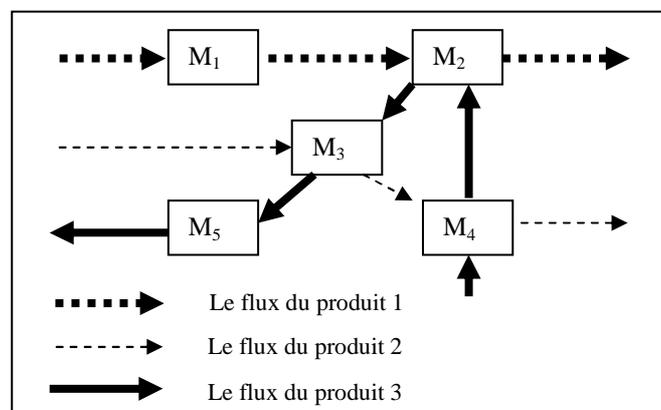
Les produits sont procédés dans une direction unique selon la même gamme de fabrication. Les machines consacrées à la réalisation de ces produits sont inévitablement disposées en série (Figure I.2). Ce type de problème est souvent rencontré dans les systèmes produisant les matières liquides ou gazeuses.



**Figure I.2 :** Exemple d'une organisation de type Flow Shop à 3 machines

#### ✓ Problème de type Job Shop

Le problème de Job Shop consiste à réaliser un ensemble de  $n$  Job (travail) sur un ensemble de  $m$  machines en cherchant d'atteindre certains objectifs. Chaque Job  $j$  est composé de  $n_j$  tâches devant être exécutées sur les différentes machines selon un ordre préalablement défini. Les travaux ne s'exécutent pas sur toutes les machines et de plus n'ont pas le même ordre d'exécution. En effet chaque travail emprunte le chemin qui lui est propre (Figure I.3).



**Figure I.3 :** Exemple d'une organisation de type Job Shop à 5 machines

### ✓ **Problème de type Open Shop**

Dans le problème de type open shop l'ordre d'exécution de tâches d'un travail est totalement libre, c'est-à-dire aucune contrainte est remarquée sur la précédence entre tâches. Les produits passent alors dans n'importe quelle direction (les gammes sont libres).

## **I.2.3 Méthodes de résolution**

### **I.2.3.1 Méthodes exactes**

Les méthodes exactes de résolution sont des méthodes qui fournissent une solution exacte et optimale en se basant sur des lois bien déterminées [10]. Ces méthodes permettent d'avoir des solutions vis-à-vis un certain critère, sous des conditions bien particulières. Dans ce contexte on trouve par exemple la méthode la plus utilisée pour résoudre les problèmes d'optimisation est la méthode par séparation et évaluation (branch and bound) [11, 12]. D'autres méthodes telles que la programmation linéaire ou la programmation dynamique sont aussi utilisées couramment [13].

L'utilisation de ce type de méthodes s'avère particulièrement intéressant dans les cas des problèmes de petites tailles. Par contre plus le problème est grand plus le recours à ces méthodes est limité, ceci est à cause de l'accroissement du temps du calcul. Les méthodes approchées sont alors les méthodes utilisées quand les méthodes optimales ne permettent pas de résoudre le problème en un temps acceptable.

### **I.2.3.2 Les méthodes approchées ou méta-heuristiques**

Les méta-heuristiques sont des stratégies de recherche itératives de haut niveau, destinées à l'exploitation de l'espace de solutions par l'utilisation de différentes techniques. Ces méthodes n'essayeront pas d'examiner toutes les solutions possibles, mais de trouver dans un temps raisonnable une solution satisfaisante par investigation intelligente de l'espace en s'appuyant sur deux principes : la diversification de la recherche dans tous les endroits de l'espace ; et l'intensification, en exploitant chaque point de l'espace d'état [14].

Un ensemble de méthodes méta-heuristiques largement utilisées, les algorithmes génétiques, le recuit simulé et la recherche Tabou, sont détaillées dans [2, 15].

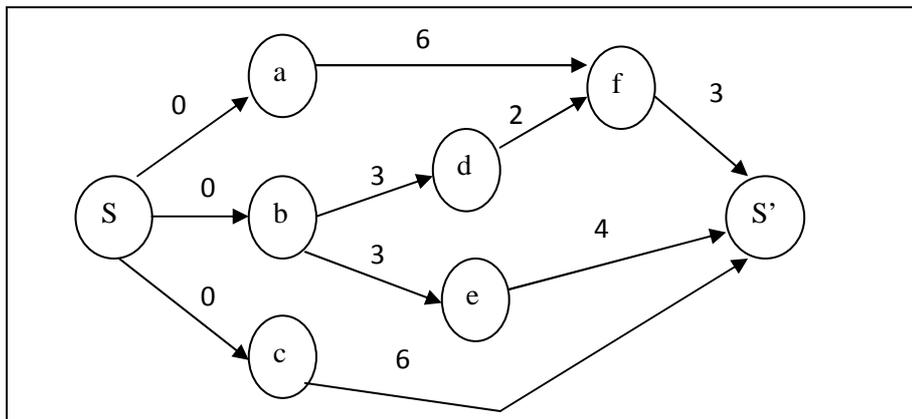
## **I.2.4 Représentations des solutions**

Deux représentations sont souvent rencontrées dans la littérature, Le graphe Potentiel-Tâches et le diagramme de Gantt.

### Le graphe Potentiel-Tâches :

Cette outil graphique a été développé par Roy et Susmann [18] grâce à la théorie des réseaux de Pétri qui ont surtout servi à modéliser les systèmes dynamiques à évènements discrets. Dans ce genre de modélisation, les tâches sont représentées par des nœuds et les contraintes par des arcs, comme le montre la Figure I.4. Ainsi, les arcs peuvent être de deux types :

- les arcs conjonctifs illustrant les contraintes de précédence et indiquant les durées des tâches,
- les arcs disjonctifs indiquant les contraintes de ressources.



**Figure I.4 :** Graphe Potentiel-Tâches (graphe conjonctif)

Dans l'exemple représenté par la Figure I.4 S et S' sont des tâches effectifs représentent successivement le début et la fin de l'ordonnement, par contre les numéros associés aux arcs représente les durées des tâches.

**Le diagramme de Gantt** est un outil graphique développé par 'Henry Gantt' 1917 qui permet de représenter la planification de tâches nécessaires à la réalisation d'ordonnement. Pour un atelier, celui-ci sera composé de plusieurs lignes horizontales, représentant chacune un équipement ou une machine. L'axe horizontal représente le temps. Les tâches seront représentées par des barres d'une longueur proportionnelle à leur durée et seront positionnées sur la ligne correspondant à l'équipement sur lequel elles se dérouleront. On observera donc sur le diagramme l'occupation des machines, l'enchaînement des tâches sur celles-ci ainsi que les dates de début et de fin de chaque tâche [7].

La Figure I.5 représente un exemple de diagramme de Gantt d'un ordonnancement de quatre jobs sur trois machines

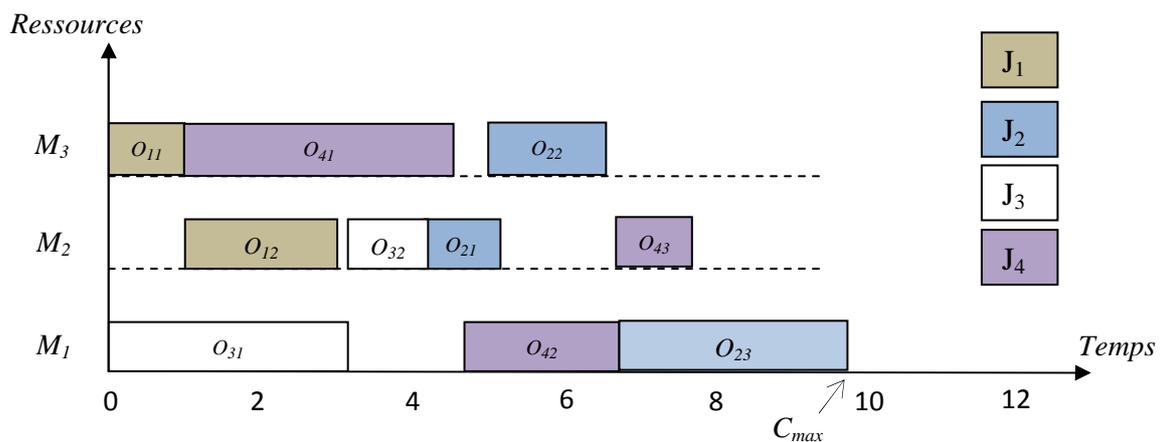


Figure I.5 : Exemple de Diagramme de Gantt

### I.3. Ordonnancement d'atelier en temps réel

#### I.3.1 Définition

Le problème d'ordonnancement en temps réel consiste à adapter en permanence les modalités d'exécutions d'ensemble des tâches par un ensemble des ressources à la situation réel du système considéré [1].

On rencontre souvent ce type de problème dans les ateliers de fabrication travaillant à la commande où le délai de livraison représente une des difficultés majeures. L'ordonnancement en temps réel offre à ces ateliers avec la prise en compte de ses caractéristiques réelles, une capacité de réagir aux déférentes aléas (internes ou externes) aux quels ces ateliers sont soumis. Les travaux de [6, 16] sont destinés à ce type de problème. Ils ont traité le problème d'affectation et d'ordonnancement des tâches de maintenance sous la contrainte temps réel.

Dans le domaine informatique la problématique de l'ordonnancement en temps réel revient à définir dans quel ordre exécuter les tâches sur chaque processeur d'une architecture donnée. Cette problématique est en générale limité à une ressource ou plusieurs ressources en parallèle (processeurs), de plus l'interruption de tâches (préemption) est souvent autorisée contrairement aux ateliers [1]. Les travaux de [17,19] donnent un exemple sur l'ordonnancement en temps réel dans les systèmes informatiques.

#### I.3.2 Approches pour l'ordonnancement temps réel d'atelier

Pour la résolution des problèmes d'ordonnancement en temps réel deux types d'approches sont utilisées. Des approches basées sur un contexte statique et d'autres basées sur un contexte dynamique [20].

Dans le contexte *statique* un ensemble des jobs est supposé connu à l'avance sur un horizon fini. La séquence de tâches est fixée par une approche prévisionnelle traditionnelle.

L'approche en temps réel consiste alors à contrôler le respect de la séquence proposée et les dates de débuts prévues, tout en prenant en compte les perturbations apparues dans le système. Différents travaux ont été réalisés dans ce contexte. On peut citer par exemple les travaux de [6, 19, 21] qui ont utilisés ce type d'approches en tenant en compte les nouveaux jobs inclus dans le plan de fabrication.

Dans le contexte *dynamique* les approches proposées considèrent que l'ensemble des jobs à ordonnancer ne sont pas connus d'avance, la solution ne s'effectue que dès qu'un job survient, l'ordonnancement donc consiste à affecter ses opérations aux ressources disponibles. Tuncel G. [22] a proposé un ensemble des règles heuristiques de base pour l'ordonnancement dynamique des systèmes flexibles de production.

### **I.3.3 Méthode d'insertion de tâches**

Une approche basée sur le contexte statique, c'est la méthode d'insertion des tâches dans un ordonnancement prévisionnel. Cette méthode consiste à sélectionner et choisir le bon endroit (emplacement) où ces nouvelles tâches doivent être incluses. Dans la littérature on trouve que l'utilisation de la méthode d'insertion de tâches n'été pas limitée au problème d'ordonnancement d'une seule ressource, mais aussi pour plusieurs ressources (comme les ressource humaines dans les problèmes de d'ordonnancement de la maintenance). On distingue alors:

#### **➤ Pour les problèmes d'une seule ressource**

Les travaux réalisés dans [19] traitent un problème correspond à la modélisation d'un radar, système composé d'une ressource unique qui exécute deux types de tâches : Tâches répétitives (préventives) comme la surveillance de l'environnement et tâches dont l'occurrence est aléatoire comme la détection d'une nouvelle cible. L'approche proposée consiste à insérer les nouvelles tâches dans l'ordonnancement prévisionnel, avec possibilité de réordonner ultérieurement les tâches déjà ordonnancées si nécessaire. Le critère d'évaluation de l'ordonnancement retenu par [19] est la somme des retards des tâches.

Une autre approche de résolution des problèmes d'ordonnancement en temps réel a été proposée dans [23]. L'objectif de cette méthode est d'ordonner dans les délais, les commandes aléatoires au fur et à mesure de leurs apparitions, toute en veillant à respecter l'exécution des commandes prévues en début de l'ordonnancement. Le point spécifique dans cette approche est que les commandes non acceptées par le système seront rejetées. La réussite de placement de commandes aléatoire est le critère à optimiser dans ce travail.

### ➤ **Pour les problèmes à plusieurs ressources**

Dans [6, 21] les auteurs se sont intéressés par le problème d'affectation et d'ordonnancement des tâches de maintenance corrective où ils ont développé une approche d'insertion dynamique de tâches reposant sur une méthode de recherche de solutions par voisinage. Cette méthode permet d'obtenir non pas une solution dominante, mais un ensemble de solutions non dominées parmi lesquelles le décideur pourra choisir. La méthode est basée principalement sur des opérateurs de voisinage qui vont permettre d'explorer, d'une manière stochastique, de nouvelles solutions.

S'appuyant sur la représentation « graphe potentiel-tâches », [1, 16] ont appliquée la méthode d'insertion sur les problèmes d'ordonnancement où les relations de succession entre tâches d'un même travail ne constitue pas obligatoirement un ordre totale entre ces tâches, inversement des modèles classiques de l'atelier à cheminement multiple (voire [1]).

## **I.4 Conclusion**

Ce premier chapitre a été consacré globalement à la présentation des problèmes d'ordonnancement, leurs principales caractéristiques et l'ensemble des méthodes de résolution utilisées. En particulier ce chapitre est destiné à l'étude du problème d'ordonnancement en temps réel. Deux groupes d'approches dédiées à ce type de problème ont été montrés ; Des méthodes basées sur le contexte statique et des méthodes basées sur le contexte dynamique. La différence essentielle entre ces deux groupes d'approches est que dans le premier, les méthodes s'appuient sur le respect du plan de fabrication prévu, par contre dans le deuxième, elles ignorent ce plan et génèrent un autre. Dans certains cas, où les travaux commencés ne peuvent pas être interrompus (comme il est le cas dans notre mémoire), cette ignorance peut rendre le problème plus difficile à résoudre, à cause de la quantité énorme d'informations qui doit être prisent en compte. La méthode d'insertion de tâches est l'une des méthodes, basant sur le contexte statique. La majorité des travaux, qui ont utilisés cette méthode, considèrent que les tâches à insérer sont des tâches indépendantes (même si elles constituent le même travail), c'est-à-dire que pour eux, la procédure d'insertion s'effectue d'une manière séparée (une par une). Cette propriété d'indépendance entre tâches représente la différence principale entre les problèmes traités par ces travaux et le problème présent (traité dans ce mémoire). Il s'agit du problème de type Job Shop où les tâches d'un même job sont toujours liées. Pour délimiter les contours de notre étude et d'identifier clairement le problème, nous sommes passé à la définition du système de production. Le chapitre suivant conduit à ce but.

## CHAPITRE II

# Description Du Système De Production

## II.1 Introduction

La résolution des problèmes d'ordonnancement pour n'importe quel système de production nécessite une bonne maîtrise de ce dernier, c'est ce que représente l'objectif principal de ce chapitre. Nous présentons dans ce qui suit une généralité sur les systèmes de production et leurs différentes caractéristiques. La gestion de ces systèmes est ensuite exposée d'une façon rapide, dans laquelle on détermine la position et le rôle de la fonction ordonnancement. Un petit rappel sur la notion « flexibilité » dans les systèmes de production a été donné. En fin, une description détaillée du système cible a été effectuée pour bien définir le problème.

## II.2 Systèmes de production

### II.2.1 Définition

« Le système de production regroupe l'ensemble des ressources qui conduisent à la création de biens ou de services » [24].

Un système de production est un ensemble de ressources réalisant une activité de production. La production représente la transformation qui s'effectue par une succession d'opérations utilisant des ressources (machines et opérateurs) et modifiant les matières premières ou composants entrant dans le système de production afin de créer les produits finis sortant de ce système et destinés à être consommés par des clients. Les modifications peuvent porter sur la forme du produit, sa structure, son apparence, etc. Les ressources appartenant au système de production mobilisées pour réaliser l'activité de production peuvent être des machines, des opérateurs, de l'énergie, des informations, des outillages... [15].

### II.2.2 Classification des systèmes

Deux classifications des systèmes de production sont souvent utilisées dans la littérature. La première classification repose sur la nature et le volume des flux physiques dans le système, tandis que la deuxième tient en compte le mode de pilotage [25].

#### II.2.2.1 Classification selon la nature et le volume des flux physiques

Généralement les systèmes de production se différencient, selon la manière de fabrication des produits, en trois classes :

- **Les systèmes travaillant en continu ou en série** : La caractéristique principale de ces systèmes est la circulation des matières en flux continu. Les ressources sont organisées sous forme d'une chaîne où les matières premières doivent passer sur toutes ces ressources.

Ce type de systèmes concerne surtout les industries dites de « process » dont la production nécessite la manipulation de matières liquides ou gazeuses [16].

➤ **Les systèmes à flux discret :** Ce modèle de systèmes représente le caractère essentiel des entreprises manufacturières. Les produits peuvent être distingués individuellement et les stocks temporaires entre les postes de travail sont souvent utilisés. Les systèmes à flux discret sont devisés en trois classes [26] :

- *les systèmes de production en grande série ou de masse :* la fabrication dans ce type de système est consacré à la réalisation d'une grande quantité de produits identiques, dont le passage des ces produit est toujours le même et dépend de leurs types.
- *les systèmes de production en moyenne série ou par lots :* contrairement à la classe précédente, il s'agit ici d'ateliers dans lesquels la diversité des produits nécessite une variété de moyens de production. Les différents produits suivent leur propre chemin sur des ressources communes et qui sont souvent regroupées par fonctionnalités équivalentes.
- *les systèmes de production unitaire :* Ce type de système se caractérise par la faible quantité des produits à fabriquer. Réunir les moyens nécessaires au bon moment et au bon endroit représente la tâche principale de la production pour ce type de système.

➤ **Les systèmes à flux hybride ou discontinu:** Ces systèmes se situent entre les deux types de systèmes précédents. Deux configurations peuvent être distinguées :

- *les deux types de systèmes (continu et discret) sont couplés :* la production est continue tout en ayant un conditionnement discret des produits. On trouve par exemple les systèmes de production de la semoule.
- *les deux aspects continu et discret cohabitent :* Dans le même système de production, les traitements sont continus mais effectués par lots. Exemple : la production des boissons.

#### II.2.2.2 Classification selon le mode de pilotage

Dans cette classification les systèmes de production sont classés, selon le type de gestion et le choix de stratégie de fabrication, en deux classes :

➤ **Les systèmes de production à la commande ;** La production pour ces systèmes est déclenchée par l'arrivée d'une certaine commande. Ce type représente généralement

les entreprises fabriquant une variété de produits dont la demande est aléatoire ou celles qui ne définissent pas leurs produits qu'à partir d'une demande précise de client [16].

