



**HAL**  
open science

# Modélisation multi-contraintes d'un système de production flexible

José Eloundou

► **To cite this version:**

José Eloundou. Modélisation multi-contraintes d'un système de production flexible. Modélisation et simulation. INSA de Rouen, 2016. Français. NNT : 2016ISAM0012 . tel-01437152

**HAL Id: tel-01437152**

**<https://theses.hal.science/tel-01437152v1>**

Submitted on 17 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Normandie Université

## THÈSE

**Pour obtenir le diplôme de doctorat**

**Spécialité Informatique**

**Préparée au sein de l'INSA de Rouen**

## **Modélisation Multi-contraintes d'un Système de Production Flexible.**

**Présentée et soutenue par  
José ELOUNDOU**

**Thèse soutenue publiquement le 11 juillet 2016  
devant le jury composé de**

Pr. Djamila OUELHADJ	Professeure à l'université de Portsmouth	Rapporteur
Dr. Sofiene DELLAGI	Maître de conférences, HDR de l'université de Lorraine	Rapporteur
Pr. Jean Marc THIRIET	Professeur de l'Université de Grenoble Alpes	Examineur
Pr. Damien TRENTESEAUX	Professeur à l'Université de Valenciennes	Examineur
Pr. Abdelaziz BENSRAIR	Professeur à l'INSA de Rouen	Directeur de thèse
Dr. David BAUDRY	Enseignant-Chercheur-HDR au CESI de ROUEN	Codirecteur de thèse
Dr. M'hammed SAHNOUN	Enseignant-chercheur au CESI de Rouen	Encadrant de thèse

**Thèse dirigée par A. BENSRAIR(Laboratoire LITIS-INSA Rouen) et D.BAUDRY(Laboratoire LINEACT -CESI)**



# Table des matières

Table des figures	12
Liste des tableaux	15
Dédicaces	17
Remerciements	19
Résumé	21
Abstract	23
Introduction générale	25
<b>1 Etat de l'art</b>	<b>29</b>
1.1 Introduction . . . . .	29
1.2 Systèmes de production : généralités et classification . . . . .	30
1.2.1 Définitions . . . . .	30
1.2.2 Les éléments physiques d'un systèmes de production . . . . .	31
1.2.2.1 Les postes de travail . . . . .	32
1.2.2.2 Les produits et les Matières premières . . . . .	33
1.2.2.3 Les dispositifs de stockage . . . . .	33
1.2.2.4 Les moyens de transport et de manutention . . . . .	34
1.2.3 Classification des systèmes de production . . . . .	34
1.2.3.1 Classification en fonction de l'importance des séries et de la répétitivité . . . . .	35

1.2.3.2	Classification en fonction de la nature des flux	35
1.2.3.3	Classification en fonction de la Configuration	36
1.2.3.4	La configuration en ligne de production	36
1.2.3.5	La configuration en îlots ou en cellules de production	37
1.2.3.6	La classification en fonction de la flexibilité	37
1.3	La Flexibilité dans un système de production	41
1.3.1	La flexibilité des machines	41
1.3.2	La flexibilité des outils de manutention	42
1.3.3	La flexibilité des opérations	44
1.3.4	La flexibilité des processus de fabrication	44
1.3.5	La flexibilité du produit	44
1.3.6	La flexibilité d'acheminement des produits	44
1.3.7	La flexibilité des volumes de produits	45
1.3.8	La flexibilité d'extension	45
1.3.9	La flexibilité des programmes de contrôle	46
1.3.10	La flexibilité de la production	46
1.3.11	La flexibilité face au marché	46
1.4	Modélisation et simulation des systèmes de production	47
1.4.1	Modèles pour les systèmes de production	47
1.4.1.1	Les Modèles Mathématiques	48
1.4.1.2	Les modèles basés sur la théorie des graphes	49
1.4.2	Les RdP et la gestion de production	52
1.5	Les Réseaux de Pétri colorés Hiérarchiques	55
1.5.1	Les réseaux de Pétri Colorés	55
1.5.2	Les RdPC Modulaires et Hiérarchiques	56
1.5.3	La simulation des systèmes de production	59
1.5.4	Les plateformes de simulation des réseaux de pétri	61
1.5.4.1	Présentation de CPN Tools	63
1.5.4.2	Les jetons	67

1.5.4.3	Définition des variables et des fonctions . . . . .	68
1.5.4.4	La création des arcs . . . . .	68
1.5.4.5	Création d'une transition . . . . .	70
1.6	Conclusion . . . . .	72
<b>2</b>	<b>Modélisation et Validation</b>	<b>73</b>
2.1	Introduction . . . . .	73
2.2	Modèles objets d'un atelier de production flexible . . . . .	74
2.2.1	La modélisation des produits . . . . .	76
2.2.2	La modélisation d'une opération . . . . .	77
2.2.3	La modélisation des stocks . . . . .	79
2.2.4	La modélisation des machines . . . . .	80
2.2.5	La Modélisation des ressources de transport . . . . .	81
2.3	Modèles structurels et RdPC Hiérarchiques . . . . .	83
2.3.1	La modélisation d'un outil . . . . .	84
2.3.2	La modélisation des opérations machine . . . . .	84
2.3.3	La Modélisation des produits . . . . .	85
2.3.4	La modélisation d'un stock . . . . .	86
2.3.5	La modélisation des ressources de transport . . . . .	87
2.3.6	La modélisation d'une machine flexible . . . . .	87
2.3.6.1	Description des places du modèle de machine . . . . .	90
2.3.6.2	Les transitions du RdP Modulaire . . . . .	91
2.4	Modélisation du transport et réseaux de pétri . . . . .	97
2.4.1	Modélisation de l'aide à la décision . . . . .	97
2.4.1.1	Conditions de franchissabilité de la transition <b>T1</b> . . . . .	100
2.4.1.2	Définition des fonctions des arcs . . . . .	102
2.4.2	Modélisation du déplacement des produits . . . . .	103
2.4.2.1	Description des places . . . . .	104
2.4.2.2	Description des transitions . . . . .	104
2.4.2.3	Description des fonctions des arcs . . . . .	104