- **Les systèmes à flux poussés** ; dans ce cadre, le déclenchement de la production est basé sur des planifications et des prévisions pour déterminer un programme de production. La disponibilité du produit venant de l'amont est le responsable de déclenchement de l'étape suivante de fabrication. Cette méthode de production implique souvent le stockage des produits finis avant leur livraison [26].

### II.2.3 Décomposition et organisation du système de production

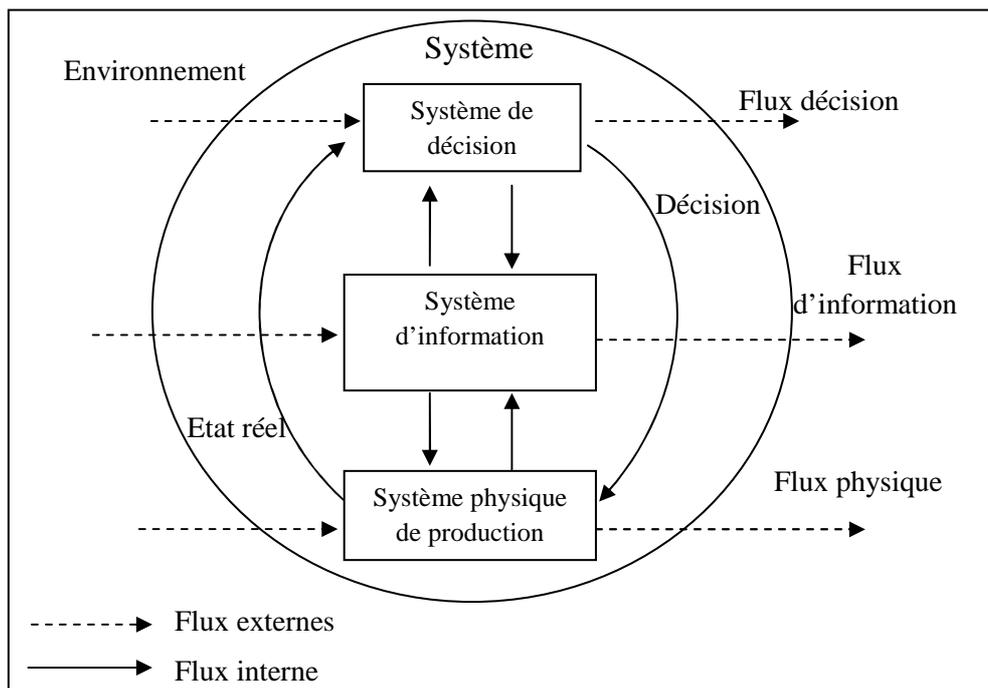
Dans la littérature, les systèmes de production sont souvent décomposés en trois sous-systèmes, Le système de décision, le système d'information et le système physique [2, 15].

**\*Le système physique de production :** C'est le responsable de la procédure de transformation. Il consiste, par ses éléments (ressources humaines et physiques), à convertir des matières premières ou composantes en produits finis.

**\*Le système d'information :** La fonction principale de ce sous système est la collection, le stockage et le traitement des informations. Il offre un lien entre le système de décision et le système physique de production, comme il peut se trouver à l'intérieur même de ces deux systèmes. Il maintient par exemple la gestion des informations utilisées lors de prises de décision dans le système décisionnel, et la création et le stockage d'informations de suivi dans le système physique.

**\*Le système de décision :** L'objectif de ce système est de contrôler le système physique de production. Il combine et organise les activités en prenant des décisions basées sur les données transmises par le système d'information. Ce système est aussi divisé en trois niveaux ; niveau stratégique concernant les décisions à long terme, niveau tactique où les décisions sont prises au moyen terme et niveau opérationnel qui est limité sur le court terme [27, 28].

La Figure suivante décrit les relations existantes entre les trois composantes du système et le type de flux circulant dans ce dernier.



**Figure II.1** : Décomposition classique d'un système de production [29]

## II.2.4 La gestion de systèmes de production

### II.2.4.1 Définition et rôle de la gestion de production

La **gestion de production** a pour objet la recherche d'une organisation efficace dans l'espace et dans le temps de toutes les activités relatives à la production afin d'atteindre les objectifs de l'entreprise [28]. La gestion de production est une fonction complexe, elle s'occupe d'un ensemble de problèmes liés à la production tels que la gestion des données, la planification, le contrôle (suivi) de production, la gestion des stocks, la prévision, l'ordonnancement etc... [2].

### II.2.4.2 Organisation hiérarchique de la gestion de production

Généralement, la fonction gestion de production est répartie en trois niveaux hiérarchiques: stratégique, tactique et opérationnel [28, 30].

- **Niveau stratégique** : Contient toutes les décisions de caractère 'long terme'. Ces décisions consistent à déterminer la politique de l'entreprise et conditionner son avenir. Elles portent essentiellement sur la gestion des ressources durables, afin que celles-ci soient toujours suffisantes pour assurer la pérennité de l'entreprise. Les ressources

visées peuvent être des machines, des hommes, des informations ou des données techniques.

- **Niveau tactique** : Dans ce niveau, les décisions sont prises à moyen terme. Leur rôle est d'assurer la liaison entre le niveau stratégique et le niveau opérationnel et de contrôler la bonne adéquation des ressources disponibles et des charges engendrées par les commandes ou les prévisions, mais sans modification profonde de la structure et du fonctionnement de l'entreprise.
- **Niveau opérationnel** : Ce niveau représente la gestion quotidienne de l'entreprise. Il englobe les décisions prises à court terme et à très court terme. Ces décisions assurent le lancement des activités et la flexibilité nécessaire à la bonne conduite de la production. On trouve dans ce niveau par exemple : la gestion des stocks, la gestion de la main d'œuvre et la gestion des équipements [31].

#### **II.3.4.3 Fonction ordonnancement dans la gestion de production**

La gestion de production s'occupe d'un ensemble de problèmes liés à la production tel que la gestion de données, la planification, l'ordonnancement, la gestion de stocks ...

La fonction ordonnancement dans le système de gestion de la production joue un rôle très important, elle consiste à gérer le temps et le rythme de la vie de l'usine.

L'ordonnancement d'atelier couvre un ensemble d'actions qui transforment les décisions de fabrication définies par le programme directeur de production en instructions d'exécution détaillées destinées à piloter et contrôler à court terme l'activité des postes de travail dans l'atelier. Elle peut être décomposée en trois sous-fonctions [24]:

- **Élaboration des ordres de fabrication** : cette tâche consiste à transformer les informations du programme directeur de production (suggestions de fabrication) en ordres de fabrication;
- **Élaboration du planning d'atelier** : cette tâche consiste, en fonction de ces ordres de fabrication et de la disponibilité des ressources consommables (matières premières, composants) et partageables (postes de travail), à déterminer le calendrier prévisionnel de fabrication (cela revient à transformer les prévisions de fabrication à court terme en ordres d'exécution à très court terme);
- **Lancement et suivi** : cette tâche consiste à *distribuer* aux postes de travail les documents nécessaires à la bonne exécution des fabrications (lancement en fabrication) et *suivre* l'exécution des fabrications (suivi de production).

## II.3 Flexibilité dans les systèmes de production

### II.3.1 Définitions de La flexibilité

Le terme flexibilité est très utilisé, notamment en gestion de production, où la recherche de flexibilité apparaît comme un des éléments les plus marquants dans l'évolution des systèmes de production [32].

Selon [33] la flexibilité d'un objet ou d'un système, est une propriété qui recouvre généralement deux aspects complémentaires mais distincts: Un aspect interne lié à une capacité de changement et de déformation à une variété d'états possibles et un aspect externe lié à une capacité d'adaptation à des modifications de l'environnement.

Nagarur [34] a défini la flexibilité des systèmes de production par la capacité d'un système à s'adapter rapidement à des changements de facteurs clefs tels que le produit, le process, la charge et les pannes des machines.

Pour le domaine de l'ordonnancement, GOTHa [35] indiquent que la flexibilité peut être exprimée comme l'existence de modifications possibles dans un ordonnancement, calculé hors-ligne, entraînant une perte de performance acceptable. Elle peut aussi être associée à un voisinage de solutions pouvant être examiné lors de l'exécution, ou à une famille d'ordonnements, sans privilégier un ordonnancement en particulier [35].

### II.3.2 Dimensions de flexibilité

Nous nous limitons dans cette partie, à la flexibilité dans les systèmes manufacturiers et plus précisément le sous-système de production (machines, système de manutention, stockage) et le sous-système de gestion de la production (investissements, planification, ordonnancement, approvisionnement).

La classification exposée ici est une classification développée par [36]. Elle différencie la flexibilité dans le système de production en sept dimensions. Une explication plus détaillée de cette classification est trouvée dans [37].

**Flexibilité process :** Capacité de faire varier les tâches nécessaires à la réalisation d'un travail. Ceci permet de réaliser plusieurs travaux différents dans le système, en utilisant une variété de machines.

**Flexibilité sur le routage :** Capacité de changer la séquence de passage des pièces sur les machines et de continuer à produire un ensemble donné de pièces même lorsqu'une machine

tombe en panne. (Existe seulement s'il y a plusieurs séquences de production possible ou lorsque chaque opération peut être réalisée sur plus d'une machine).

**Flexibilité sur les opérations :** Capacité de changer l'ordre de certaines opérations dans une gamme de production.

**Flexibilité machine :** Capacité de changer d'outils et d'assembler ou monter les fixations nécessaires sans interférer avec la production et sans temps de reconfiguration long. C'est la facilité du système à faire les changements nécessaires sur les machines pour produire un ensemble donné de pièces.

**Flexibilité produit :** Capacité de mettre en œuvre, rapidement et de manière économique, les changements nécessaires à l'intégration de nouvelles pièces dans le plan actuel.

**Flexibilité sur le volume :** Capacité de faire fonctionner un atelier à des niveaux de production différents tout en restant à un niveau de profit acceptable.

**Flexibilité sur la production** Capacité de faire varier rapidement et de manière économique la gamme de pièces qu'un atelier peut produire. Cette flexibilité ne peut être atteinte que lorsque les autres dimensions sont satisfaites.

**Flexibilité sur l'expansion :** Capacité de construire un système ou de l'agrandir facilement et de manière séparable.

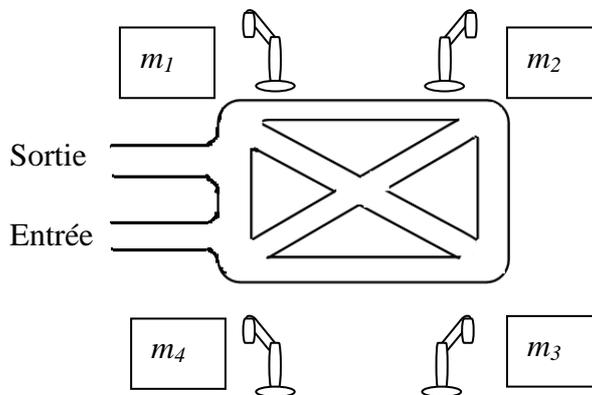
## II.4 Système à étudier et hypothèses proposés

### II.4.1 Organisation du système et modèle de transport

Le mode de fonctionnement de ce système appartient à la classe « production au flux discrets », plus précisément le type de job shop. Il est constitué de quatre machines  $m_1$ ,  $m_2$ ,  $m_3$ , et  $m_4$  positionnées sous forme d'un carré (Figure II.2). Les produits sont transportés à l'aide des chariots automatiques, qui peuvent contenir une ou plusieurs pièces (lot) du même produit (pour chaque chariot). Le chariot peut passer devant les machines et dans toutes les directions et cela suivant l'ordonnancement prédéfini et la gamme opératoire du produit qu'il contient. Un bras manipulateur est installé à côté de chaque machine. Il sert à déplacer la/les pièces du chariot vers les machines et l'inverse.

On considère que le système n'autorise pas les stocks intermédiaires. Alors la conséquence immédiate est qu'une machine n'est pas disponible dès la terminaison d'une tâche. Celle-ci doit attendre que la machine suivante dans la gamme opératoire soit libérée pour lui transférer

la pièce. De cette contrainte, on a inspiré la propriété « succession sans interruption » qui impose toujours la non séparation entre tâches.



**Figure II.2:** Système de production à quatre machines

#### II.4.1.1 Les machines

Les machines représentent l'élément fondamental du système, ce sont des équipements matériels réalisant des tâches spécifiques. Elles appartiennent au groupe des ressources disjonctives, c'est-à-dire elles ne peuvent effectuer qu'une seule tâche à la fois. On considère que la disponibilité de ces machines est toujours maintenue (pas de pannes toute au long de la phase d'exécution). Alors deux états caractérisent le fonctionnement des machines :

- l'état occupé ; la ressource est entraînée de réaliser une certaine tâche (état de marche) ;
- et l'état non occupé ou libre ; la ressource n'exécute aucune tâche (état de repos).

#### II.4.1.2 Les commandes

On désigne par le terme commande ici, le travail (job) à réaliser sur une matière première. Il englobe l'ensemble des tâches nécessaires à la fabrication de certain produit.

La réalisation de chaque commande est caractérisée par une date souhaitée (délai de livraison) qu'on veut respecter. Il est possible de dépasser ce délai dans le cas où la commande est non prioritaire. Deux types de commandes sont alors tenus en compte :

- des commandes déjà planifiées selon un ordonnancement prévisionnel, qui se répètent dans un cycle bien déterminé ;
- et des commandes urgentes, se caractérisent par le fait que leur arrivée est aléatoire.

### II.4.1.3 Les tâches

Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début ou de fin, dont la réalisation est caractérisée par une durée défini. Un ensemble de propriétés sur les tâches sont à souligner :

- \*) La durée de chacune de ces tâches est déterminée par l'union des temps suivant :
  - le temps de transport, c'est le temps de déplacement de la/les pièces vers la machine concernée ;
  - le temps d'exécution, qui représente le temps dans lequel une pièce doit rester dans la machine.
  - le temps de préparation des ressources.
- On note que s'il s'agit d'un lot, ce temps est calculé par la somme des temps de toutes les pièces de ce lot.
- \*) La notion de préemption des tâche est non permit, c'est-à-dire que les tâches n'acceptent aucune interruption une fois sont commencées.
- \*) La propriété succession sans interruption impose de ne pas avoir une séparation entre les tâches.

Dans certaine cas, il est possible d'étendre la durée d'exécution d'une tâche, la ressource concernée n'est pas donc disponible pour exécuter une autre tâche pendant cette durée.

### II.4.2 Formulation du problème et notations utilisée

Supposant que notre système est entrain de réaliser un ensemble  $J = \{1...n\}$ , de  $n$  job (commande). Chaque job  $j$  est constitué de  $n_j$  tâches.  $O_{ij}^{uk}$  représente la  $i^{\text{ème}}$  tâche du job  $j$  qui doit être exécutée sans interruption sur une unique ressource  $k \in M$  dans l'ordre\*  $u$ . Chaque tâche  $O_{ij}^{uk}$  est caractérisée par une date de début  $t_{ij}^{uk}$ , date de fin  $c_{ij}^{uk}$  et une durée d'exécution  $p_{ij}^{uk}$ .

La réalisation de ces tâches s'effectue selon un ordonnancement prévisionnel « hors ligne » caractérisé par :

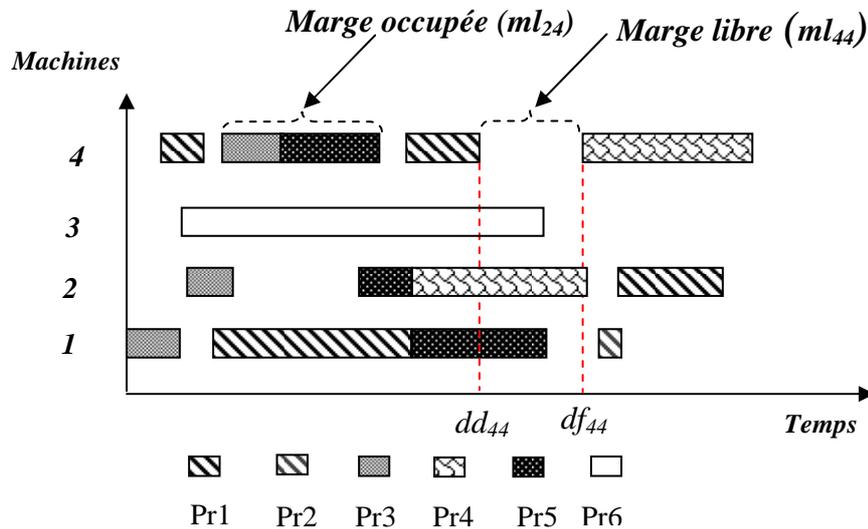
- des marges occupées  $MO_k$ , sont des intervalles de temps où la machine  $k$  est entrain d'exécuter une ou plusieurs tâches successives ;
- des marges libres  $ML_k$  ce sont des intervalles de temps où la machine  $k$  est en état de

---

\* l'ordre  $u$  n'est défini qu'après la définition du plan prévisionnelle

repos (des temps morts).  $ml_{uk}$  : La marge libre numéro  $u$  de la machine  $k$ , elle est définie par une date de début  $dd_{uk}$  et date de fin  $df_{uk}$  et une durée  $d_{uk}$ .

La notion des marges est bien expliquée par la (Figure II.3) qui représente un exemple d'un ordonnancement prévisionnel.



**Figure II.3** : Diagramme de Gantt d'un ordonnancement prévisionnel

À un instant donné  $T^a$  de l'exécution de cet ordonnancement, une nouvelle commande urgente représentée par le job  $j^a$  de  $n^a$  tâches apparaît.

Le problème imposé est quelle est la bonne méthode à suivre pour que notre système puisse s'adapter face à l'occurrence de cette nouvelle commande et comment doit-il réagir dans le cas où la commande urgente est prioritaire ?

## II.5 Conclusion

Dans ce chapitre on a donné une présentation générale sur les systèmes de production, leurs classifications et leurs manières de gestion. En suite on s'est intéressé au système de production concerné par ce travail où on a identifié sa position par rapport aux autres systèmes, ainsi que ses caractéristiques principales qui définissent son fonctionnement. La caractéristique qui apparaît la plus importante par laquelle ce système se diffère aux autres systèmes est l'absence des stocks intermédiaires. De cette caractéristique où on a inspiré la propriété succession sans interruption qui impose de ne pas avoir une séparation entre les tâches. Un ensemble des hypothèses sont aussi proposées, leur rôle est de simplifier le travail et de définir clairement son cadre générale. En fin nous avons donné une explication du problème d'ordonnancement à résoudre dans ce mémoire. La résolution de ce problème et l'approche utilisée seront abordées dans le chapitre suivant.

## CHAPITRE III

# Méthode D'insertion Des Nouvelles Commandes

### **III.1. Introduction**

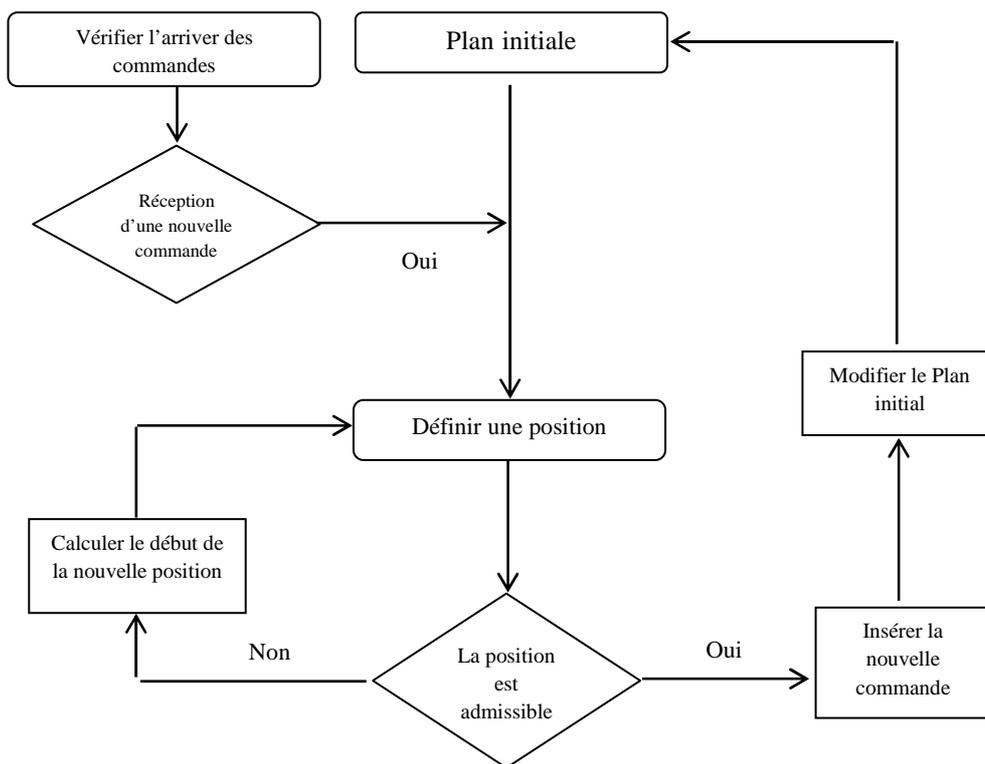
Pour la résolution du problème imposée, nous avons proposé une méthode approchée qui consiste à générer un ordonnancement en temps réel et qui donne la possibilité d'introduire la commande urgente dans le plan prévisionnel. L'objectif de cette méthode est de fournir au système considéré une capacité non seulement d'absorber une commande supplémentaire, mais aussi de la traiter le plus vite possible pour avoir la possibilité d'absorber une autre commande qui peut apparaître dans le futur.

Dans cette direction, nous avons devisée le travail en deux parties selon deux cas possible. La première partie (sans délai de livraison) consiste à résoudre le problème dans le cas où la commande urgente est non prioritaire. On cherche ici d'insérer cette commande dans le plan prévisionnel sans tenir compte du délai de livraison de celle ci. La deuxième partie (avec délai de livraison) donne une solution dans le cas où la commande urgente ne doit être pas dépassée son délai de livraison. Cette solution est proposée avec possibilité de modifier le plan prévisionnel si c'est nécessaire. Un exemple d'application est effectué à la fin de ce chapitre pour évaluer les performances de la méthode proposé.

### **III.2. Première partie : cas sans délai de livraison**

L'approche proposée dans cette partie est une approche itérative. Elle consiste à chercher, dans le plan prévisionnel, une position admissible dans laquelle on peut insérer les nouvelles tâches. Cette recherche est effectuée par la génération d'un ensemble des itérations successives, définissant dans chaque itération une position possible. La position définie est ensuite soumis à un teste d'admissibilité pour évaluer sa capacité d'absorber la nouvelle commande. La génération des itérations sera automatiquement arrêtée dès qu'une position admissible est trouvée. L'insertion de la nouvelle commande s'effectue donc dans cette position selon la propriété succession sans interruption. Nous obtenons par conséquent un nouveau plan.

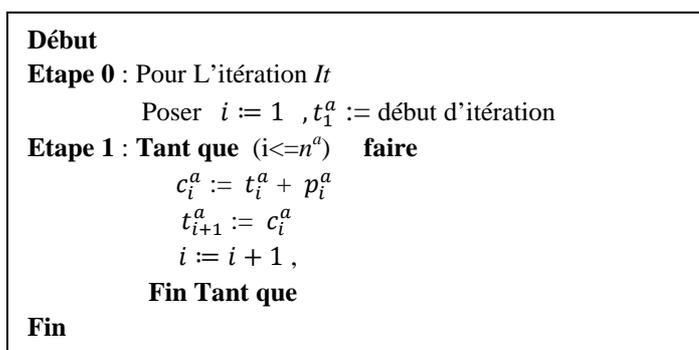
L'organigramme qui résume les différentes étapes de la démarche suivi dans la première partie est représenté par la Figure III.1



**Figure III.1 :** Organigramme de l'approche sans délai de livraison

### III.2.1 Définition d'une position

La définition d'une position est réalisée par l'opération de positionnement de l'ensemble des tâches sur le plan prévisionnel. Ce positionnement se fait selon la propriété succession sans interruption c'est-à-dire toujours garder la non séparation entre tâches. Commencant alors par le début de l'itération, les tâches sont placées de tel sort que la fin d'une tâche est obligatoirement le début de la tâche suivante, on parle donc d'une position du bloc de tâches au lieu de chaque tâche séparément. En effet pour dire qu'une telle position de la nouvelle commande est admissible il faut que toutes les positions des tâches soient des positions admissibles. **L'Algorithme III.1** décrit le positionnement des tâches



**Algorithme III.1 :** Définition d'une position

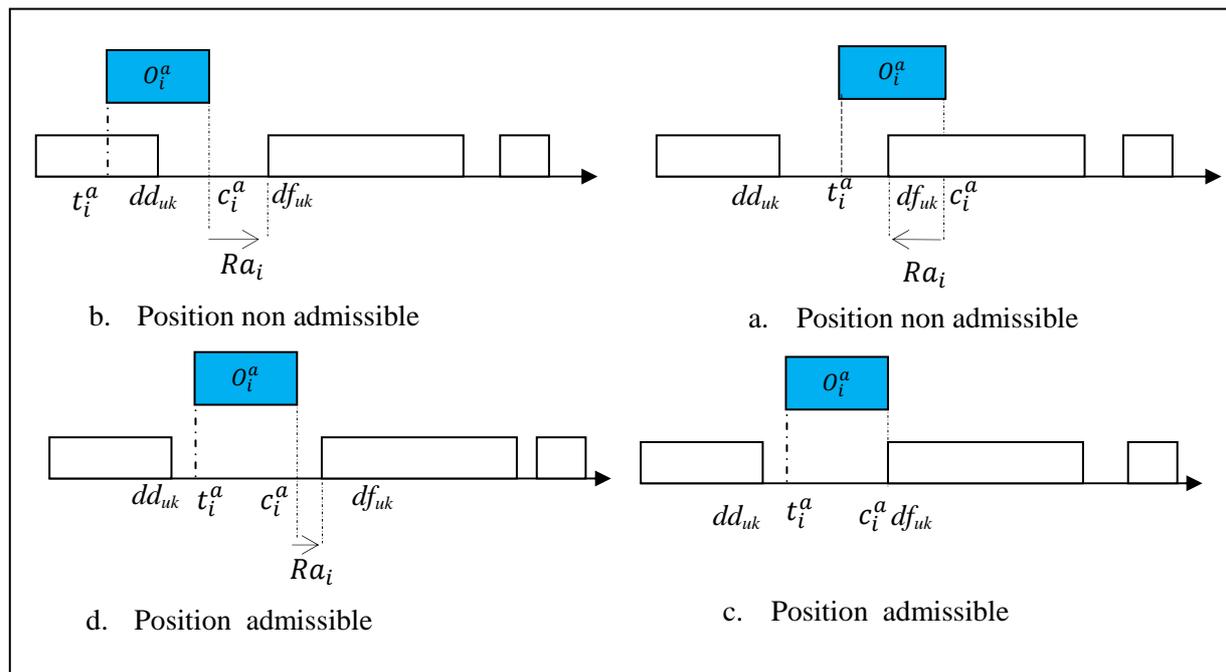
**III.2.2 Notion d'admissibilité**

L'admissibilité de la position d'une tâche  $O_i^a$  s'effectue selon un teste d'admissibilité qui nous permet de vérifier la possibilité d'insérer  $O_i^a$  dans cette position, ce teste repose sur deux conditions essentielles :

- La première condition impose que la date début de la tâche  $O_i^a$  doit être située dans une marge libre de la machine  $m_i$  où elle doit être exécutée, c'est-à-dire, il existe une marge  $ml_{uk}$  où  $t_i^a$  appartient à l'intervalle  $[dd_{uk} \ df_{uk}[$  ( $k=m_i$ ).
- La deuxième condition exige que le rapport d'admissibilité «  $Ra_i$  » doit être positif ou égale à zéro.  $Ra_i$  représente l'intervalle du temps entre la date fin de la tâche  $O_i^a$  et celle de la marge libre  $ml_{uk}$ ,

$$Ra_i = df_{uk} - c_i^a \tag{III.1}$$

Alors pour une telle tâche  $O_i^a$  quatre états de positionnement sont possibles. Ils sont représentés d'une manière explicite dans la Figure III.2.



**Figure III.2 :** Les différents cas possible d'une position de la tâche  $O_i^a$

Dans une itération, le test d'admissibilité doit être fait pour toutes les tâches de la commande urgente. Un algorithme (Algorithme III.2) est proposé dans ce cas, il permet de vérifier l'admissibilité de ces positions.

On note :

- ✓  $Nd_{It} = \{ O_i^a \in J_a / O_i^a \text{ n'a pas une position admissible} \}$  l'ensemble des tâche qui n'ont pas des positions admissible dans l'itération  $It$ .
- ✓  $Ra_i^{It}$  le rapport d'admissibilité de la tâche  $O_i^a$  dans l'itération  $It$ .

**Début**

**Etape 0 :** Initialisation  
 Pour l'itération  $It$   
 Poser  $i := 1$  ,  $t_1^a :=$  début d'itération,  $Nd_{It} = \emptyset$

**Etape 1 :** **Tant que** ( $i \leq n^a$ ) **faire**  
      $k = m_i$  la machine nécessaire pour réaliser  $O_i^a$   
     **Si**  $\exists ML_{uk}$  tel que  $t_i^a \in [dd_{uk}, df_{uk}]$  **alors**  
         Aller à l'étape 2  
     **Si non** position non admissible  
          $Nd_{It} = Nd_{It} \cup O_i^a$   
          $i := i + 1$  ,  $t_{i+1}^a := t_i^a + p_i^a$   
     **Fin si**

**Fin Tant que**

**Etape 2 :** Calculer  $Ra_i^{It} = df_{ik} - c_i^a$   
     **Si**  $Ra_i^{It} \geq 0$  **alors**  
         Position admissible  
     **Si non** Position non admissible  
          $Nd_{It} = Nd_{It} \cup O_i^a$   
     **Fin si**  
          $i := i + 1$  ,  $t_{i+1}^a := t_i^a + p_i^a$  aller à l'étape 1

**Fin**

### Algorithme III.2 : Test d'admissibilité des positions

Après cette étape on peut dire que la détection d'une position admissible implique la fin de la recherche et la position détectée représente la solution recherchée. Dans le cas où la position est non admissible ; on doit passer à une autre itération et recommencer le teste. Le passage d'une itération à une autre consiste à définir le point de départ de l'itération suivante qui coïncide toujours avec le début de la nouvelle position  $t_1^a$ .

### III.2.3 Détermination des dates départs des positions

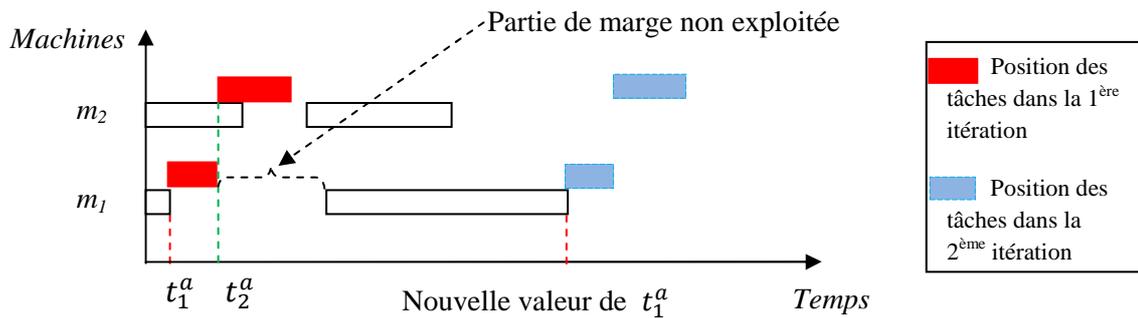
Pour trouver la date de début de la nouvelle position on est devant trois propositions possibles, nous nous intéressons par la troisième qui représente notre proposition.

- **La première proposition :** consiste à décaler la valeur de  $t_1^a$  par unité temps pour chaque itération. L'objectif est d'essayer de parcourir le maximum du plan. On remarque que cette

proposition est non optimale, car ce décalage d'une unité de temps génère un agrandissement du temps de calcul et peut être des itérations non utiles.

- **La deuxième proposition :** a comme idée de donner à  $t_1^a$  directement la valeur de date début de la plus proche marge libre  $dd_k(\text{secc } t_1^a)$ . Le but de cette proposition est de faire éliminer les itérations non utiles où le  $t_1^a$  située dans une marge occupée.

Cette proposition, malgré qu'elle assure une minimisation du temps de calcul par l'élimination des itérations non utiles, mais elle peut faire perdre une partie de marge libre non exploité comme le montre la Figure III.3.



**Figure III.3 :** Calcul de la nouvelle valeur de  $t_1^a$  « cas de la deuxième proposition »

- **La troisième proposition**

Comme il est montré avant, l'objectif est de faire exploiter le maximum possible de marges libres existantes dans le plan et en même temps de minimiser le temps de calcul. Pour cela, nous proposons que, pour calculer la nouvelle date début de l'itération  $t_1^a$  on a besoin :

- ✓ de calculer pour chaque tâche de la commande urgente la différence entre sa date début et le début de la plus proche marge libre de la même machine. Cette différence est traduite par la relation suivante :

$$Rp_i = dd_{uk}(\text{secct}_i^a) - t_i^a \tag{III.2}$$

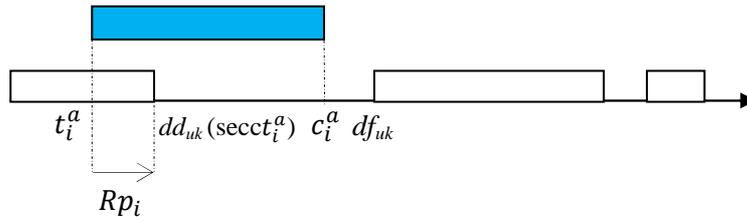
On désigne par  $Rp_i$  le rapport de passage qui représente cette différence et  $dd_{uk}(\text{secct}_i^a)$  le début de la première marge libre après la date  $t_i^a$ . Ces notions sont expliquées dans la Figure III.4.

- ✓ de sélectionner parmi les valeurs de  $Rp_i$  celle qui est la plus grande.

$$Rp_{max} = \max_{i=1...n_a} (Rp_i) \tag{III.3}$$

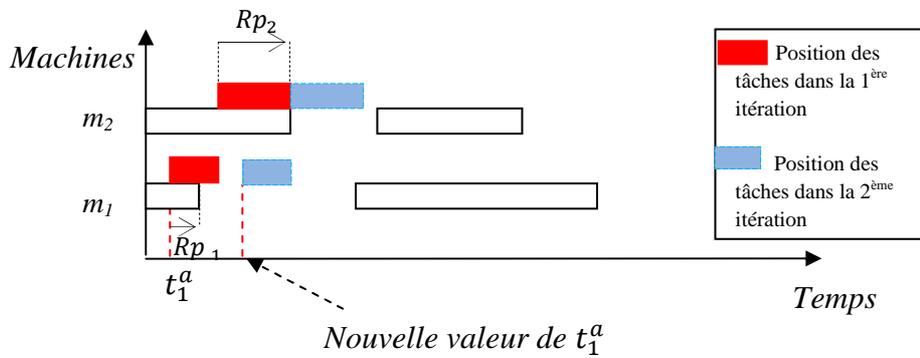
Donc La nouvelle valeur de  $t_1^a$  sera calculer par la relation suivante :

$$t_1^a = t_1^a + Rp_{max} \tag{III.4}$$



**Figure III.4 :** Exemple d'une position de la tâche  $O_i^a$

Le choix de  $Rp_{max}$  est basé sur le fait que dans cette intervalle du temps  $[t_i^a, Rp_i^t + t_i^a [$ , il apparait bien claire qu'il existe au moins une tâche qui n'a pas une position admissible, c'est la tâche qui a le  $Rp_{max}$ . Le but de ce choix est d'éliminer un ensemble des itérations non utiles qui peuvent se générer dans le cas où on choisit d'autre valeur du rapport  $Rp$ . Un exemple de définition de la valeur  $t_1^a$  par cette proposition est montré dans la Figure III.5



**Figure III.5 :** Calcul de la nouvelle valeur de  $t_1^a$  « cas de la troisième proposition »

**Preuve mathématique du choix de  $Rp_{max}$**

On considère par exemple que la commande urgente contient trois tâches ( $O_1^a, O_2^a$  et  $O_3^a$ ) et  $O_3^a$  est la tâche qui a le  $Rp_{max}$

Si on choisit pour calculer  $t_1^a$  une autre valeur inférieure à  $Rp_{max}$ , soit par exemple  $Rp_w$  on trouve :

Pour la tâche  $O_3^a$  sa nouvelle valeur de début  $t_3^a$  sera :

$$t_3^a = t_3^a + Rp_w \tag{III.5}$$

D'où  $Rp_w < Rp_{max}$  et  $t_3^a > 0$  III.6

on peut faire  $t_3^a + Rp_w < t_3^a + Rp_{max}$  III.7

et encours  $t_3^a + Rp_w < dd_{uk}(secct_3^a)$  III.8

alors  $t_3^a < dd_{uk}(secct_3^a) < df_{uk}(secct_3^a)$  III.9

À partir de la relation (III.9) on peut déduire que  $O_3^a$  n'aura aucune position admissible avant la date  $dd_{uk}(\text{secct}_3^a)$  parce que  $t_3^a$  n'appartient pas à aucune marge libre dans l'intervalle  $[t_3^a, t_3^a + Rp_w]$ .

### III.2.4 L'insertion de la nouvelle commande

L'objectif dans cette partie est d'apporter au système une capacité de traiter rapidement la nouvelle commande. La solution du problème d'ordonnancement alors se base sur le choix de la première position admissible trouvée. La nouvelle commande doit être immédiatement insérer dans cette position et le nouveau plan sera établi sur la base de l'ancien plan et les tâches insérées dans celui ci.