2.5	Validation des modèles développés . . . . .	105
2.5.1	Définition du problème . . . . .	105
2.5.2	La présentation de l'atelier de production didactique du CESI . . .	105
2.5.2.1	Le produit . . . . .	105
2.5.2.2	Les postes de travail . . . . .	106
2.5.2.3	Les Etapes d'assemblage d'un téléphone portable . . . . .	109
2.5.2.4	Les stocks et le transport . . . . .	111
2.5.2.5	La taille des stocks tampons . . . . .	112
2.5.2.6	Gammes de production . . . . .	113
2.5.2.7	Détermination de la durée des opérations pour les postes de travail déterministes . . . . .	113
2.5.2.8	Détermination de la durée des opérations pour les postes <i>Robot1</i> et <i>Robot2</i> . . . . .	114
2.5.3	Modélisation et simulation de la ligne de production à l'aide des RdPC . . . . .	117
2.5.3.1	Modélisation des postes de travail . . . . .	118
2.5.3.2	Modélisation du transport . . . . .	118
2.5.3.3	Modèle global de la ligne de production . . . . .	119
2.5.3.4	Validation des modèles par la ligne de production. . . . .	121
2.5.3.5	Simulation de la ligne de production . . . . .	122
2.6	Conclusion . . . . .	123
<b>3</b>	<b>Évaluation et aide à la décision</b>	<b>125</b>
3.1	Introduction . . . . .	125
3.2	Problème de transport . . . . .	126
3.2.1	Définition et contraintes . . . . .	126
3.2.2	Description du problème du transport dans un atelier de production	128
3.3	Méthodologie d'aide à la décision . . . . .	129
3.3.1	Objectifs et étapes méthodologiques . . . . .	129
3.3.2	Liste des actions potentielles pour l'aide à la décision dans la gestion du transport . . . . .	130

3.3.3	Les critères d'évaluation des activités de transport . . . . .	131
3.3.4	Établissement de la Table d'Évaluation Instantanée de l'Activité de transport . . . . .	131
3.3.4.1	La définition des fonctions d'affectation des ressources . . . . .	132
3.3.5	La Présentation de la table d'évaluation des activités de transport . . . . .	133
3.4	La Flexibilité de Routage (FR) . . . . .	134
3.4.1	Méthodologie d'évaluation de la Flexibilité de Routage . . . . .	137
3.5	Présentation de la stratégie de décision . . . . .	140
3.5.1	L'algorithme de gestion statique du transport . . . . .	141
3.5.1.1	l'exécution de l'algorithme . . . . .	143
3.5.1.2	Analyse de l'algorithme statique de gestion du transport . . . . .	145
3.5.2	Algorithme de gestion dynamique de l'activité de transport . . . . .	145
3.5.2.1	Excécution de l'algorithme . . . . .	149
3.5.2.2	Discussion sur l'algorithme de gestion dynamique . . . . .	149
3.6	Etude de cas . . . . .	150
3.6.1	Présentation du cas d'étude . . . . .	150
3.6.1.1	Organisation spatiale de l'atelier de production . . . . .	151
3.6.1.2	Les caractéristiques des machines . . . . .	151
3.6.1.3	Gamme de production . . . . .	152
3.6.1.4	Description de la production . . . . .	152
3.6.1.5	Critères d'évaluation des opérations de transport . . . . .	153
3.6.2	Modélisation de l'atelier de production . . . . .	153
3.6.3	Simulations et résultats . . . . .	155
3.6.3.1	Les paramètres du modèle de simulation . . . . .	155
3.6.3.2	Paramétrage de la table d'évaluation . . . . .	156
3.6.3.3	Déroulement de la simulation . . . . .	157
3.6.4	Analyse de l'impact de l'ordonnancement dynamique . . . . .	158
3.6.4.1	La situation A-F (Assigné - FIFO) . . . . .	158



3.6.4.2	La situation A-NF (Assigné - Non FIFO)	160
3.6.4.3	La situation NA-F (Non Assigné - FIFO)	161
3.6.4.4	La situation NA-NF (Non Assigné - Non FIFO)	162
3.6.5	Analyse des indicateurs de performances du système par type d'atelier de production	164
3.6.5.1	Analyse de la distance totale parcourue par la ressource de transport	164
3.6.5.2	Analyse du makespan	165
3.6.5.3	Stocks en entrée des machines	167
3.6.5.4	Stocks en sortie des machines	168
3.6.5.5	Analyse de La flexibilité de routage	171
3.7	Conclusion	172
<b>Conclusion générale</b>		<b>175</b>
<b>Annexes</b>		<b>180</b>
3.8	Fonctions du Réseau de Pétri Modulaire	180
3.8.1	Fonction <i>garde1</i>	180
3.8.2	Fonction <i>ReponseStock</i>	180
3.8.3	fonction <i>PreparationMachine</i>	181
3.8.4	fonction <i>ModificationInfoMachine</i>	181
3.8.5	fonction <i>CheckMachine</i>	181
3.8.6	fonction <i>EntreeProduitMachine</i>	181
3.8.7	fonction <i>garde2</i>	182
3.8.8	fonction <i>liberationMachine</i>	182
3.8.9	fonction <i>ModificationInfoMachine</i>	182
3.8.10	fonction <i>Ajouter_produit_stock</i>	182
3.9	Fonction <b>r1</b>	182
<b>A Autres fonctions développées</b>		<b>183</b>