Les étapes de la démarche pour la première partie sont traduites par l'algorithme général suivant :

```

Début
Etape 0 : Initialisation
    Commencer par la date d'apparition  $t_1^a := Ta$ ,
    Poser  $i := 1$ ,  $Nd_H = \emptyset$ ,  $It = 1$ 
Etape 1 : Définition de la position
    Si ( $i \leq n^a$ ) Alors
         $t_i^a = t_{i-1}^a + p_{i-1}^a$  pour  $i = 2 \dots n$ 
         $c_i^a := t_i^a + p_i^a$ 
        Aller à l'étape 2
    Sinon
        Aller à l'étape 3
    Fin
Etape 2 : Teste d'admissibilité
    Si  $\exists ML_{uk}$  tel que  $t_i^a \in [dd_{uk}, df_{uk}]$  Alors
        Calculer  $Ra_i^{It} = df_{uk} - c_i^a$ 
         $Rp_i^{It} = dd_k(\text{secc } t_i^a) - t_i^a$ 
        Si  $Ra_i^{It} \geq 0$  Alors
            Position de  $O_i^a$  est admissible
            Passer à la tâche suivante  $i = i+1$ 
            Aller à l'étape 1
        Sinon
            La position est non admissible
             $Nd_H = Nd_H \cup O_i^a$ 
            Passer à la tâche suivante  $i = i+1$ 
            Aller à l'étape 1
        Fin si
    Fin si
Etape 3 : La décision
    Si  $Nd_H = \emptyset$  (toutes les tâches ont des positions admissibles) Alors
        Insérer la nouvelle commande
        Changer le plan initial
    Sinon
        Put  $t_1^a := t_1^a + Ra_{max}^{It}$ 
        Passer à l'itération suivante  $It = It + 1$ 
        Aller à l'étape 1
    Fin si
Fin

```

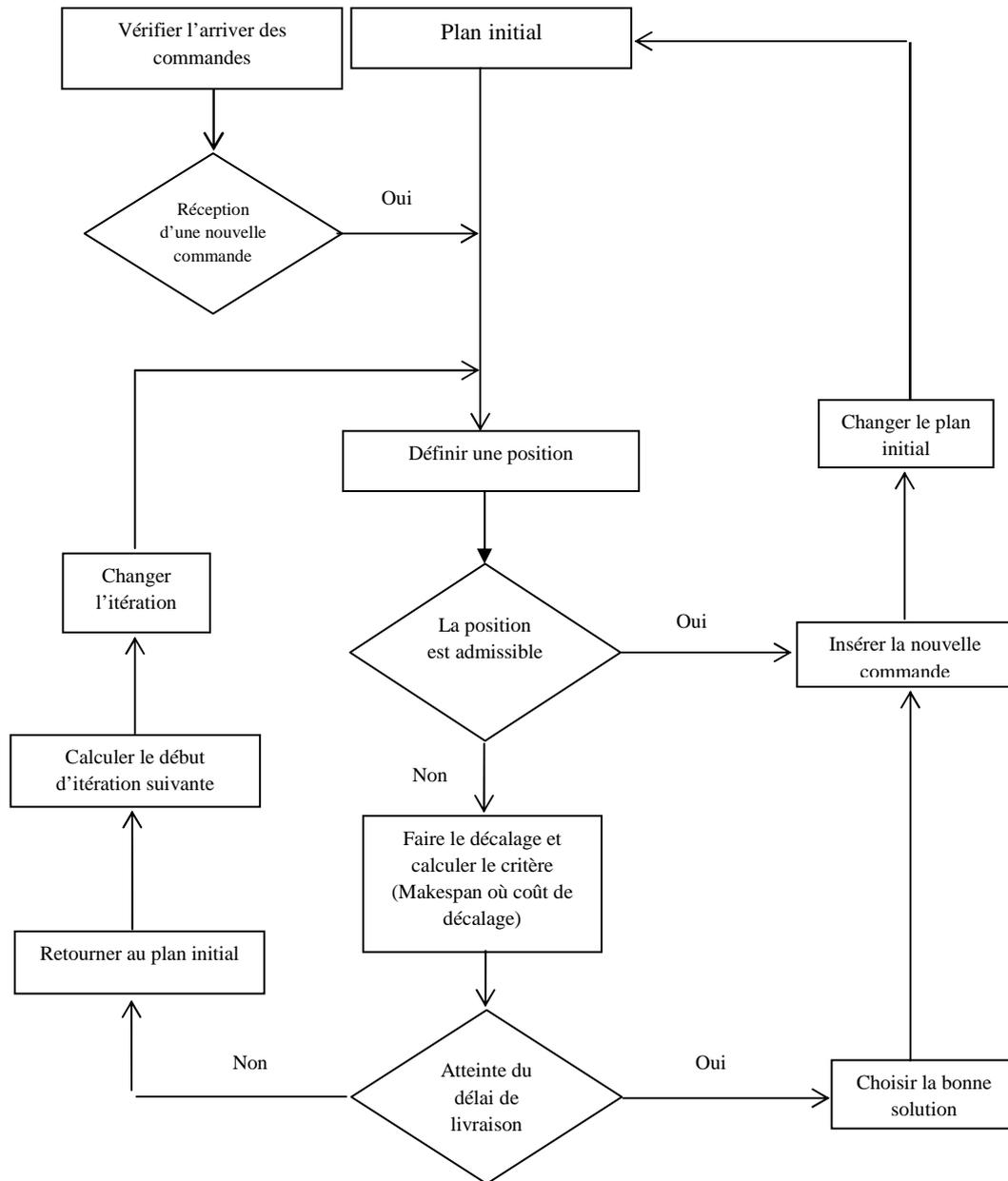
**Algorithme III.3** : Insertion des nouvelles tâches dans l'ordonnancement prévisionnel

Dans le premier cas, le délai de livraison n'était pas exigé (la commande est non prioritaire), donc on a eu la possibilité d'insérer la nouvelle commande sans perturber l'ordre de passage existant, c'est-à-dire au pire des cas l'insertion se fait à la fin du plan prévisionnel. Par contre, si les commandes arrivées sont des commandes prioritaires, dans ce cas, il est nécessaire d'apporter certaines modifications au plan prédéfini dans le cas d'absence d'une position admissible. La partie suivante a été consacrée pour ce type de problème.

### III.3 Approche qui prend en compte le délai de livraison

Dans cette partie la procédure est aussi basée dans son déroulement sur la recherche d'une position d'insertion admissible par la génération d'un ensemble des itérations. La spécificité de cette approche est que, pour chaque itération et dans le cas où la position définie n'est pas admissible, on doit effectuer une procédure de décalage qui sert à modifier l'ordre de passage existant, de tel sort que la position soit admissible. L'opération de la recherche est recommencée après une remise du plan à l'état initial. Un ensemble de solutions est donc généré à la fin de la recherche qui doit être arrêtée, soit par la détection d'une position admissible, soit par l'arriver au délai de livraison. Le but visé est de définir parmi ses solutions la bonne solution qui conduit à une modification minimale du plan initial. Deux critères sont utilisés pour sélectionner cette solution, la durée totale d'exécution  $C_{max}$  (Makespan) et les coûts supplémentaire générés par l'opération de décalage.

L'organigramme de la Figure III.6 résume les étapes de cette procédure.



**Figure III.6 :** Organigramme de la deuxième procédure

### III.3.1 Définition d'une position admissible

La détermination d'une position admissible dans cette partie reste la même que pour le premier cas. Elle consiste à positionner les tâches de la commande urgente sur le plan prévisionnel par la détermination des dates début et de fin de chaque tâche, ensuite de faire subir cette position à un teste d'admissibilité pour tester sa possibilité d'accepter les nouvelles tâches. Alors, l'obtention d'une position admissible implique l'arrêt de la recherche et maintenir cette solution. Dans le cas contraire celle ci conduit au passage à la procédure de décalage. La sélection des tâches concernées par le décalage et le calcul du taux de décalage de chaque tâche sont montrés dans le paragraphe suivant.

### III.3.2 Procédure de décalage

La modification du plan prévisionnel se réalise par l'opération de décalage d'une ou plusieurs tâches afin de générer des espaces libres supplémentaires pour qu'une position non admissible devienne admissible. D'une autre manière c'est une quantité du temps additionnée aux dates début et de fin de ces tâches (Figure III.7).

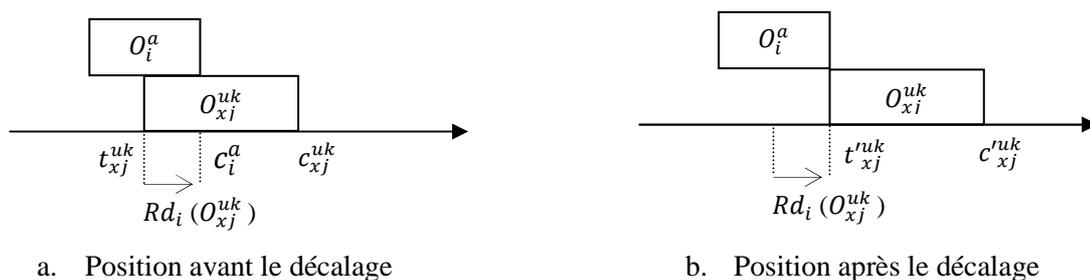
#### III.3.2.1 Déroulement de la procédure

Dans une itération, la procédure de décalage consiste à créer une position admissible pour la première tâche de l'ensemble des tâches de la nouvelle commande qui n'ont pas des positions admissibles. Suivant les résultats générées par cette opération, une mis à jour des données du plan initiale est ensuite procédée. Un teste d'admissibilité sera effectué après cette mis à jours pour déterminer le nouveau ensemble. Ces étapes seront répétées jusqu'à ce que toutes les tâches de la commande aient des positions admissibles.

#### III.3.2.2 Création d'une position admissible et détermination du nouvel ordre

On définit par  $Rd_i(O_{xj}^{uk})$  le temps de décalage (retard) que doit subir la tâche  $O_{xj}^{uk}$  pour permettre l'insertion de la tâche  $O_i^a$ . Ce retard est déterminé par la différence entre la date début de la tâche concernée par le décalage et la date fin de la tâche à insérer (l'équation III.10).

$$Rd_i(O_{xj}^{uk}) = c_i^a - t_{xj}^{uk} \quad \text{III.10}$$



**Figure III.7 :** Exemple d'une opération de décalage effectuée pour insérer la tâche  $O_i^a$

La Figure III.7 explique l'opération de décalage effectuée sur la tâche  $O_{xj}^{uk}$  où  $t_{xj}^{ruk}$  et  $c_{xj}^{ruk}$  représentent les nouvelles données de la tâche  $O_{xj}^{uk}$ , leurs calculs se font par les relations suivantes:

$$t_{xj}^{ruk} = t_{xj}^{uk} + Rd_i(O_{xj}^{uk}) \tag{III.11}$$

et  $c'_{xj}^{uk} = c_{xj}^{uk} + Rd_i(O_{xj}^{uk})$  III.12

L'exemple suivant montre comment elle est réalisée exactement la procédure sur le plan prévisionnel.

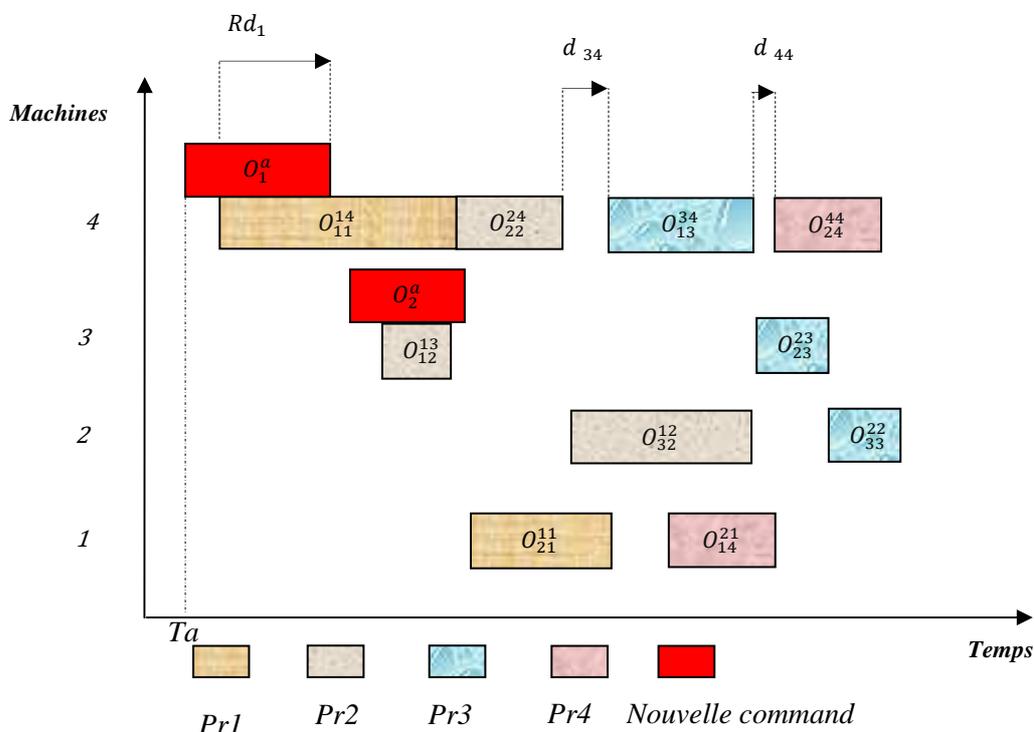
**Exemple** Supposons que notre système est entrainé de réaliser l'ensemble des produits selon un plan prédéfini. Une commande constituée de deux tâches  $O_1^a$  et  $O_2^a$  apparaît dans le temps  $T^a$ .

➤ Premier résultats

Dans une itération  $It$ , une position de la nouvelle commande a été définie comme le représente la (Figure III.8). Cette position est non admissible alors pour insérer la commande on doit décaler certaines tâches. La première tâche concernée par le décalage est la tâche  $O_{11}^{14}$ , elle doit être déplacée dans le temps par :

$$Rd_1(O_{11}^{14}) = Rd_1 \tag{III.13}$$

avec  $Rd_1 = c_1^a - t_{11}^{14}$ .



**Figure III.8 :** Un ordonnancement prévisionnel avec une position pour la nouvelle commande

Le décalage de tâche  $O_{11}^{14}$  génère :

\*) Au niveau de la même machine

1. Un décalage par  $Rd_1$  des tâches de la même marge occupée qui ont des dates débuts supérieur à  $t_1^a$ .

Donc le temps de décalage de la tâche  $O_{22}^{24}$  est  $Rd_1(O_{22}^{24}) = Rd_1$ . On peut dire aussi que

$$Rd_1(O_{22}^{24}) = Rd_1 - d_{24} \quad \text{III.14}$$

$$\text{Parce que } d_{24} = 0$$

2. Un décalage des tâches des marges suivantes

La tâche  $O_{13}^{34}$  sera décalée par une quantité de temps

$$Rd_1(O_{13}^{34}) = (Rd_1 - d_{34})^+ \quad \text{III.15}$$

$$\text{avec } (Rd_1 - d_{34})^+ = \begin{cases} Rd_1 - d_{34} & \text{si } Rd_1 > d_{34} \\ 0 & \text{si } Rd_1 < d_{34} \end{cases} \quad \text{III.16}$$

La tâche  $O_{24}^{44}$  sera décalée par

$$Rd_1(O_{24}^{44}) = [Rd_1 - (d_{34} + d_{44})]^+ \quad \text{III.17}$$

$$\text{avec } [Rd_1 - (d_{34} + d_{44})]^+ = \begin{cases} Rd_1 - (d_{34} + d_{44}) & \text{si } Rd_1 > (d_{34} + d_{44}) \\ 0 & \text{si } Rd_1 < (d_{34} + d_{44}) \end{cases} \quad \text{III.18}$$

\*) Au niveau des produits

Selon la propriété (succession sans interruption) l'ensemble des tâches des produits Pr1 et Pr2 doit être décalées par le temps de décalage  $Rd_1$ . Les tâches du produit Pr3 doivent aussi être décalées mais par  $Rd_1(O_{13}^{34})$  et de même les tâche du produit Pr4 doit être décalées par  $Rd_1(O_{24}^{44})$

#### ➤ Résultats générales

Généralement dans le plan prévisionnel la tâche  $O_{ij}^{uk}$  peut obéir à l'un des trois types de décalage

- Soit par le décalage généré par la tâche  $O_i^a$  (source de décalage) qui est défini comme suit :

$$Rd_i(O_{ij}^{uk}) = Rd_i \quad \text{III.19}$$

$$\text{Où } Rd_i = c_i^a - t_{ij}^{uk}$$

- Soit par le décalage généré par les tâches précédentes des autres produits qui s'exécutent sur la même machine. Le temps de décalage sera :

$$Rd_i(O_{xj}^{rk}) = \left\{ \begin{array}{ll} Rd_i - \sum_{i=s}^r d_{ik} & \text{si } Rd_i > \sum_{i=s}^r d_{ik} \\ 0 & \text{si } Rd_i < \sum_{i=s}^r d_{ik} \end{array} \right\} \quad \text{III.20}$$

On désigne par  $s$ : l'ordre de la première tâche concernée par le décalage dans la machine  $k$

$r$ : ordre de la tâche  $O_{xj}^{rk}$

- Soit par le décalage généré par le même produit

$$Rd_i(O_{ij}^{uk}) = Rd_i(O_{i'j}^{u'k'}) \quad \text{III.21}$$

Où  $O_{i'j}^{u'k'}$  représente la première tâche qui a été décalée pour le produit  $j$

Dans le cas où la tâche  $O_{ij}^{uk}$  doit avoir plusieurs types en même temps, le  $Rd_i(O_{ij}^{uk})$  est défini par le maximum

$$Rd_i(O_{ij}^{uk}) = \max [Rd_i(O_{ij}^{uk})^+, Rd_i(O_{i-1j}^{uk})] \quad \text{III.22}$$

### Remarques

Dans l'opération du décalage deux cas sont distingués et pour lesquels on doit stopper l'opération et passer à l'itération suivante :

- Cas où la date début de la tâche concernée par le décalage est inférieure à  $T^a$ .
- Cas où la mise à jour fait changer la position de la tâche précédente vers une position non admissible.

### III.3.3 Calcul de la nouvelle valeur de début des itérations

Dans cette partie, le passage d'une itération vers une autre repose sur la valeur minimale du rapport de passage ( $Rp_{min}$ ). Le but visé par ce choix est d'essayer d'examiner tout le plan et de ne pas éliminer les itérations qui peuvent contenir la bonne solution. Donc le calcul de la nouvelle valeur de début des itérations se fait par la relation (III.23)

$$t_1^a = t_1^a + Rp_{min} \quad \text{III.23}$$

### III.3.4 Choix du critère et définition de la bonne solution

Dans cette partie, la bonne solution est définie premièrement par la première position admissible trouvée avant que le délai de livraison n'est atteint et sans modification du plan prévisionnel, deuxièmement par la première solution qui donne une modification minimale du plan initial dans le cas où aucune position admissible n'est détectée. Cette modification touche généralement deux facteurs ; le facteur temps où la modification représente le changement des dates de livraison des commandes prévisionnelles (la satisfaction du client en termes de délai) et le facteur coût où la modification est traduite par le surcoût additionné au coût de production de ces commandes. Deux critères sont alors retenus pour définir la bonne solution ; la durée totale d'exécution « Makespan » et le coût de décalage (surcoût généré par l'opération de décalage). Il est à noter que le choix du critère dépend des préférences des décideurs.

#### III.3.4.1 Durée totale d'exécution « Maksepan »

Pour ce critère la bonne solution est définie par la position qui donne un minimum du Makespan dont l'objectif est de minimiser le délai de livraison des commandes.

Le calcul du Makespan, dans une itération, est effectué toujours à la fin de l'opération de décalage. La relation qui définit la valeur du Makespan est la suivante :

$$C_{max} = \max(\max_{j=1}^n(c'_{n_jj}), c'^a_{n_a}) \quad \text{III.24}$$

$c'_{n_jj}$  : La date de fin de la dernière tâche du job  $j$  (déjà ordonnancé);

$c'^a_{n_a}$  : La date de fin de la dernière tâche de la commande urgente.

#### III.3.4.2 Coût de décalage

Ce coût représente le coût supplémentaire généré par l'opération de décalage d'une tâche sur une machine où chaque machine est caractérisée par un coût unitaire de décalage différent.

Le calcul du coût total d'une procédure de décalage s'effectue par la relation suivante :

$$CG = \sum_1^4 (CU_{m_i} * \sum_1^{n_{m_i}} Rd_i(O_{\text{déc}}^{m_i})) \quad \text{III.25}$$

$CG$  : Le coût global de décalage

$CU_{m_i}$  : Le coût unitaire de décalage d'une tâche dans la machine  $m_i$

$n_{m_i}$  : Le nombre des tâches décalées utilisant la machine  $m_i$

$Rd_i(O_{\text{déc}}^{um_i})$  : Le temps de décalage de la tâche  $O_{\text{déc}}^{um_i}$

$O_{d\acute{e}c}^{um_i}$  : La  $u^{\text{ème}}$  tâche de la machine  $m_i$  qui a été décalée

L'algorithme suivant (Algorithme III.4) montre les différentes étapes de l'approche proposée pour la deuxième partie

<p><b>Etape 0 :</b>          Posez <math>Nd_{It} = \emptyset, i=1</math> et <math>t_1^a = T^a</math></p> <p><b>Etape 1 :</b> Chercher une position  <b>Tanque</b> <math>i \leq n_a</math> <b>faire</b>              Posez <math>k := m_i</math>              <b>si</b> <math>\exists ML_{uk}</math> réalise <math>t_i^a \in [dd_{uk}, df_{uk}[</math> <b>alors</b>                  Calculer <math>Ra_i^{It} = df_{uk} - c_i^a</math> et <math>Rp_i^{It} = dd_k(\text{secc } t_i^a) - t_i^a</math>                  Aller à l'<u>Etape 2</u>              <b>Sinon</b> La position est non admissible                  <math>Nd_{It} = Nd_{It} \cup O_i^a</math>                  passer à la tâche suivante <math>t_{i+1}^a := t_i^a + p_i^a, i := i+1</math>              <b>Fin si</b>  <b>Fin;</b> Aller à l'<u>Etape 3</u></p> <p><b>Etape 2 :</b> Tester l'admissibilité  <b>Si</b> <math>Ra_i^{It} \geq 0</math> <b>alors</b>              La position est admissible  <b>Sinon</b> La position est non admissible              <math>Nd_{It} = Nd_{It} \cup O_i^a</math>  <b>Fin si</b> passer à la tâche suivante <math>t_{i+1}^a := t_i^a + P_i^a, i := i+1</math>              Aller à l'<u>Etape 1</u></p> <p><b>Etape 3 :</b>  <b>Si</b> <math>Nd_{It} = \emptyset</math> Alors              Insérer la nouvelle commande  <b>Sinon</b>              Faire la procédure de décalage              Calculer les nouvelles valeurs              <math>C_{max}^{It} = \max(\max_{j=1}^n(c'_{n_j}), c'_{n_a})</math>              <math>CG^{It} = \sum_1^4(CU_{m_i} * \sum_1^{n_{m_i}} Rd_i(O_{d\acute{e}c}^{m_i}))</math>              <b>Si</b> le délai n'est pas atteint <math>t_1^a \leq DI^a - \sum_1^{n_a} p_i^a</math> <b>alors</b>                  Trouver <math>Rp_{min}^{It} = \min_1^{n_a}(Rp_i^{It})</math>                  Calculer la nouvelle valeur <math>t_1^a := t_1^a + Rp_{min}^{It}</math>                  Passer à l'itération suivante <math>i := 1, It := It+1</math>              Aller à l'<u>Etape 1</u>              <b>Sinon</b>                  Choisir le critère et sélectionner la bonne solution                  Insérer la nouvelle commande et modifier le plan prévisionnel              <b>Fin si</b>  <b>Fin si</b></p> <p><b>End</b></p>
--

**Algorithme III.4 :** Insertion des nouvelles tâches avec au sans décalage

### III.4 Exemple d'application

L'application suivante a comme objectif d'évaluer l'efficacité de l'approche d'insertion proposée et de quantifier son utilité apportée au système de production considéré. Pour cela on expose le déroulement de cette méthode et les résultats obtenus dans le cas d'apparition d'une nouvelle commande. En suite, et pour valorisé le gain fourni par la solution attribuée par cette approche, on fait une comparaison entre celle-ci et la solution donnée dans le cas où on fait bloquer la réalisation des commandes prévisionnelles jusqu'à la fin du traitement de la commande urgente. Cette comparaison est effectuée suivant les deux critères Makespan et coût de décalage.

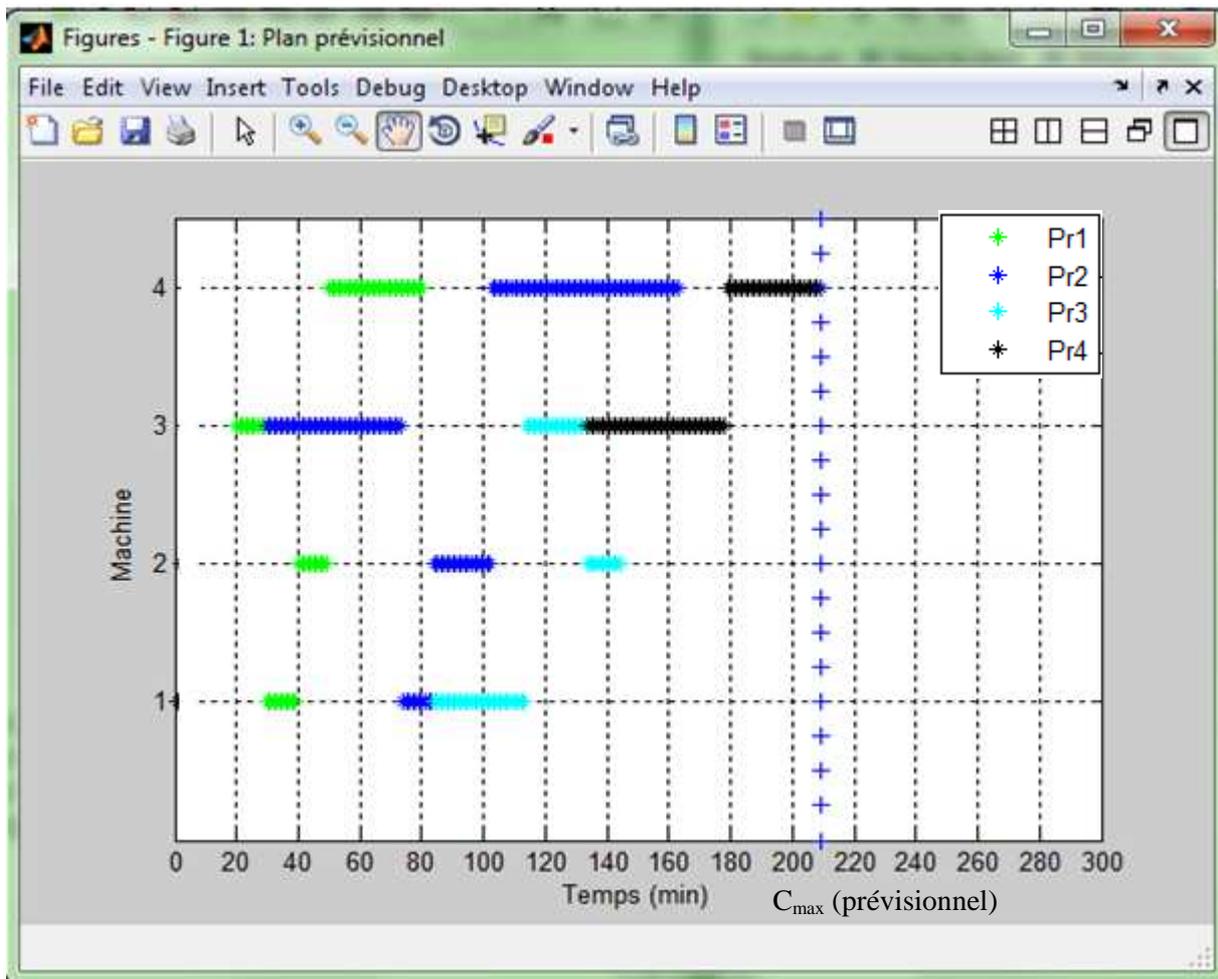
#### III.4.1 Données concernant le plan prévisionnel

Supposant que notre système est entrain de réaliser quatre types de produit Pr1, Pr2, Pr3 et Pr4, représentant quatre commande différentes. Les caractéristiques spécifiques de chacune de ces commandes sont montrées dans le Tableau III.1, celui ci définit pour chaque tâche d'un produit la durée d'exécution et la machine à utilisée.

produits	Pr1				Pr2				Pr3			Pr4	
Tâches	1	2	3	4	1	2	3	4	1	2	3	1	2
Machines utilisées	3	1	2	4	3	1	2	4	1	3	2	3	4
Durées des tâches (min)	10	10	10	30	44	10	19	30	30	20	10	45	30

**Tableau III.1** : Caractéristiques des commandes prévisionnelles

Les commandes prévisionnelles devront être réalisées suivant un plan prévisionnel qui représente pour le système un cycle de production. Les positions des différentes tâches sont définies par le diagramme de Gantt de la Figure III.9.



**Figure III.9 :** Diagramme de Gantt du plan prévisionnel

La Figure III.9 expose l'ordonnancement prévisionnel suivant lequel le système doit produire les quatre types de produit. Cet ordonnancement est caractérisé par un ensemble de marges libres représentant les temps morts des machines. Le tableau suivant (Tableau III.2) indique les dates de début et de fin de ces marges libre.

M <sub>1</sub>		M <sub>2</sub>		M <sub>3</sub>		M <sub>4</sub>	
dd <sub>i1</sub>	df <sub>i1</sub>	dd <sub>i2</sub>	df <sub>i2</sub>	dd <sub>i3</sub>	df <sub>i3</sub>	dd <sub>i4</sub>	df <sub>i4</sub>
0	0	0	0	0	20	0	0
0	30	0	40	30	30	0	50
40	74	50	84	74	114	80	103
84	84	103	134	134	134	163	179
114	+∞	144	+∞	179	+∞	209	+∞

**Tableau III.2 :** Caractéristiques des marges libres existantes dans le plan prévisionnel

### III.4.2 Application de l'approche d'insertion des tâches

Dans ce qui suit on montre les résultats obtenus par l'application de l'approche d'insertion dans le cas d'existence d'une nouvelle commande (N C). L'outil utilisé pour cette application est le logiciel MATLAB

MATLAB est un environnement complet, ouvert et extensible pour le calcul et la visualisation. Il dispose de plusieurs centaines de fonctions mathématiques, scientifiques et techniques. L'approche matricielle de MATLAB permet de traiter les données sans aucune limitation de taille et de réaliser des calculs numériques et symboliques de façon fiable et rapide. Grâce aux fonctions graphiques de MATLAB, il est très facile de modifier interactivement les différents paramètres des graphiques pour les adapter selon nos souhaits.

#### *Données :*

Supposant une nouvelle commande représentée par quatre tâches (Tableau III.3) apparaît dans le temps  $T^a=50$  min et qui doit être réalisé avant le délai  $Dl^a=250$ min.

Tâche	1	2	3	4
Machine à utilisée	1	3	2	4
Durée des tâches (min)	10	50	20	10

**Tableau III.3 :** Caractéristiques de la nouvelle commande

Ce tableau présente l'ensemble des tâches caractérisant la réalisation de la nouvelle commande ainsi que les machines utilisées pour chacune de ces tâches.

#### *Résultats :*

L'application de l'approche d'insertion montre qu'après sept itérations de la recherche aucune position admissible n'est trouvée avant le délai de livraison et sans modification du plan prévisionnel. Mais avec la procédure de décalage on a obtenu les résultats suivants :

Itération	1	2	3	4	5	6	7
Début de la position définie	50	64	83	84	114	124	129
Possibilité d'insertion	0	0	0	1	1	1	1
Makespan (min)	209	219	238	239	269	259	264
Coût de décalage (DA)	0	11	31	33	66	30	33

**Tableau III.4 :** Résultats obtenu pour chaque itération

Le tableau III.4 décrit à chaque itération le début de la position définie par l'opération de la recherche qui représente le début de la première tâche de la commande urgente. Il décrit aussi la possibilité de faire le décalage et d'insérer la nouvelle commande dans ces positions ainsi que les différentes valeurs des deux critères Makespan et le coût de décalage résultant de l'opération de décalage effectuée dans chaque itération.

La bonne solution déterminée par le premier critère Makespan est défini dans le Tableau III.5.a, par contre l'autre déterminée selon le critère coût de décalage est montrée dans le Tableau III.5.b.

Tâche	1	2	3	4
Date début	84	94	144	164
Date fin	94	144	164	174

Tâche	1	2	3	4
Date début	124	134	184	204
Date fin	134	184	204	214

**a .** La solution retenue selon le Makespan      **b .** La solution retenue selon le coût de décalage

**Tableau III.5 :** Le choix de la bonne solution

***Interprétation des résultats :***

\* Selon le Tableau III.4 on remarque que pour les trois premières itérations de la recherche, le décalage n'était pas permis. Cela revient au fait que l'exécution de certains produits concernés par cette opération est déjà commencée et que celle-ci ne doit pas être interrompue comme le montre l'exemple de la Figure III.10

\* L'augmentation remarquée dans la deuxième et la troisième itération, concernant les valeurs du Makespan et le coût de décalage, présente seulement une image sur ce que doit passer si le décalage est permis.

\* Le choix du critère Makespan a conduit à sélectionner la position définie dans la quatrième itération (Figure III.11.a et b) par contre le choix du critère coût de décalage donne comme solution la position définie dans la sixième itération.

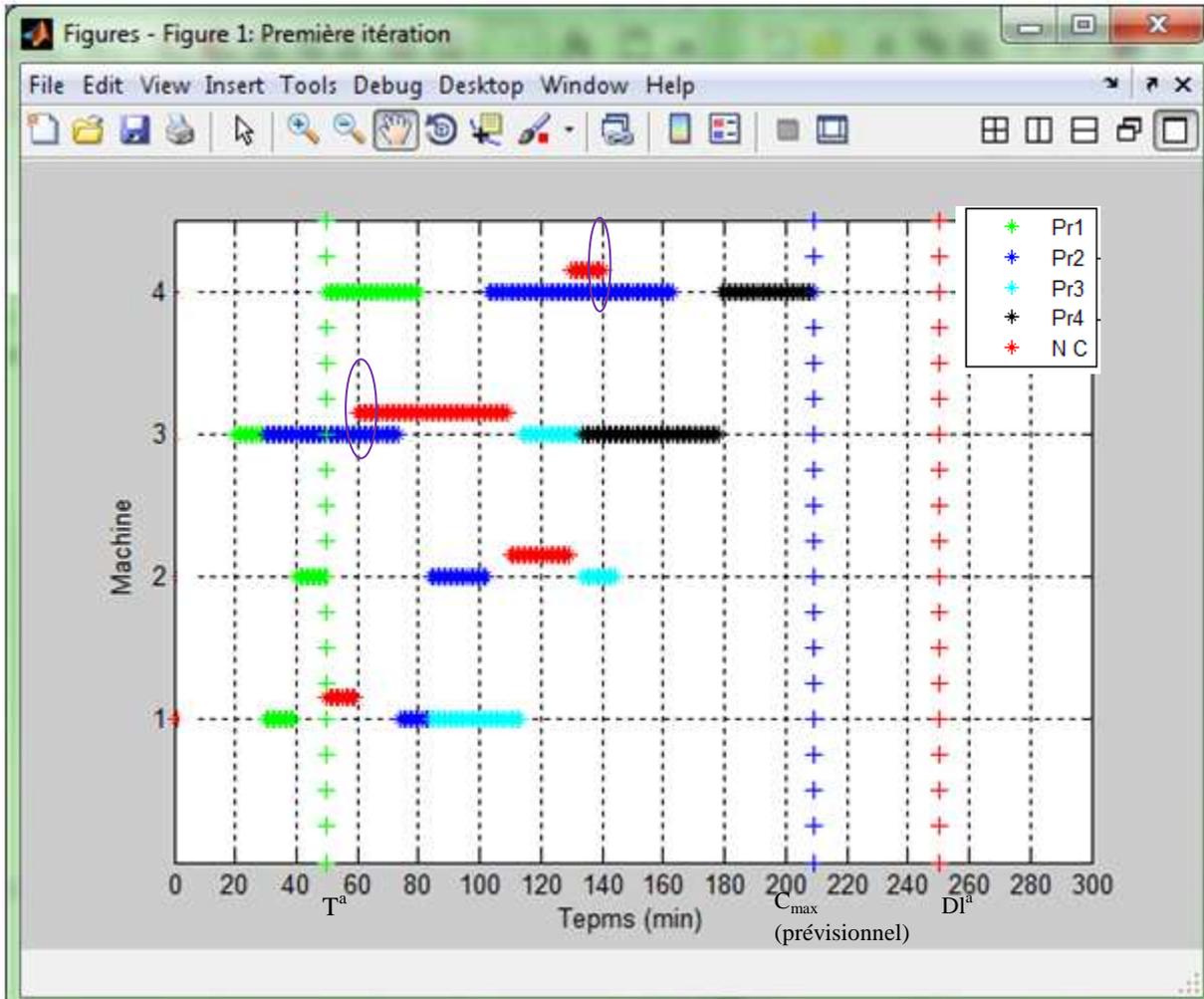


Figure III.10 : Résultat de la première itération

La Figure III.10 présente la position d'insertion définie dans la première itération où on distingue que le produit concerné par l'opération de décalage est le produit Pr2. L'exécution de ce produit a été commencée avant la date de début de la position définie, donc le décalage ici n'est pas autorisé.