**Bibliographie**

**201**

# Table des figures

1.1	Description globale d'un système de production manufacturier . . . . .	31
1.2	Configuration en ligne de production . . . . .	36
1.3	Schéma d'un atelier de production en îlots de production . . . . .	37
1.4	Illustration de la flexibilité des machines . . . . .	41
1.5	Illustration de la flexibilité des moyens de manutention : deux situations sont présentées. La situation A avec un transporteur A et la situation B avec un transport B. . . . .	43
1.6	Un exemple de Réseau de Pétri . . . . .	50
1.7	Exemple d'un RdP coloré pour illustrer les RdPCM . . . . .	57
1.8	Exemple d'un RdP coloré pour illustrer les RdPCM . . . . .	57
1.9	le RdPCM obtenu à partir du RdPC de la figure 1.7 . . . . .	58
1.10	Illustration d'une place sur CPN TOOLS . . . . .	65
1.11	Nombre de jetons d'une place sous CPN TOOLS . . . . .	67
1.12	Jetons de types entiers et de type chaîne de caractère sous CPN TOOLS . . . . .	68
1.13	Description d'un arc sous CPN TOOLS . . . . .	69
1.14	illustration d'un arc pondéré par une fonction sous CPN TOOLS . . . . .	69
1.15	Description d'un arc reliant une transition à une place aval et pondéré par une variable. . . . .	69
1.16	illustration d'un arc pondéré par une fonction sous CPN TOOLS . . . . .	70
1.17	Illustration des règles d'utilisation des variables au niveau des arcs sortant des transitions lorsque la place est pondéré par une fonction . . . . .	70
1.18	Illustration de la condition d'utilisation des variables sur les arcs sortant d'une transition . . . . .	70
1.19	Les caractéristiques d'une transition sur la plateforme CPN TOOLS . . . . .	71

---

1.20	Illustration de l'utilisation de la fonction de transfert d'une transition sous CPN TOOLS . . . . .	71
2.1	Notre Démarche de travail . . . . .	74
2.2	Diagramme des classes d'un atelier de production flexible . . . . .	75
2.3	Diagramme de composition d'un atelier de production flexible . . . . .	76
2.4	Classe produit . . . . .	77
2.5	Classe produit avec des variables pour les problèmes d'ordonnancement . . . . .	78
2.6	classe outil . . . . .	78
2.7	Classe opération . . . . .	78
2.8	Classe STOCK . . . . .	79
2.9	classe machine . . . . .	81
2.10	États de la machine . . . . .	82
2.11	Classe Transport . . . . .	83
2.12	Classe d'association d'une opération de transport . . . . .	83
2.13	classe machine . . . . .	88
2.14	Représentation d'un RdP modulaire d'une machine. . . . .	89
2.15	Réseau de Pétri Coloré Modulaire d'une machine . . . . .	90
2.16	Réseau de pétri Coloré modulaire de l'activité de choix de la machine d'arrivée et de départ par la ressource de transport dans un atelier contenant trois machines, un stock d'entrée et un stock de sortie des produits . . . . .	99
2.18	Modélisation de l'opération de transport dans un atelier de production à quatre machines possédant chacune un stock d'entrée et un stock de sortie. L'atelier de production lui même possède une seule zone de sortie des produits. . . . .	103
2.19	Composition du produit . . . . .	106
2.20	L'atelier d'assemblage flexible du laboratoire IRISE du CESI . . . . .	106
2.21	Le <i>ROBOT1</i> de déchargement et chargement des stocks d'entrée de et sortie respectivement. . . . .	107
2.22	Les différents composants du poste de travail <i>Robot2</i> . . . . .	108
2.23	Le poste de travail <i>DéposeCouvercle</i> . . . . .	108
2.24	Le poste de travail <i>Presse</i> . . . . .	109
2.25	Le poste de travail <i>Perceuse</i> . . . . .	109

2.26	Le poste de travail <i>Four</i> . . . . .	110
2.27	Schéma du fonctionnement de l'atelier flexible , . . . . .	111
2.28	Durées des opérations de prise de stock d'entrée réalisées par le Robot 1 . . . . .	114
2.29	Durée de l'opération de sortie des produits . . . . .	115
2.30	Approximation par la loi normal ou log-normal du temps opératoire du <i>Robot1</i> . . . . .	116
2.31	Approximation par la loi log-normal ou gamma du temps opératoire du <i>Robot2</i> . . . . .	117
2.32	Modèle objet d'un poste de travail . . . . .	117
2.33	Modèle objet d'un stock . . . . .	118
2.34	RdPM du transport (convoyeur) de la ligne de production du CESI . . . . .	119
2.35	Modèle Hiérarchique de la ligne de production du CESI de Rouen . . . . .	120
3.1	Schéma illustrant le problème de transport dans une unité de production . . . . .	126
3.2	Algorithme de gestion statique du transport . . . . .	142
3.3	Algorithme de gestion dynamique des opérations de transport . . . . .	148
3.4	Illustration de la configuration de l'atelier de production . . . . .	151
3.5	Exemple d'un atelier à cheminement multiples à quatre machines . . . . .	154
3.6	Situation AF . . . . .	159
3.7	Comparaison graphique des différentes stratégie d'ordonnancement dans le cas d'un atelier type A-NF . . . . .	161
3.8	Comparaison graphique des différentes stratégie d'ordonnancement dans le cas d'un atelier type NA-F . . . . .	162
3.9	Comparaison graphique des différentes stratégie d'ordonnancement dans le cas d'un atelier type NA-NF . . . . .	163
3.10	Distance parcourue par la ressource de transport [m] . . . . .	164
3.11	Makespan des différents stratégie d'ordonnancement pour toutes les situa- tion considérés . . . . .	166
3.12	Charge moyenne des stock en entrée . . . . .	168
3.13	Charge moyenne des stocks en sortie des machines . . . . .	169
3.14	Variation de la flexibilité de routage selon l'ordonnancement et le type d'atelier . . . . .	171



# Liste des tableaux

1.1	Tableau comparatif type de systèmes de production [1]	40
1.2	tab :ReviewRDP	53
1.2	tab :ReviewRDP	54
1.3	Les étapes du processus de simulation des systèmes de production [2]	61
1.4	Liste de plateformes de modélisation et de simulation des RdP	62
1.5	structure de données simples sur CPN TOOLS	65
2.1	Durée des opérations des machines possédant des temps de process constants	113
2.2	Les transitions du modèle de la ligne de production	121
2.3	Durée des opérations du produit dans le cas 1	122
3.1	Table d'évaluation de l'activité de transport	133
3.2	Table d'évaluation initiale	144
3.3	Etape 1 : Table obtenue après avoir effectué un tri sélectif sur la suivant le critère de la première colonne de la table 3.2	144
3.4	Etape 3 de l'algorithme : tri de la table d'évaluation courante suivant le critère de la troisième colonne	144
3.5	Table d'évaluation normalisée de l'activité de transport	146
3.6	Table d'évaluation normalisée de l'activité de transport	147
3.7	Position des machines dans l'atelier de production	152
3.8	temps de montage des outils	152
3.9	plan de production des produits	152
3.10	Gamme de production	156
3.11	Valeurs des indicateurs de performances pour un atelier A-F	159