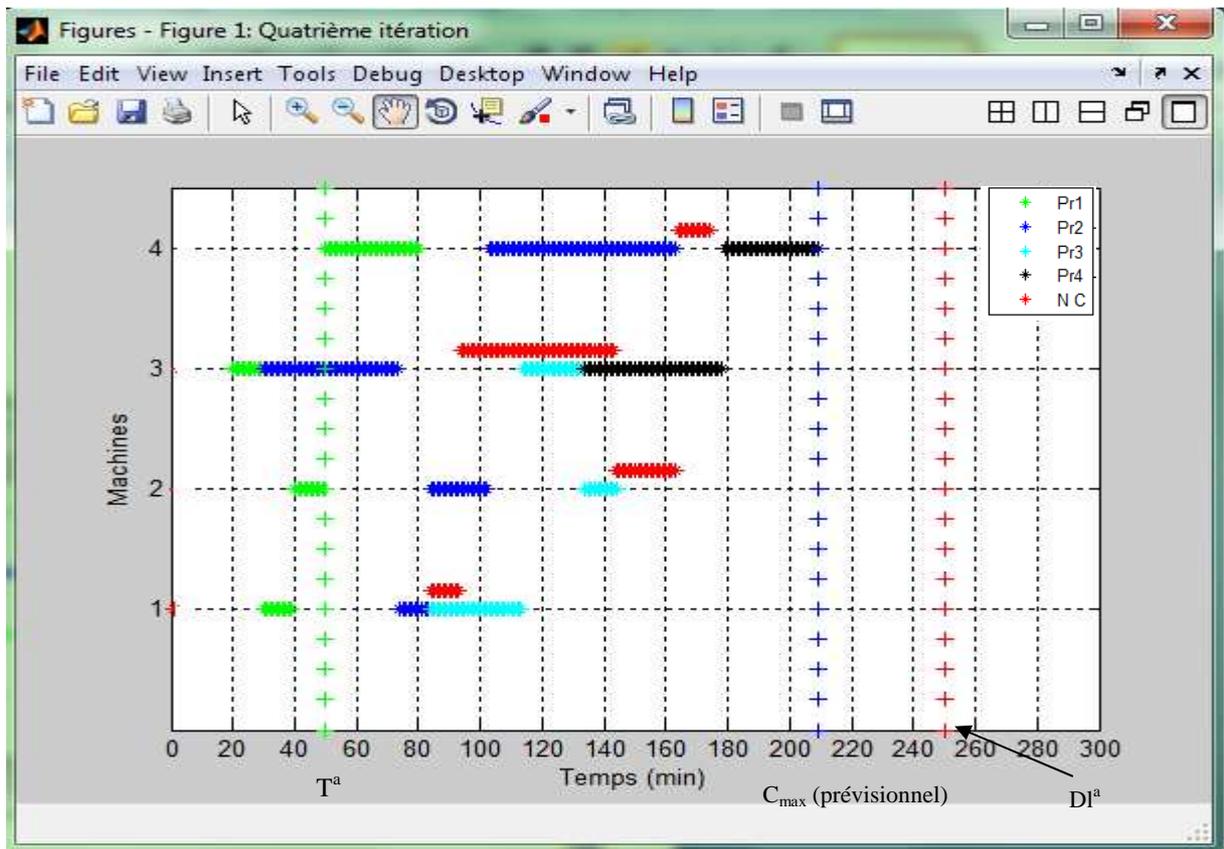


Figure III.11.a : Résultat de la quatrième itération avant le décalage

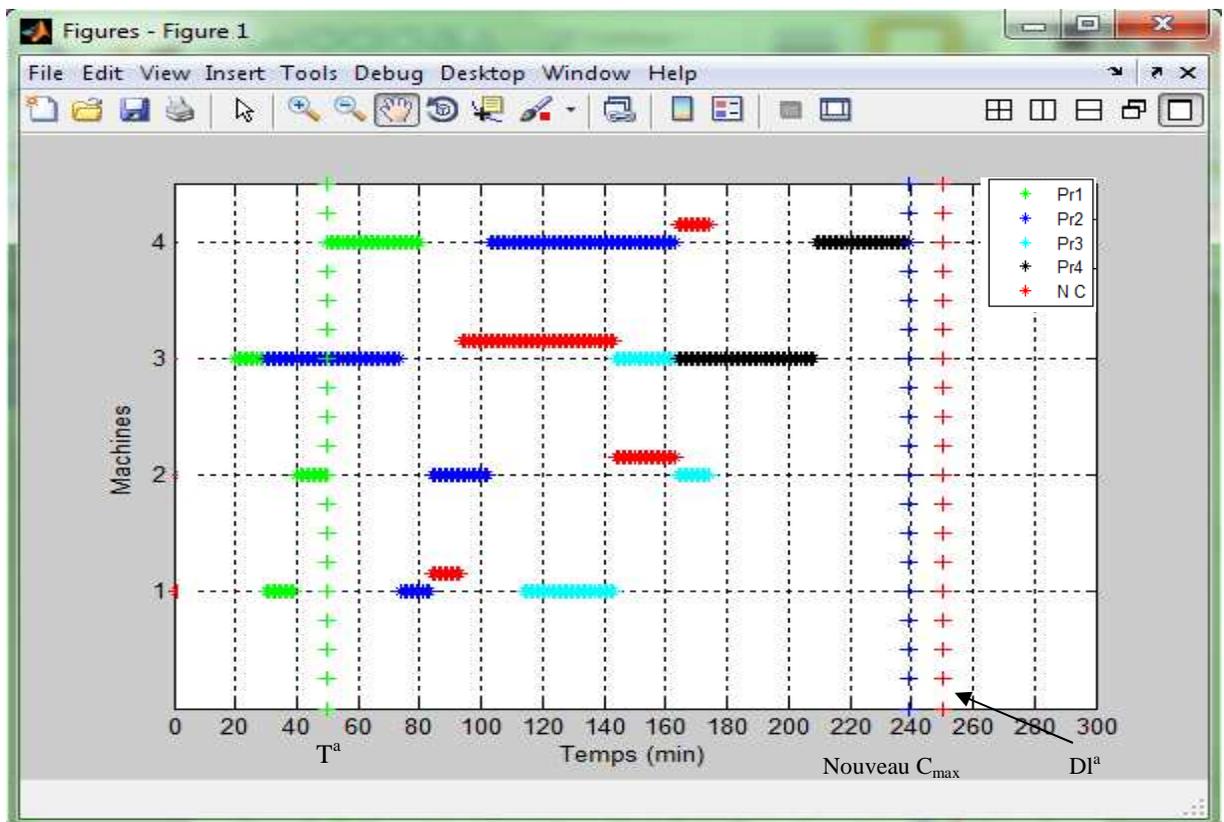


Figure III.11.b : Résultat de la quatrième itération après le décalage

### III.4.3 Analyse par variation de la date d'apparition

En se basant sur la même commande urgente, L'objectif dans cette étape est de tester l'approche proposée pour des valeurs différentes de la date d'apparition de cette commande. Pour ceci on fait varier d'une façon uniforme les dates d'apparition et on définit à chaque fois la valeur du Makespan et le coût correspondant à la bonne solution qu'on peut trouver. Sur la base du plan prévisionnel on fait ensuite une comparaison entre ces valeurs et les valeurs qu'on peut trouver dans le cas où la réalisation des commandes prévisionnelles est interrompue par l'apparition de la nouvelle commande.

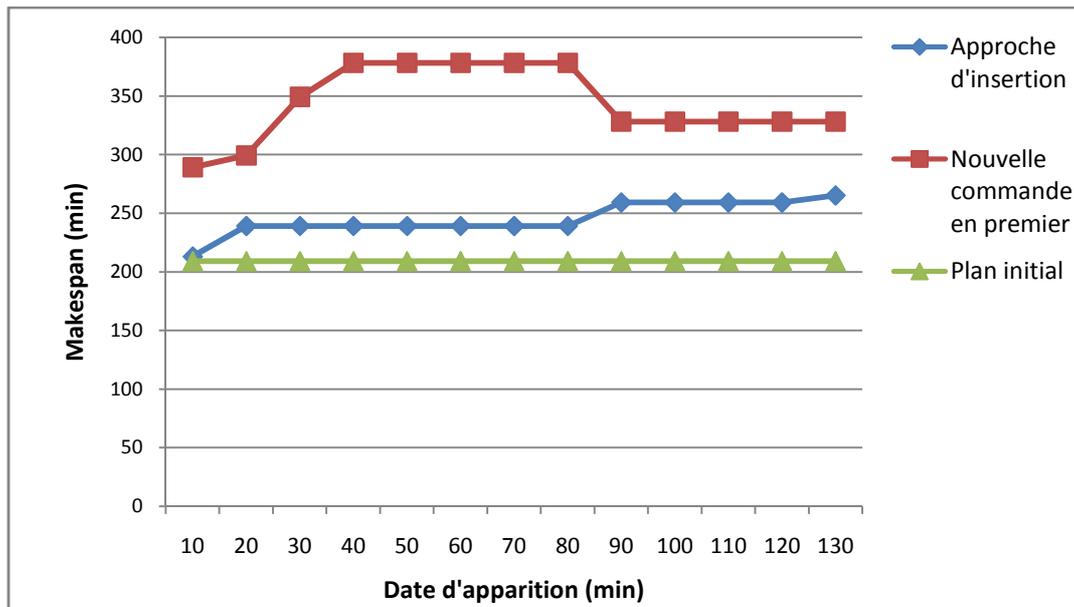
#### III.4.3.1 Résultats selon le Makespan

Le tableau suivant montre les différentes valeurs du Makespan qui sont retenues pour les différentes dates d'apparition de la commande urgente.

Date d'apparition	Approche d'insertion	Nouvelle Commande en premier	Plan initial
10	213	289	209
20	239	299	209
30	239	349	209
40	239	378	209
50	239	378	209
60	239	378	209
70	239	378	209
80	239	378	209
90	259	328	209
100	259	328	209
110	259	328	209
120	259	328	209
130	265	328	209

**Tableau III.6 :** Résultats des valeurs du Makespan selon la variation des dates d'apparition de la nouvelle commande

À partir de ce tableau et selon la variation du Makespan on peut tracer les courbes suivantes (Figure III.12)



**Figure III.12 :** Variation du Makespan selon la date d'apparition

Cette Figure représente la variation du Makespan suivant la variation de la date d'apparition de la nouvelle commande

Par comparaison entre ces trois courbes on remarque bien que l'augmentation du Makespan générée par l'approche d'insertion est toujours minimale par rapport à la deuxième proposition

#### III.4.3.2 Résultats selon le coût de décalage

Supposant que le coût unitaire de chacune des machines est défini dans le tableau suivant :

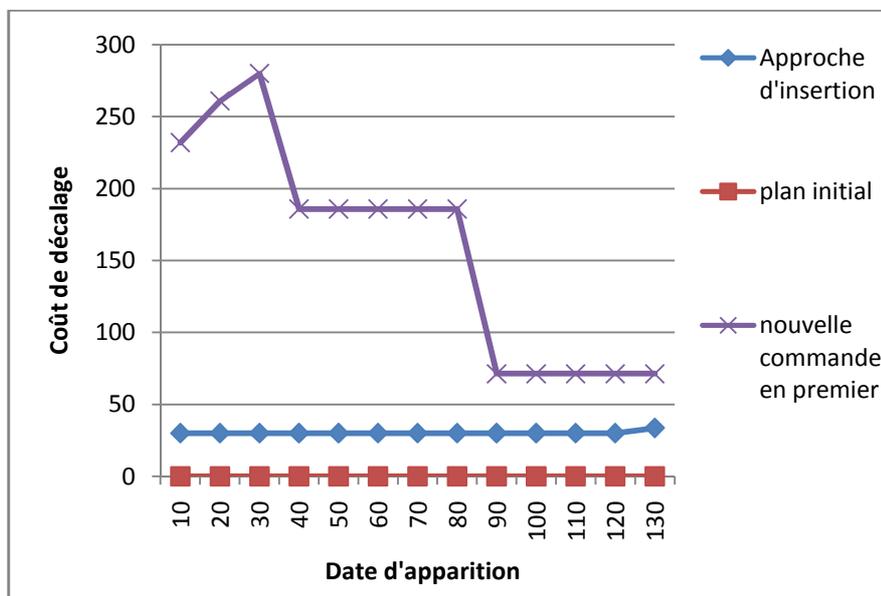
Machines	1	2	3	4
Coût unitaire (DA/min)	0.05	0.25	0.20	0.4

**Tableau III.7 :** Les coûts de décalage unitaire des machines

Le tableau suivant présente les différentes valeurs du coût de décalages obtenus par les deux propositions avec variation des dates d'apparition.

Date d'apparition	Approche d'insertion	Nouvelle Commande en premier	Plan initial
10	30	232	--
20	30	261	--
30	30	280	--
40	30	185,9	--
50	30	185,9	--
60	30	185,9	--
70	30	185,9	--
80	30	185,9	--
90	30	71,4	--
100	30	71,4	--
110	30	71,4	--
120	30	71,4	--
130	33.6	71,4	--

**Tableau III.8 :** Résultats retenus pour les valeurs du coût de décalage



**Figure III.13 :** Variation du coût de décalage

D'après cette Figure on remarque que quel que soit la date d'apparition de la nouvelle commande, le coût de décalage résultant de l'approche d'insertion est toujours constant sauf

la dernière. Cette valeur constante des coûts de décalage revient au fait que la bonne solution déterminée par ces dates d'apparition est toujours la même, donc c'est le même coût calculée.

#### III.4.4 Analyse par variation des commandes

Dans cette étape on fixe la date d'apparition et le délai de livraison et on fait varier la nouvelle commande. Pour ceci, on a choisi dix commandes différentes pour les introduire dans le plan initial (séparément).

L'ensemble des caractéristiques de ces commandes sont montré dans le tableau suivant :

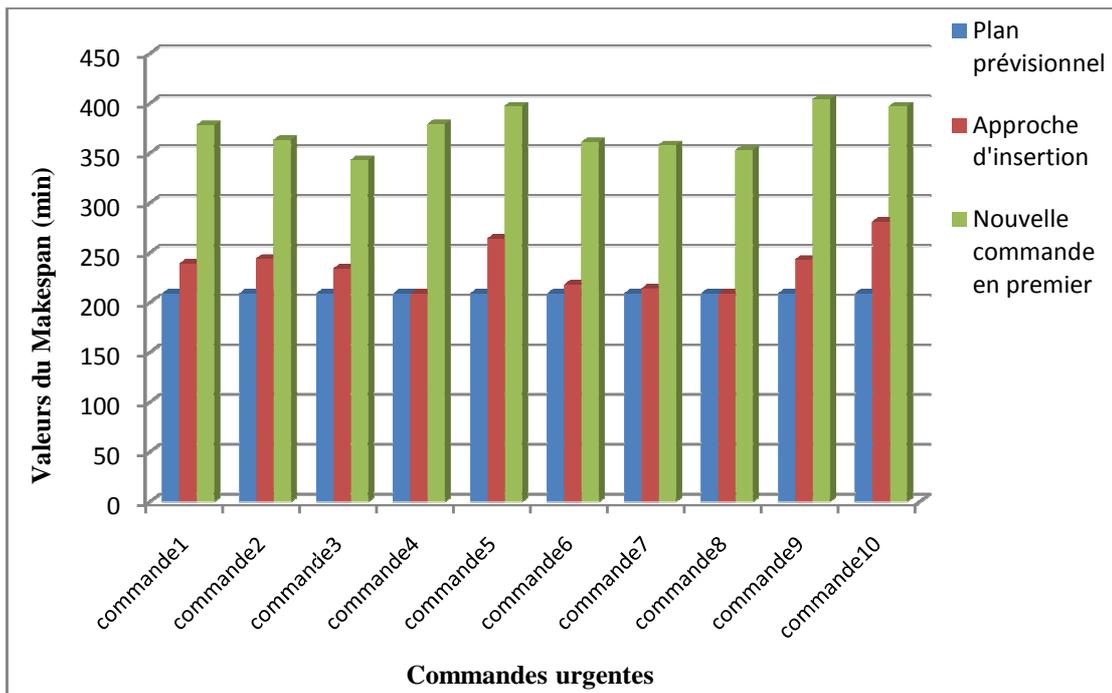
Commande	1				2		3			4			5				
Tâche	1	2	3	4	1	2	1	2	3	1	2	3	1	2	3	4	5
Machine à utilisée	1	3	2	4	2	4	3	4	1	3	1	2	4	1	3	2	2
Durée des tâches	10	50	20	10	40	35	10	15	30	14	35	42	7	11	14	35	42
Commande	6				7		8			9		10					
Tâche	1	2	3	4	1	1	2	3	1	2	1	2	1	2	3	4	5
Machine à utilisée	4	3	2	1	2	3	2	1	3	1	2	1	4	3	2		
Durée des tâches	30	10	23	10	70	10	25	30	74	42	7	11	14	35	42		

**Tableau III.9 :** Caractéristique des commandes choisies

##### III.4.4.1 Résultats selon le Makespan

Nouvelles commandes	Approche d'insertion	Nouvelle commande en premier	Plan prévisionnel
Commande 1	239	378	209
Commande 2	244	363	209
Commande 3	234	343	209
Commande 4	209	379	209
Commande 5	264	397	209
Commande 6	218	361	209
Commande 7	214	358	209
Commande 8	209	353	209
Commande 9	243	404	209
Commande 10	281	397	209

**Tableau III.10 :** Résultats du Makespan obtenu par variation des commandes



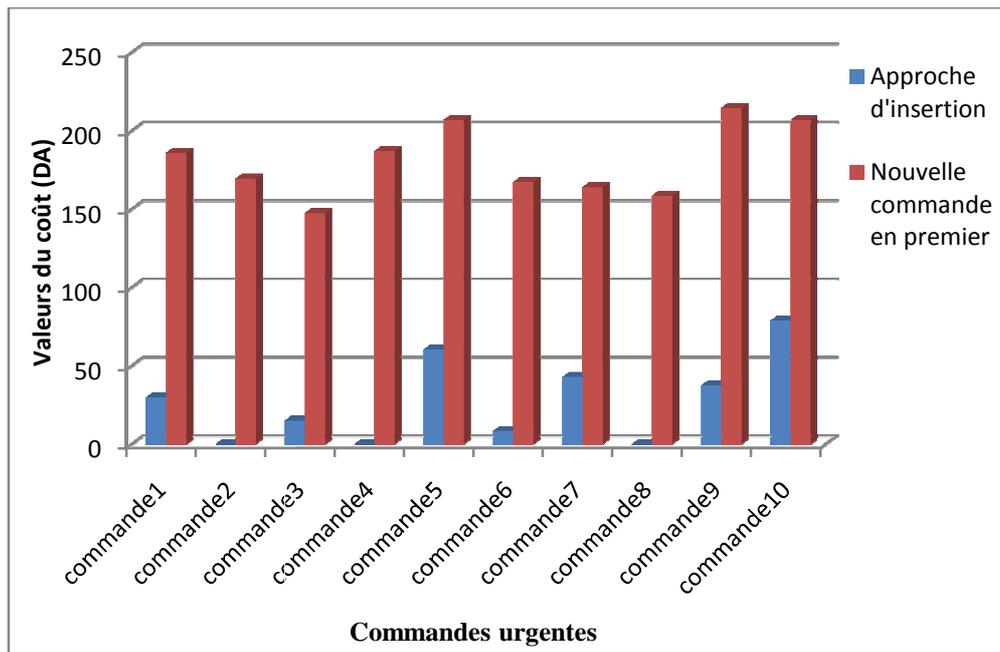
**Figure III.14 :** Histogramme du Makespan suivant la variation des commandes

À partir de cet histogramme (Figure III.14) on peut constater que l'utilisation de l'approche d'insertion a généré des augmentations minimales au niveau du Makespan par rapport à la deuxième proposition (nouvelle commande en premier).

#### III.4.4.2 Résultats selon le coût de décalage

Nouvelles commandes	Approche d'insertion	Nouvelle commande en premier	Plan prévisionnel
Commande 1	30	185,9	--
Commande 2	0	169,4	--
Commande 3	15	147,4	--
Commande 4	35	187	--
Commande 5	60,5	206,8	--
Commande 6	8.4	167,2	--
Commande 7	0	163,9	--
Commande 8	0	158,4	--
Commande 9	37.4	214,5	--
Commande 10	79	206,8	--

**Tableau III.11 :** Résultats du coût de décalage obtenu par variation des commandes



**Figure III.15 :** Histogramme du coût suivant la variation des commandes

On remarque bien selon ces résultats que la solution trouvée par l'application de l'approche d'insertion est toujours meilleure que celle générée par l'interruption du plan prévisionnel.

### III.5 Conclusion

Dans cette partie du mémoire on a donné une description détaillée de l'approche d'insertion des tâches proposée. On a montré alors le détail de ses deux parties (sans et avec délai de livraison). Cette approche est traduite par un algorithme qui sert à introduire les nouvelles commandes dans le plan prévisionnel.

Théoriquement on peut dire que l'approche proposée conduit d'une manière importante aux objectifs tracés au début de ce chapitre. Cela revient aux causes suivant :

- Le démarrage de l'opération de la recherche à partir de l'instant d'apparition de la commande et le choix de la première position trouvée permet au système de traiter la commande le plus vite possible, c'est le premier objectif.
- Avec le  $Rp_{max}$  retenu pour le calcul de la valeur de la date  $t_1^a$  nous pouvant minimiser le temps de calcul et éliminer un ensemble des itérations non utiles.

L'utilisation de l'outil MATLAB dans l'application numérique effectuée à la fin de ce chapitre nous a permis d'analyser les performances de l'approche d'insertion et d'inspirer son utilité apportée au système considéré.

À partir de deux analyses réalisées, soit par variation de la date d'apparition où par variation des nouvelles commandes, on a pu déduire que les résultats donnés par l'approche d'insertion sont des résultats acceptables pour les deux critères Makespan et coût de décalage.

# Conclusion Générale

## Conclusion générale

La principale contribution de notre travail consigné dans ce mémoire est la résolution de l'un des problèmes d'ordonnancement d'un système de production de type job shop. Ce problème est présenté par l'occurrence des nouvelles commandes qui doivent être insérées dans le programme de production courant de ce système. La propriété « succession sans interruption » imposée par l'absence des stocks intermédiaires dans le système étudié a fait la différence entre le problème traité dans ce mémoire et d'autres qui sont trouvées dans la littérature et qui concernent l'insertion des tâches indépendantes.

Pour la résolution de ce problème nous avons proposé une approche approximative elle consiste à générer un ordonnancement en temps réel qui donne la possibilité d'introduire ces commandes urgentes dans le plan prévisionnel. Dans cette direction, nous avons devisé l'approche en deux parties selon deux cas possible. La première partie (sans délai de livraison), elle consiste à résoudre le problème dans le cas où les commandes urgentes ne sont pas prioritaires en cherchant à les insérer sans tenir compte de leurs délais de livraison. La deuxième partie (avec délai de livraison), elle donne une solution dans le cas où les commandes urgentes ne doivent pas dépasser ces délais. Cette solution est proposée avec possibilité de modification du plan prévisionnel si c'est nécessaire. L'exemple d'application effectué à la fin de ce mémoire nous a permis d'évaluer l'efficacité de l'approche proposée et de valoriser son intérêt apporté au système considéré.

Notre travail a permis donc de développer un algorithme permettant d'insérer les nouvelles commandes dans un ordonnancement existant, sous la contrainte temps-réel. Cet algorithme est considéré comme efficace et donne de bons résultats dans le cadre des hypothèses définies auparavant. Un inconvénient qui peut être significatif est que l'algorithme proposé ne peut pas traiter le cas où une commande contient deux ou plus de tâches (non successives) et qui doivent être exécutées sur une même machine (c'est un cas très rare pour les systèmes manufacturières).

On peut donc résumer l'utilité de ce travail par les points suivants :

- Proposition d'une approche approximative pour la résolution du problème d'insertion des nouvelles commandes pour le système considéré.
- La création d'un premier algorithme qui permet de générer des positions d'insertion possibles sous la contrainte succession sans interruption.
- La création d'un deuxième algorithme qui permet de tester l'admissibilité des ces positions

- Utilisation de deux critères Makespan et coût de décalage pour sélectionner la bonne solution
- Développer un algorithme global destiné à l'insertion des nouvelles commandes dans les deux cas (sans et avec délai de livraison).

***Perspectives***

Il serait très intéressant de pouvoir améliorer certains autres aspects dans l'approche proposée qui sont envisagés en perspectives :

- l'utilisation de méthodes performantes comme les Algorithmes Génétique et la Recherche Tabou... dans la phase de définition des positions d'insertion.
- tenir compte d'autres critères pour la sélection des solutions tels que la fusion entre les deux critères (Makespan et coût de décalage).

# Bibliographie

## Bibliographie

- [1] Lopez P. et Roubellat F., « *Ordonnancement de la production* », Hermès Sciences, IC2 Productique, 2000.
- [2] Mebarek K., « *Utilisation des stratégies Méta-heuristiques pour l'ordonnancement des ateliers de type Job Shop* », Mémoire de Magister, Université de Batna, Algérie, 2008.
- [3] Azem S., « *Ordonnancement des systèmes flexibles de production sous contrainte de disponibilité des ressources* », Thèse de doctorat, École Nationale Supérieure des Mines de Saint-Étienne, Gardanne, 2010.
- [4] Esquirol P. et Lopez P. « *L'ordonnancement* », Economica, Paris, 1999.
- [5] Dupy M. « *contribution à l'analyse des systèmes industriels et au problème d'ordonnancement à machines parallèles flexibles : application aux laboratoires de contrôle qualité en industrie pharmaceutique* », Thèse de doctorat, Toulouse, 2005.
- [6] Marmier F., « *Contribution à l'ordonnancement des activités de maintenance sous contrainte de compétence : une approche dynamique, proactive et multicritère* », Thèse de doctorat, Université Franche-Comté, 2007.
- [7] Younes M. « *Aide à la décision d'ordonnancement inter-entreprise dans le cadre des travaux de l maintenance, cas : Société de Maintenance de l'Est (SME)* » Mémoire de Magister, Université de Batna, Algérie, 2011.
- [8] Jean-Charles B. et Roubellat F., « *A new method for workshop real time scheduling* », International Journal of Production Research, 34(6), 1996.
- [9] Hentons H., « *Contribution au pilotage des systèmes de production de type Job Shop* », Thèse de doctorat, INSA Lyon, 1999.
- [10] Boukef Ben Othman H., « *Sur l'ordonnancement d'ateliers job-shop flexibles et flow shop en industries pharmaceutiques Optimisation par algorithmes génétiques et essais particuliers* », Thèse de doctorat, Ecole Centrale De Lille, 2009.
- [11] Pinedo M., « *Scheduling: Theory, Algorithms and Systems* », Third edition, Springer, 2008.
- [12] BAPTISTE P. and LE PAPE C., « *A Constraint Based Branch and Bound Algorithm for Preemptive Job Shop Scheduling* », Proceedings of the International Workshop on Production Planning and Control, Mons, Belgium, 1996.
- [13] Bronson R., Naadimuthu G., « *Operation research* », Second edition, Schaum's Outline Series, 1997.
- [14] Blum C., Andrea R., « *Metaheuristics In Combinatorial Optimization: Overview And Conceptual Comparison* », ACM Computing Survey, Vol. 35, n 3, 2003.

- [15] Letouzey A., « *Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs* », Thèse de doctorat, Institut National Polytechnique, Toulouse, 2001.
- [16] Artigues C., « *Ordonnancement En Temps Réel d'Ateliers avec temps de préparation des ressources* », Thèse de doctorat, Université Paul Sabatier, Toulouse, 1997
- [17] Kermia O., « *Ordonnancement temps réel multiprocesseur avec contraintes de précédence et de périodicité stricte et de latence* », Thèse de doctorat, université de Paris XI, 2009.
- [18] Roy B., Susmann B., « *Les Problèmes d'ordonnancement avec Contraintes Disjonctives* », Note DS n 9 bis, SEMA, Montrouge, 1964.
- [19] Duron1 C., Ould Louly1 M.A. et Proth3 J.-M., « *Ordonnancement d'une ressource unique : insertion d'une tâche aléatoire sous la contrainte temps-réel* », 5<sup>e</sup> Conférence Francophone de Modélisation et Simulation, Nantes (France), 2004.
- [20] Stankovic J., Spuri M., Natale M., et Bittazzo G. « *implication of Classical scheduling Results for real time systems* », Computer, vol, 23 n°4, 1995.
- [21] Marmier F., Varnier C., Zerhouni N., « *Ré-ordonnancement partiel et dynamique d'un planning* », 7<sup>ème</sup> Congrès International de Génie Industriel, Québec, Canada, 2007.
- [22] Tuncel G., « *A Heuristic Rule-Based Approach for Dynamic Scheduling of Flexible Manufacturing Systems* », ISBN 978-3-902613-02-8, Itech Education and Publishing, Vienna, Austria, 2007.
- [23] Kouider A., Ourari S., Benabbas S. et Mihobi M., « *Approche heuristique pour l'ordonnancement temps réel d'une machine dans un contexte perturbé* », USTBH, Al Alia Bab Azzouar, algérie, 2002.
- [24] Javel G., « *Organisation et gestion de la production, cours avec exercices corrigés* », 4<sup>ème</sup> édition, Dunod, Paris, 2010.
- [25] Giard V., « *Gestion de la production* », Economica, 2003.
- [26] Tamani K., « *Développement d'une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production* », Thèse de doctorat, Université de SAVOIE, 1992.
- [27] Anthony, R. N., « *Planning and Control Systems: Framework for Analysis* », Harvard Graduate School of Business Administration, Boston, 1965.
- [28] Giard V., « *Gestion de production* », 2<sup>ème</sup> édition, Economica, paris, 1988.

- [29] Mirdamadi S., « Modélisation du processus de pilotage d'un atelier en temps réel à l'aide de la simulation en ligne couplée à l'exécution », thèse, Université de Toulouse, 2009.
- [30] VACHER J., « *Un système adaptatif par agents avec utilisation des algorithmes génétiques multi-objectifs : application à l'ordonnancement d'atelier de type job-shop  $N \times M$*  », Thèse de Doctorat, Université du Havre, 2000.
- [31] Blondel F., « *Gestion de la production* », 3<sup>ème</sup> édition, DUNOD, Paris, 2004.
- [32] Trung L., « *utilisation d'ordres partiels pour la caractérisation de solutions robustes en ordonnancement* », Thèse de doctorat, institut national des sciences appliquées, Toulouse, 2005.
- [33] Erschler J., De Terssac G., « *Flexibilité et rôle de l'opérateur humain dans l'automatisation intégrée de production* », Rapport laas no 88137, Laboratoire d'Analyse et d'Architecture des Systèmes, Toulouse, France, 1988.
- [34] Nagarur N., « *Some performance measures of flexible manufacturing systems* », International Journal of Production research, 30(4), 1992.
- [35] GOTH A., « *Flexibilité et Robustesse en Ordonnancement* ». Bulletin de la Société Française de Recherche Opérationnelle et d'Aide à la décision, vol. 8, 2002.
- [36] Browne J., Dubois D., Rathmill K., Sethi S.P., and Stecke K.E., « *Classification of flexible manufacturing systems* », The FMS Magazine, 2(2) , 1984.
- [37] Thomas C., « *Analyse de la flexibilité : le cas d'une unité de production d'aluminium* », Thèse de doctorat, Institut National Polytechnique de Grenoble, 3003.

# Annexes

## Annexes

### Application par MATLAB

Insertions des données :

```
Date d'apparition de la commande = 50
Nombres des taches = 4
Machines à utiliser = [1 3 2 4]
Les durées des taches = [10 50 20 10]
Délai de livraison = 250
```

Résultat numérique :

```
Insertion impossible Voulez vous faire le décalage ? ((oui=1, non=0)): 1

***** Le nombre des itérations *****
NIT =
  7

***** La possibilité d'insertion dans chaque itération *****
poit =
  0  0  0  1  1  1  1

***** Début de la position dans chaque itération *****
debutit =
  50  64  83  84  114  124  129

Quel critère vous choisissez?((Makespan =1, Coût de décalage =2)): 1

***** Le Makespan de chaque itération *****

Maxepen =

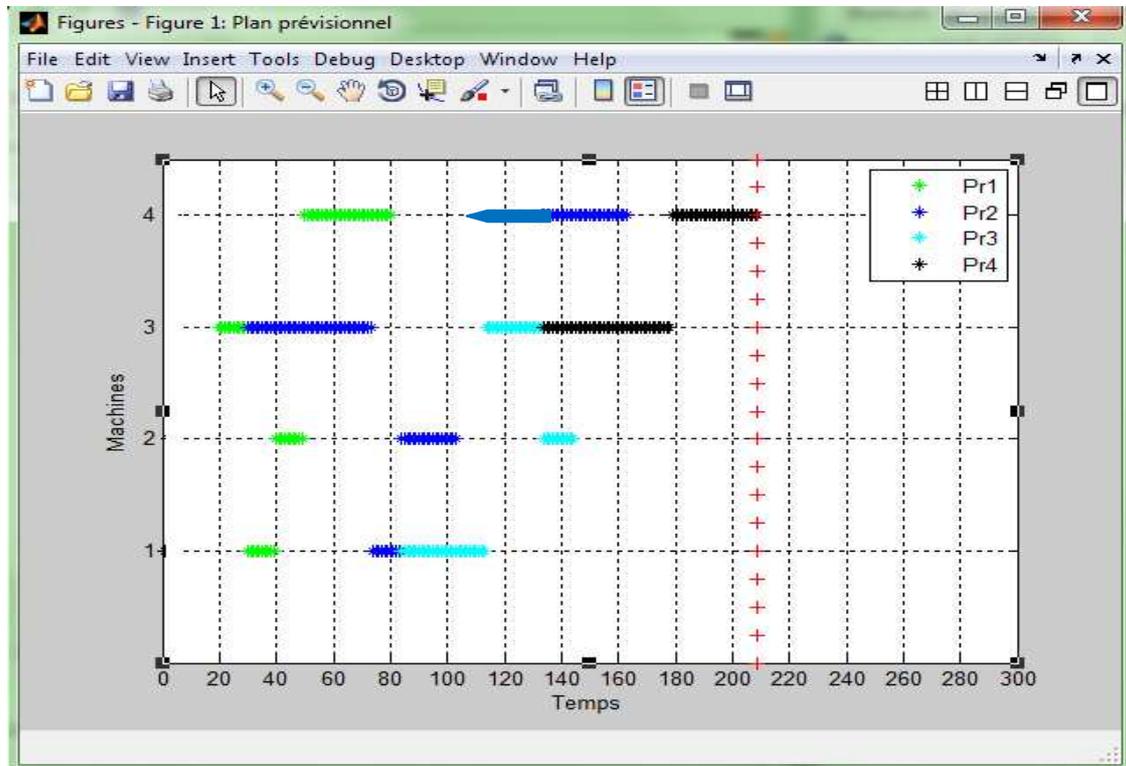
  209  219  238  239  269  259  264

***** Itération objectif *****
obj =
  4

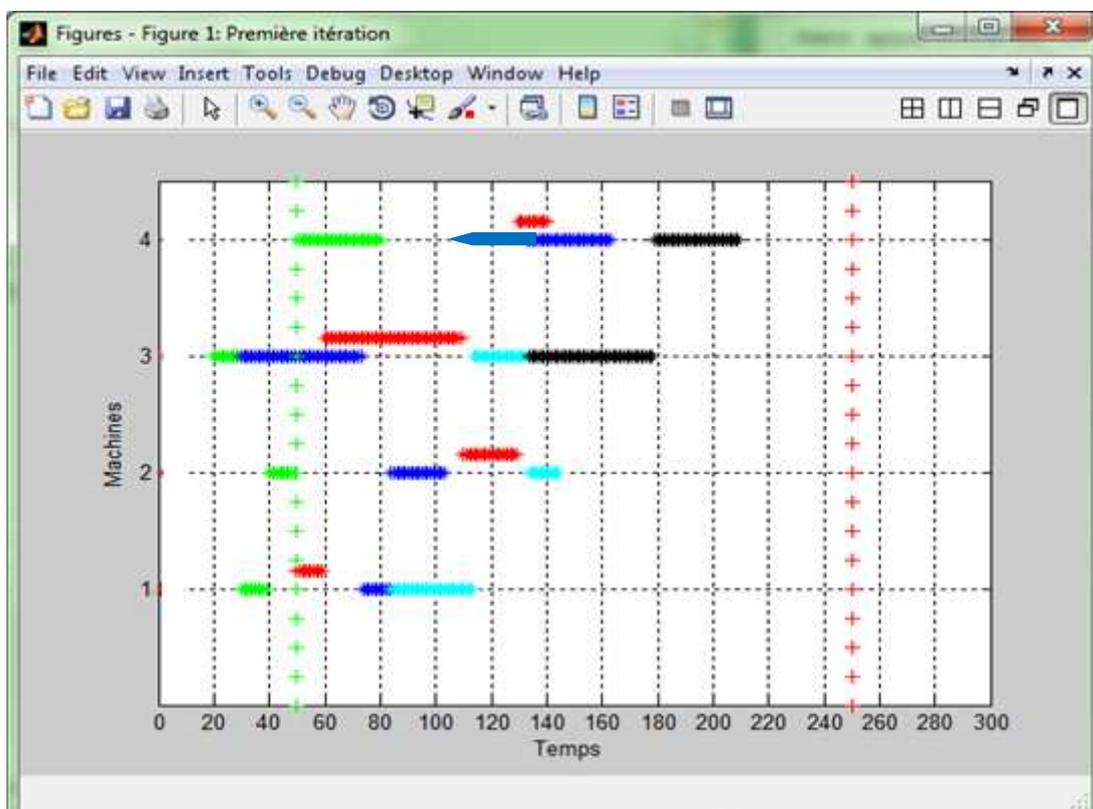
***** La position objectif *****
T =
  84  94  144  164

C =
  94  144  164  174
```