---

3.12 Valeurs des indicateurs de performances pour un atelier A-NF . . . . .	160
3.13 Valeurs des indicateurs de performances pour un atelier NA-F . . . . .	162
3.14 Valeurs des indicateurs de performances pour un atelier NA-NF . . . . .	163
3.15 Distance parcourue par la ressource de transport . . . . .	164
3.16 Temps total d'écoulement des produits (Makespan) . . . . .	166
3.17 Taille moyenne des stocks en entrée . . . . .	167
3.18 Charge moyenne totale des stocks en sortie des machines . . . . .	168
3.19 Stock des machines dans le cas Od1-A-F . . . . .	170
3.20 Entropie du système de production . . . . .	171





# Dédicaces

A mon père, ma mère, mes frères et soeurs, merci.

A Adrienne et Christian, pour vous le voyage s'est terminé, mais pour moi le périple continue. Je pense à vous et je vous aime.



# Remerciements

Je tiens à remercier l'ex-région Haute Normandie aujourd'hui Région Normandie pour m'avoir accordé cette bourse de recherche. Je tiens à remercier mes directeurs de thèse, le Pr. Abdelaziz BENSRAHAI et le Dr. David BAUDRY pour leurs conseils, leur soutien et leurs idées. Je tiens aussi à exprimer ma profonde reconnaissance au Dr. M'hammed SAHNOUN pour son encadrement tout au long de ces deux dernières années. Je pense que ce travail aurait difficilement pu voir le jour sans son aide, et son travail acharné.

Je remercie également le Directeur de la Recherche National du CESI le Dr. Belhacène MAZARI et le Directeur du département Recherche du CESI de Rouen Mme Anne LOUIS pour la confiance qu'ils ont mis en moi. Je remercie aussi Mme Christine TSAFACK, Assistante du Département Recherche du CESI de Rouen pour l'aide qu'elle nous a apporté dans l'édition et l'impression de ce document.

Je tiens à exprimer ma gratitude au Directeur du LITIS Thierry PAQUET et aux membres de son équipe. Un remerciement particulier à Brigitte DIARRA pour avoir assuré la coordination administrative entre l'INSA de Rouen et le CESI de Rouen dans le cadre de la co-tutelle de thèse.

Enfin je ne saurais oublier mes collègues du Département Recherche du CESI de Rouen, Dr. Mourad MESSADIAA, le Dr. Fabrice DUVAL, Mohamed-Amin BENATIA, Sara MAHDIHKAH, Elodie PILLON, Vincent HAVARD pour leur bonne humeur et leur accueil chaleureux.

Je remercie également Mme NGUIDJOL Jeannette pour avoir relu mon manuscrit et diminué le nombre de fautes d'orthographe.



## Résumé

Les ateliers flexibles représentent pour les industriels un réel atout face à la concurrence et à l'imprévisibilité du marché. La mise en place et la gestion de ces catégories d'ateliers de production est souvent complexe, difficile et coûteuse. Ce travail de thèse, propose un outil d'aide à la décision permettant d'accompagner les industriels dans l'acquisition et la gestion d'ateliers flexibles de production.

Cet outil s'appuie sur un modèle de simulation basé sur les Réseaux de Pétri Colorés Hiérarchiques. En effet, la simulation permet d'observer les variations des paramètres du système et de déterminer ses critères de performance. Les objectifs de cet outil sont de permettre la modélisation, la simulation et l'évaluation des systèmes de production en prenant en compte plusieurs contraintes rencontrées dans les systèmes industriels. Nous avons utilisé ce simulateur pour tester et valider une solution au problème d'acheminement des produits au cours de la production.

Le choix des Réseaux de Pétri(RdP), au détriment des modèles analytiques, se justifie par le fait qu'ils offrent une représentation graphique permettant une description des processus plus simplifiée que des modèles analytiques. Au-delà de la modélisation et la simulation, les RdP permettent de vérifier les blocages d'un système.

Parmi les modèles que nous avons développés nous pouvons citer : les modèles des machines, des stocks, des ressources de transport et des produits dans un système de production flexible. Nous avons insisté sur la modélisation des produits, des machines et du système de transport qui représentent le cœur des problématiques de flexibilité. Nous avons validé ces modèles sur un cas réel à savoir la chaîne de production du CESI de Rouen. Ensuite pour résoudre le problème du transport, nous avons proposé et déployé une méthode d'aide à la décision basée sur l'entropie et le classement de règles de dispatching. Cette méthode permet de définir, en fonction de l'état actuel, un ordre de priorité entre les règles de dispatching. Nous avons comparé cette nouvelle méthode à la méthode classique où l'importance des règles de priorité est définie préalablement, avant la phase de production. Notre proposition a donné des résultats intéressants que nous avons exposés et discutés.

**Mots clés :** Réseaux de Pétri, modélisation, simulation, systèmes de production flexible.



## Abstract

Flexible Manufacturing Systems are real opportunity for industrials to face the competition and the unpredictability of the market. The management implementation of this kind of manufacturing systems are complex difficult and expensive.

In this thesis work, we propose a decision making tool for managing Flexible Manufacturing Systems. The latter is based on a simulation model with Hierarchical Colored Petri Nets. The simulation allows the observation of the parameters variation of a system, and the determination of performance criteria. The objectives of this tool are to enable the modeling, simulation and evaluation of production systems taking into account most of the constraints encountered in industrial systems. We used this simulator to test and validate a solution to the problem of routing products during production. The choice of Petri nets (PN) to the detriment of analytical models is justified by the fact that Petri Nets with its graphical representation allows a simplified description of processes as analytical models ; Petri Nets can be used for simulation ; and with simulation we can predict the behaviour of a systems during a production phase. Petri Nets allow, beyond modeling and simulation to prevent the dead locks of systems. These are the reasons why we use Petri Nets to develop our models.