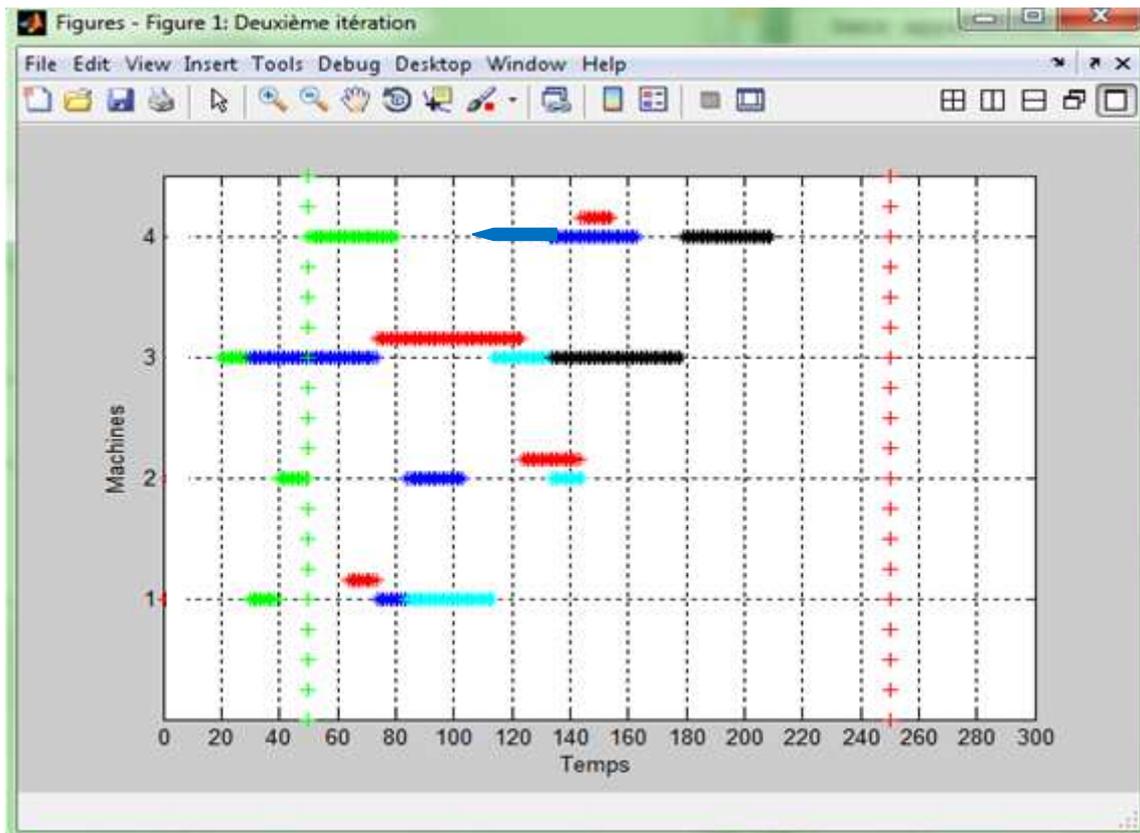
Résultats graphique :



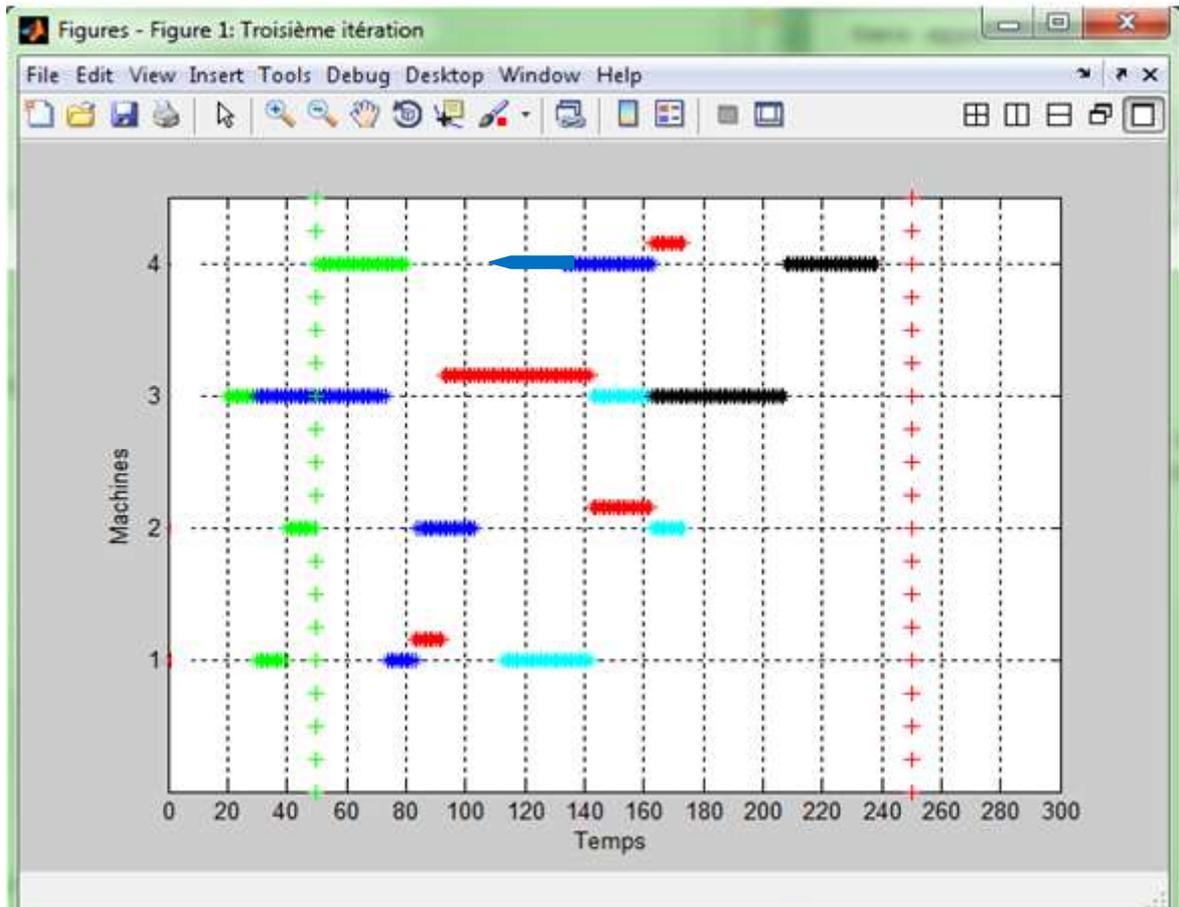
**Plan prévisionnel**



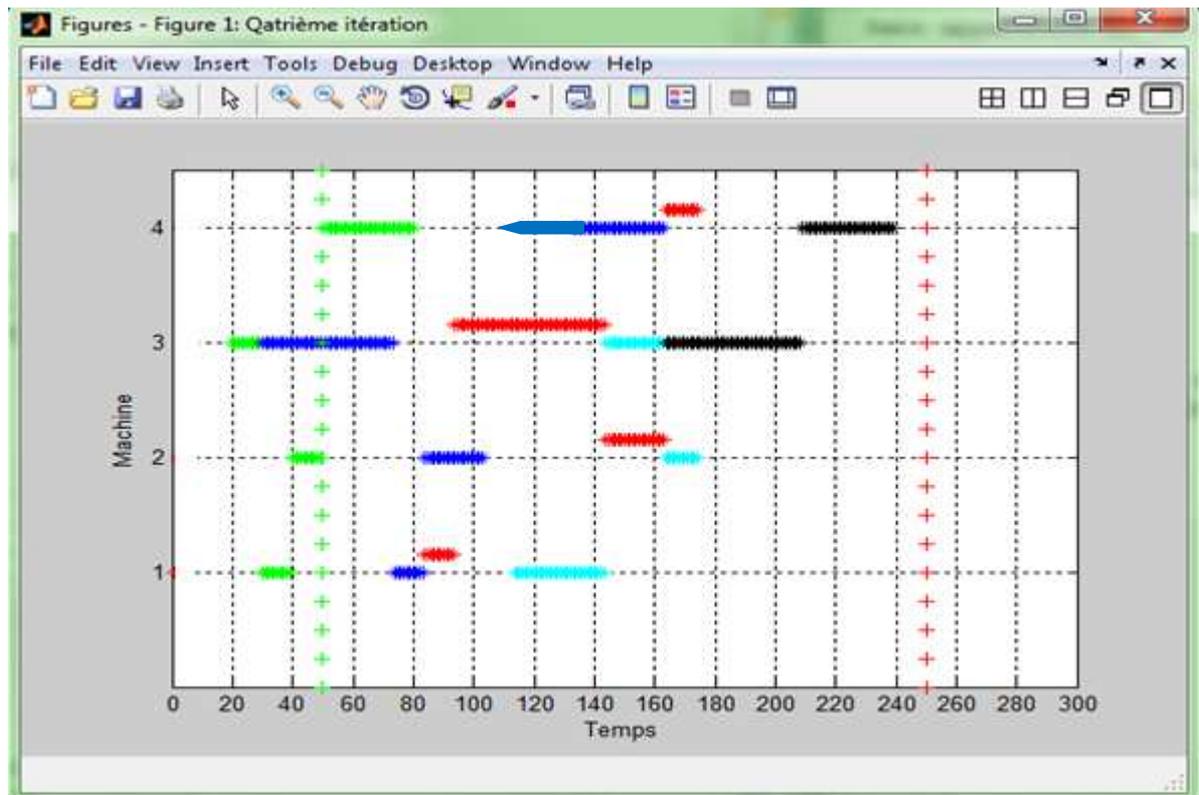
**Première itération**



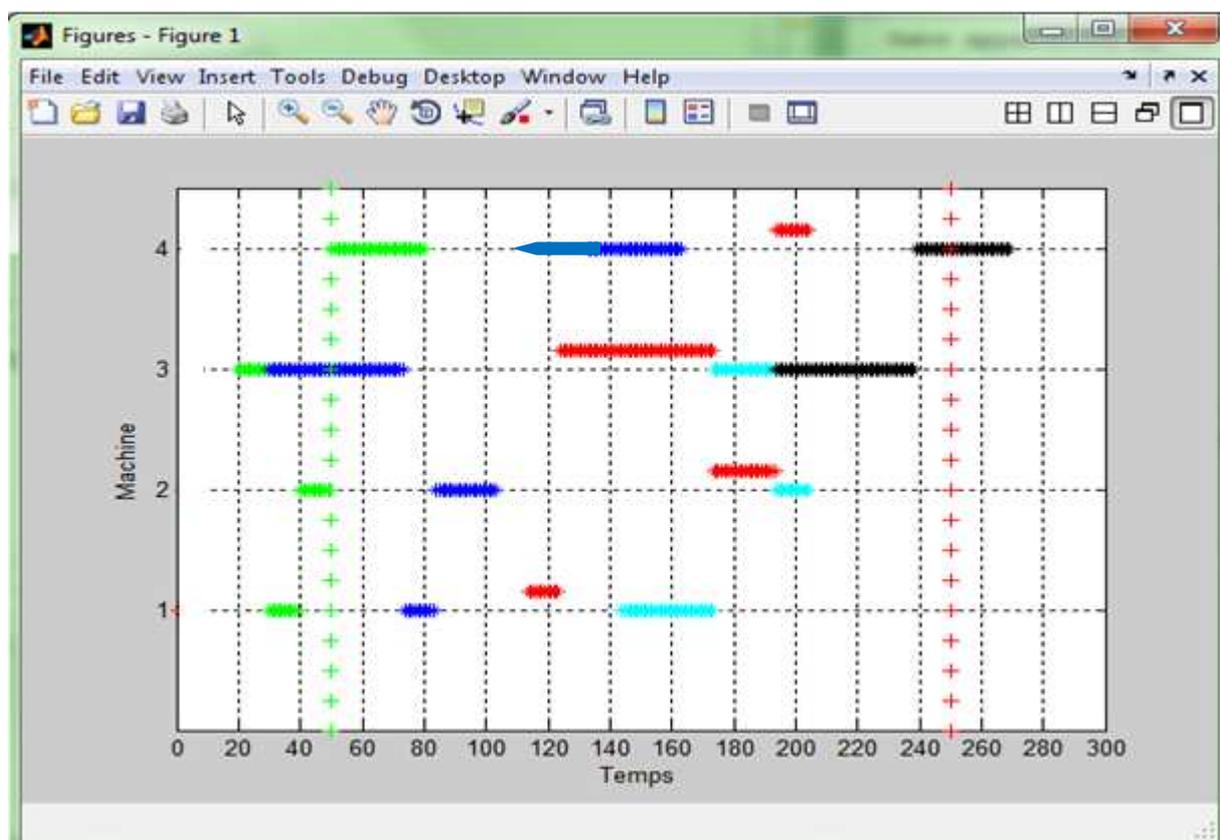
Deuxième itération



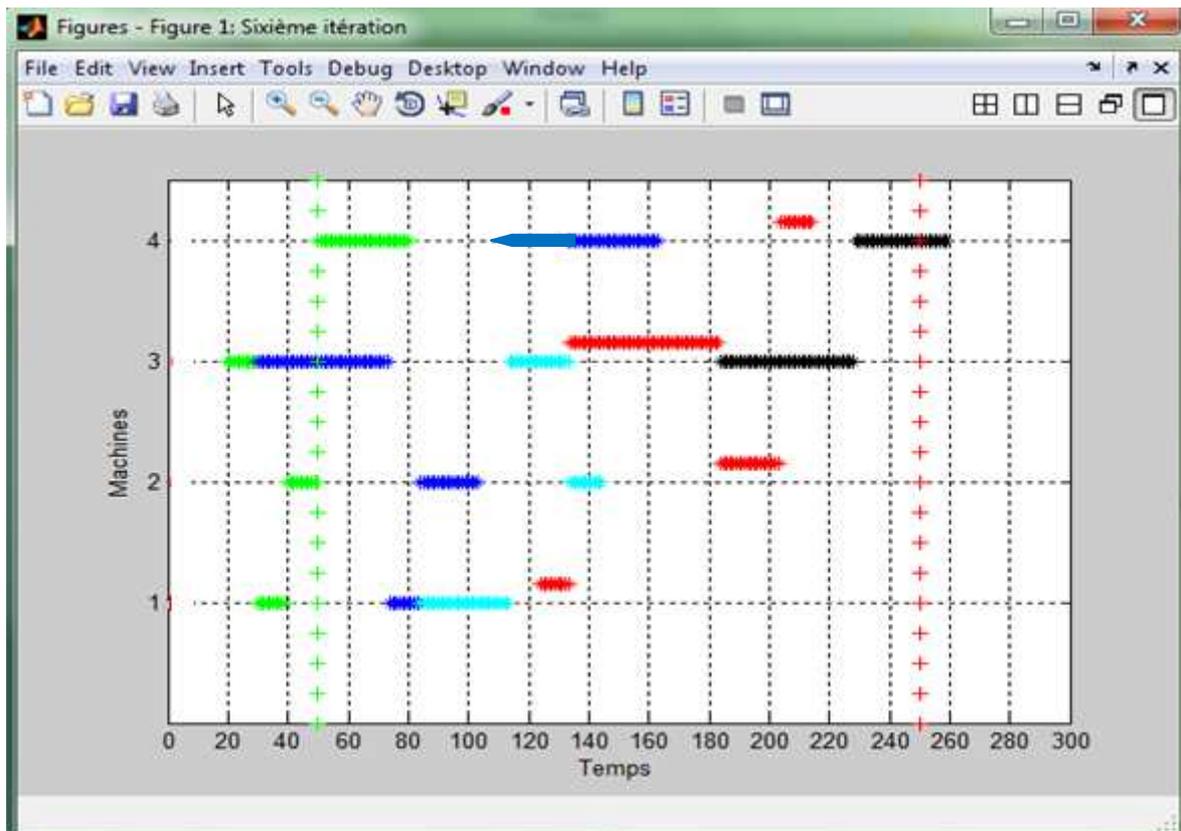
Troisième itération



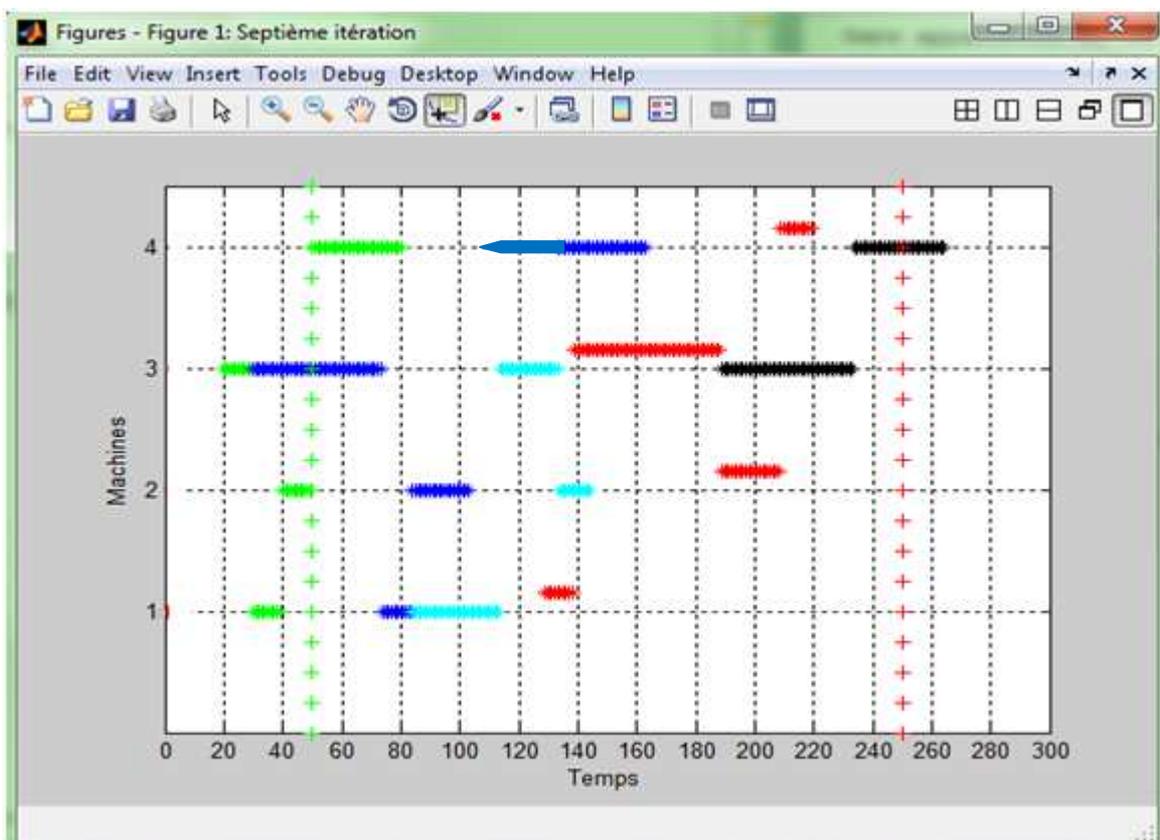
Quatrième itération



Cinquième itération



Sixième Itération



Septième Itération