Among the models that we have developed we can mention : machine, stocks, transportation resources and products models. We insisted on product and machine modelling which are in the center of flexibility issues. In order to study the routing problem, We defined models to represent the transport activity. We validated these models on a real case namely the production chain of CESI of Rouen. Then to solve the routing problem, we developed a decision making method based on entropy and dispatching rules. This method allows to modify the dispatching rules weight in function of the state of the manufacturing systems during the production(size of stocks ; machine failure). We compared this method with an another one where the weight of dispatching rules are unchanged during the production phase. We obtained interesting result that we have presented and discussed.

**Keywords :** Petri Nets, Modelling, Simulation, Manufacturing systems.





# Introduction générale

La personnalisation des produits manufacturés, la concurrence internationale, les normes réglementaires nationales et internationales, le coût de l'énergie, la variation et la diversification de la demande pour ne citer que celles-là, sont des contraintes externes auxquelles les entreprises du secteur industriel doivent faire face. En plus de cela il faut rajouter les contraintes internes telles que : délai de livraison, le temps de production, les transports et la disponibilité des ressources. Face à ces aléas, les entreprises doivent faire évoluer leur outil de production en leur apportant de la flexibilité.

Cette flexibilité implique des coûts importants dû aux prix des machines à commande numérique, des moyens de transport de plus en plus en robotisés et des systèmes de gestion. Ainsi l'adoption des systèmes de production flexibles nécessite des études approfondies avant leur mise en application, tant sur le plan économique que sur le plan des nouvelles performances qu'elles pourraient apporter à l'entreprise. Pour cela les entreprises ont besoin d'outils afin de pouvoir évaluer les changements qu'elles pourraient subir de manière involontaire (variation du niveau de commande, pannes, ...) mais aussi celles qu'elles provoquent (introduction d'un nouveau produit, acquisition d'une nouvelle machine, etc). Les organes de décision ont donc besoin d'outils de prédiction permettant d'observer l'impact que pourrait provoquer une de leur décision ou une de leur action sur les performances de leur appareil de production. Ces outils de prédiction sont le plus souvent basés sur des modèles de simulation et des méthodes d'aide à la décision.

Cette nécessité d'apporter des réponses rapides aux entreprises lorsque celles-ci se trouvent dans des processus d'adoption des systèmes de production flexibles a poussé plusieurs équipes de recherche à développer des travaux qui ont pour but de proposer aux Petites et Moyennes Entreprises des outils d'aide à la conception et à la décision leurs permettant d'adopter facilement des systèmes de production flexibles.

Afin de répondre aux défis de l'industrie du futur, le CESI avec son laboratoire LI-NEACT développe un "démonstrateur industrie du futur", et mène des travaux de recherche dans le cadre de la performance industrielle. Pour cela des outils d'aide à la décision et la conception des ateliers flexibles sont développés dans l'objectif de :

- Prendre en compte la flexibilité des produits et des machines dans la modélisation des systèmes de production.
- Intégrer les contraintes spatiales (position des machines, taille de l'atelier, position des stocks, ...) et temporelles (temps de transport, temps de chargement, temps de

préparation des machines, ...) dans les ateliers de production.

- Assurer le pilotage du système de production et la gestion des conflits dûs au partage des ressources.
- Simuler le comportement d'un système de production existant ou d'une future implantation afin d'observer des indicateurs de performance tels que : les temps totaux de production, la flexibilité d'acheminement des ressources ou encore les distances parcourues par les transporteurs.
- Permettre à l'utilisateur de définir sa propre politique d'ordonnancement.
- ...

Pour arriver à mettre en place cet outil d'aide à la conception, nous avons abordé dans ce travail les problématiques suivantes :

- La modélisation et la simulation des systèmes de production flexibles,
- La mesure de la flexibilité de routage dans les ateliers de production flexible,
- La gestion des ressources de transport et d'acheminement des produits dans un atelier de production flexible.

Ainsi, ce manuscrit est structuré en trois chapitres qui englobent plusieurs sections chacun :

Le premier chapitre est un chapitre d'état de l'art qui comporte quatre sections aux travers desquelles nous abordons des généralités sur les système de production, la notion de flexibilité, les méthodes de modélisation et simulation et enfin les réseaux de pétri. Dans la première section, nous abordons les généralités sur les systèmes de production. Nous présentons plusieurs classifications de ceux-ci en fonction de différents critères. Nous insistons dans cette classification sur le critère de flexibilité qui permet de distinguer trois types de systèmes de production à savoir : les systèmes de production dédiés, les systèmes de production flexibles et les systèmes de production reconfigurables. La section 2 aborde les différents types de flexibilité dans les systèmes de production flexibles. La section 3, présente et compare les modèles couramment utilisés dans les travaux de recherche pour appréhender les problématiques des systèmes de production. Cette comparaison nous a permis de faire émerger les réseaux de Pétri colorés hiérarchiques comme outil de modélisation. C'est pour cela que la quatrième section présente les Réseaux de Pétri colorés hiérarchiques. Ensuite nous abordons brièvement la simulation pour l'étude des systèmes de production. Pour terminer, nous dressons une liste des outils de simulation des réseaux de Pétri et une introduction à l'outil CPN TOOLS utilisé dans le cadre de notre activité de modélisation et de simulation.

Le deuxième chapitre nous présentons les modèles développés afin de construire notre outil d'aide à la conception des ateliers flexibles de production. Ce chapitre comprend quatre sections. La première section présente une modélisation objet des entités d'un système de production. Il représente les modèles des machines flexibles, de stocks, des

produits et des ressources de transport. Dans un premier temps nous avons utilisé le formalisme UML pour modéliser notre système de manière simple. La deuxième section présente le passage de l'UML vers les Réseaux de Pétri colorés Hiérarchiques. La troisième section présente le modèle de transport dans un atelier de production flexible sous le formalisme des Réseaux de Pétri Hiérarchiques. La dernière section de ce chapitre présente l'étude réalisée sur la ligne de production du CESI de Rouen. L'objectif de celle-ci étant de vérifier la validité des modèles que nous avons développés dans un cas réel.

Le troisième chapitre aborde la problématique de gestion dynamique du transport et en particulier le problème d'acheminement des produits dans un atelier de production. Il est présenté en deux sections. La première section aborde deux stratégies différentes pour répondre au problème d'acheminement des produits en temps réel dans un atelier de production. L'une de ces stratégies est statique et l'autre dynamique. Cette dernière s'appuie sur une méthode d'aide à la décision basée sur une table d'évaluation des opérations de transports dans l'atelier ; et sur une mesure de l'entropie des critères de décision. Dans la deuxième section nous avons implémenté ces deux méthodes dans notre simulateur basé sur les réseaux de Pétri afin d'en étudier les performances sur un atelier de type JOB SHOP flexible composé de quatre machines et d'une ressource de transport.



# Chapitre 1

## Les systèmes de production flexibles et les réseaux de Pétri : Etat de l'art

### Introduction

Nous avons fait part dans l'introduction générale de notre intention de proposer un modèle de simulation permettant d'évaluer les performances d'un système de production afin d'apporter des éléments aux prises de décision des gestionnaires de ces derniers. Avant de proposer un tel outil il nous paraît important de présenter les systèmes de production, dans leur complexité et dans leur diversité afin de circonscrire notre cadre de travail. C'est l'un des objectifs de ce chapitre. L'autre objectif de ce chapitre est de présenter les Réseaux de Pétri colorés Hiérarchiques et de justifier de leur utilisation pour répondre à notre problématique. Nous avons donc structuré ce chapitre de la manière suivante :

- En section 1.2, nous présentons des généralités sur les systèmes de production (définition, typologie, classification) ;
- En section 1.3, nous nous intéressons à la notion de flexibilité dans un système de production ;
- Dans la section 1.4, nous abordons les problématiques de modélisation (sous-section 1.4.1). Dans cette section nous justifions les raisons pour lesquelles nous utilisons les Réseaux de Pétri Colorés Hiérarchiques comme outil de modélisation et nous présentons un état de l'art sur l'utilisation des réseaux de Pétri pour la gestion de production.
- Par la suite nous avons consacré la section 1.5 à la présentation formelle des réseaux de Pétri Colorés Hiérarchiques. Nous terminons ce chapitre en listant quelques plateformes de modélisation et de simulation des Réseaux de Pétri et en présentant la plateforme de simulation CPN TOOLS que nous allons utiliser dans le cadre de notre travail (sous-section 1.5.4).

# Systèmes de production : généralités et classification

## Définitions

Un système de production est un ensemble de ressources qui concourent à la fabrication d'un bien [3]. La fabrication d'un bien se fait par le biais de diverses opérations telles que : des opérations manuelles, d'usinage et/ou d'assemblage. L'assemblage est l'action de fabriquer des objets par agrégation d'objets plus simples [4]. En plus de l'assemblage et de l'usinage, il faut aussi noter les activités de transport et de stockage. Un système de production est présenté par [3] comme une entité composée de :

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

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

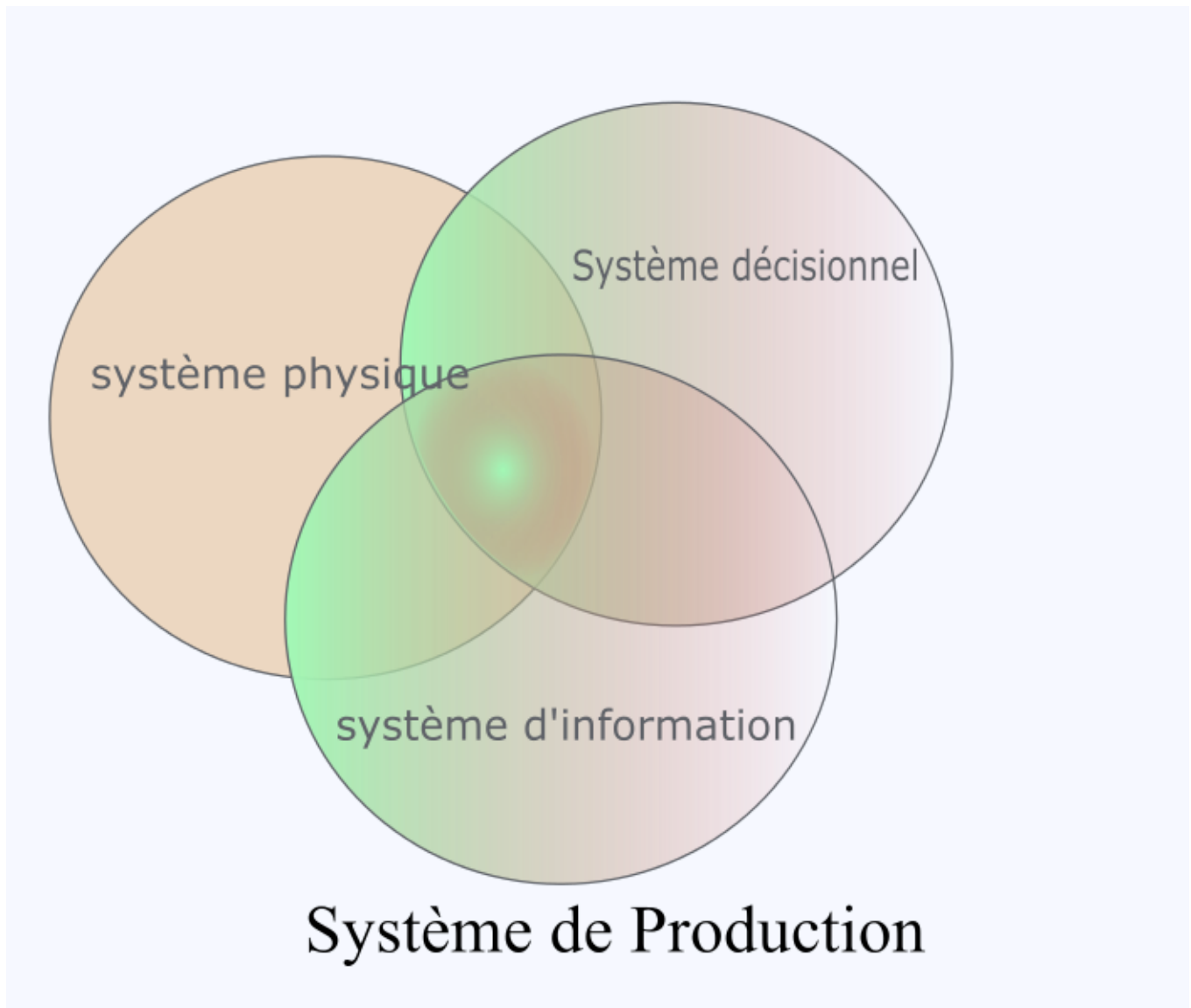


FIGURE 1.1 – Description globale d'un système de production manufacturier

### **Les éléments physiques d'un systèmes de production**

Les éléments physiques d'un système de production sont :

- Les postes de travail ;
- Les produits et matières premières ;
- Les dispositifs de stockage.



## Les postes de travail

D'après [5], un poste de travail est *une machine ou un endroit aménagé spécifiquement où peut être exécutée une opération donnée*. L'auteur rajoute aussi qu'un poste de travail se définit par son activité (fabrication, contrôle, manutention), par un aménagement et un outillage et enfin par une description de la compétence de l'opérateur. Les caractéristiques d'un poste de travail ou d'une machine sont : son activité, sa capacité et son temps d'exécution.

- La capacité d'un poste de travail : elle représente la quantité de produits traités de manière simultanée par ce dernier. Cette caractéristique permet de définir deux types de poste de travail les postes capacitaires et les postes non capacitaires [6]. Un poste est capacitair lorsque l'on peut augmenter dans une certaine mesure sa capacité. Par exemple un poste de travail où s'effectue une opération manuelle peut être considéré comme capacitair dans la mesure où on peut rajouter des personnes en renfort. Par contre un poste qui est composé d'une machine automatisée peut être considéré comme non capacitair si on ne peut pas augmenter le nombre de produits que la machine peut traiter de manière simultanée.
- La fonctionnalité : c'est l'ensemble des opérations que peut réaliser un poste de travail. Si ce dernier n'effectue qu'une seule opération, on dira qu'il a une fonction unique. C'est en général le cas des machines. Dans le cas où au niveau d'un poste de travail on peut réaliser plusieurs tâches on dira que le poste de travail ou la machine est multifonction. Il faut aussi préciser que le fait qu'un poste de travail soit multifonction ne veut pas forcément dire qu'il soit multitâche. En effet la plus part des postes de travail sont monotâches, on parle de contraintes disjonctives. Cependant il n'est pas exclu que plusieurs tâches soient effectuées sur un même produit de manière simultanée par des opérateurs différents et sur un même poste de travail.
- Les temps d'exécution : ce sont les durées des opérations effectuées par les postes de travail. Celles-ci dépendent de la nature du produit qui est traité et de celle de la ressource qui effectue l'opération. De cette manière, un robot aura une très faible variabilité dans ces temps d'exécution contrairement à un humain.
- La Modularité : C'est la capacité pour une machine de modifier sa structure physique afin de pouvoir effectuer d'autres opérations. La notion de modularité est mise en avant pour la première fois dans les travaux de [1] définissant la notion de systèmes de production reconfigurables que nous aborderons dans la partie 1.2.3.6.2.

Les postes de travail imposent aux systèmes de production un certain nombre de contraintes à savoir :

- Les contraintes de disponibilité : une machine qui ne peut traiter qu'un nombre de produit maximal à la fois, génère donc des contraintes de disponibilité.

- Les contraintes liées au multi-outillage et à la modularité : certains systèmes de production possèdent des machines outils, ou modulaires mais cette caractéristique engendre des temps de changement d'outillage souvent à l'origine de temps d'attente qui peuvent être longs si l'ordonnement des produits ne les intègre pas.

## Les produits et les Matières premières

Dans un système de production une matière première correspond à toute ressource consommable rentrant dans la confection d'un produit. Les matières premières sont de toutes origines : animale, végétale, minérale etc ... Les différentes manipulations que subissent les matières premières aboutissent ainsi à la fabrication d'un produit. On peut citer comme exemples l'assemblage, l'usinage, le traitement de surface ... Un produit peut se définir comme le résultat d'un ensemble d'opérations impliquant des ressources consommables(matières premières) et des ressources non consommables(postes de travail, moyens de manutention. Ces opérations mobilisent des machines et des moyens de transport pendant des durées de temps qui peuvent être déterministes ou non. Les opérations qui concourent à la fabrication d'un produit constituent la gamme de production de celui-ci. Les produits qui sont fabriqués dans un système de production génèrent des contraintes sur ce dernier. On peut citer à titre d'exemple :

- Les contraintes liées aux processus de fabrication : la fabrication d'un produit se fait par plusieurs étapes qui peuvent être liées par des contraintes de précédence<sup>1</sup>. Les contraintes de précédence des opérations sont à l'origine des problématiques de séquençement des opérations sur les lignes d'assemblage.
- Les contraintes liées à la commande : ce sont généralement les dates de fin au plus tard de fabrication des produits. Les contraintes liées à la commande couplées aux contraintes liées aux processus de fabrication génèrent des problèmes d'ordonnement.

## Les dispositifs de stockage

Nous appellerons dispositif de stockage tout espace ou toute zone d'un atelier de production où les matières ne subissent ni transformation ni déplacement pendant une durée de temps plus ou moins importante. Les stocks dans un système de production permettent de stocker de la matière première mais aussi de donner une certaine cadence à la production.

Lorsque qu'un dispositif de stockage a pour but de contenir des matières premières nécessaires à la fabrication d'un produit on parle de **stock d'approvisionnement**. Par contre lorsqu'il permet la rétention de produits en cours de production on parle **stock tampon**. Enfin lorsque le dispositif de stockage contient des produits finis et contrôlés on parle de **zone d'expédition**.

---

1. supposons que nous avons deux opérations  $O_1$  et  $O_2$  si l'opération  $O_1$  ne peut être réalisé sans que l'opération  $O_2$  ne soit achevée, on dit qu'il existe une contrainte de précédence entre  $O_1$  et  $O_2$ .

## Les moyens de transport et de manutention

Les moyens de transports dans un système de production ont pour objectif d'assurer le déplacement des matières premières des zones de stockage vers les zones de traitement (machines ou postes de travail), mais aussi d'assurer le déplacement des produits semi finis à travers les différents postes de travail. En fonction de la configuration et de l'organisation d'un système de production le transport peut être assuré par un convoyeur, un véhicule guidé, des transpalettes ou des personnes. Dans un système de production les moyens de transport influent sur la performance par deux facteurs principaux à savoir :

- La disponibilité des moyens de transport : dans un système de transport les ressources assurant le transport ne sont pas toujours en nombre suffisant pour répondre aux besoins de déplacement des ressources ou des produits, cela engendre alors des contraintes de disponibilité. Les problèmes de disponibilité des moyens de transport sont surtout très importants lorsque des contraintes de délais de commande sont à satisfaire ou lorsque les contraintes de limitation des stocks s'impose au système.
- La flexibilité : dans un système de production la flexibilité d'une ressource de transport est l'aptitude de cette dernière à assurer des déplacements à travers les différentes stations de travail du système. Cette aptitude rentre dans la définition de la flexibilité d'acheminement ou "routing flexibility" que nous définissons dans le paragraphe 1.3.6

Nous venons de définir dans cette partie qu'un système de production est un ensemble constitué de machines, de produits, de ressources de transport et de matières premières. Dans la prochaine section nous allons présenter comment les différents éléments d'un système de production s'articulent afin d'assurer la production de biens et de services de manière efficiente.

## Classification des systèmes de production

Dans l'organisation des systèmes de production il faut prendre en compte deux aspects : l'organisation spatiale de l'outil de production ou l'implantation et la gestion et le contrôle des flux. Cependant avant d'évoquer l'organisation des systèmes de production nous allons présenter une typologie des systèmes de production. Dans cette partie nous allons présenter l'architecture d'un système de production, et dans le même temps la configuration d'un système de production. L'objectif de cette partie est d'avoir une vision descendante de l'architecture d'un système de production. En effet on distingue plusieurs éléments dans l'architecture d'un système de production : les machines ou stations de travail, les cellules ou îlots de production et l'atelier de production. La machine constitue l'unité élémentaire dans un système de production ; cela suppose que l'on ne peut pas diviser l'action d'une machine. Une cellule de production est constituée d'une ou plusieurs machines qui sont reliées soit par un moyen de transport automatisé et continu ou alors par un opérateur qui assure le transport des produits entre les stations de travail.

Un atelier de production quant à lui est constitué d'un ensemble d'îlots de production. L'architecture d'un système de production dépend de la typologie de la production.

Dans [7] on identifie les systèmes de production en fonction de trois critères à savoir : le mode de production, l'importance des séries et la répétitivité de la nature des flux, et la relation client. Il existe aussi une classification des systèmes de production en fonction du niveau d'évolution de ce dernier par rapport à son environnement c'est à dire son niveau de flexibilité.

### **Classification en fonction de l'importance des séries et de la répétitivité**

En fonction de l'importance des séries et de la répétitivité on distingue :

- *La production unitaire* : elle est destinée à la conception d'un seul et unique produit. Une fois le produit conçu le système de production n'a plus de raison d'être. Ce système de production est le plus souvent utilisé dans le domaine du génie civil pour la conception d'ouvrage d'art.
- *La production en série (petite, moyenne, grande)* : elle est destinée à la production en plusieurs exemplaires d'un ou plusieurs produits. En fonction de la quantité de produits on parle de petite série (environ 100 exemplaires), moyenne série (1000 exemplaires) et de grande série (plus de 100 000 exemplaires)

### **Classification en fonction de la nature des flux**

*Un flux est tout déplacement d'éléments dans l'espace et dans le temps [8]* ; ainsi un non-flux est un état de stagnation (non déplacement dans l'espace) dans lequel se trouvent des éléments dans un système de production. En partant de cette définition on peut classifier les systèmes de production en fonction de la nature des flux lors de la production. En fonction de la nature des flux on distingue trois formes de production :

- Production à flux continu. En exemple, les industries de la sidérurgie ou de la chimie ou l'on manipule des fluides.
- Production à flux discontinus ou discrets : les principales industries manufacturières qui font de l'assemblage.
- Production par projet. On peut citer les industries du bâtiment, ou l'industrie aéronautique.

En fonction de la stratégie d'une entreprise vis à vis de sa clientèle on distingue :

- Production pour stock : dans ce cas de figure l'entreprise produits uniquement pour refaire ses stocks. Ici il n'y a pas en général les contraintes liées aux délais de livraison.

- Production à la commande : l'entreprise produit en fonction de la demande des clients.

La typologie des systèmes de production montre que la production de biens peut être influencée par le mode de production (petites séries, moyenne série etc.) Mais elle l'est aussi par la relation avec le client ou encore la nature des flux. Pour s'adapter à ces influences les systèmes de production présentent des configurations d'implantation qui peuvent être tout aussi variées.

### Classification en fonction de la Configuration

La configuration d'un système de production est l'agencement spatial des machines, des stocks et des ressources de transport. Il existe des configurations classiques telles que les configurations en ligne de production et en atelier de production. Il existe une confusion entre la notion d'**atelier de production** et de celle de **configuration en atelier de production**. En réalité la notion de configuration en atelier de production est utilisée par opposition aux autres configurations (configurations en ligne, en U). Cependant un atelier de production peut être constitué de plusieurs cellules de production possédant chacune des configurations différentes. Le prochain paragraphe donne un aperçu des configurations que l'on peut trouver dans un atelier de production.

#### La configuration en ligne de production

Une ligne de production est un ensemble de stations de travail (machines ou postes de travail) inter-connectées entre elles à l'aide d'un moyen de transfert mécanisé (généralement un convoyeur)[9]. Sur une ligne de production les postes de travail sont placés dans un ordre précis. La figure 1.2 illustre une ligne de production de quatre machines.

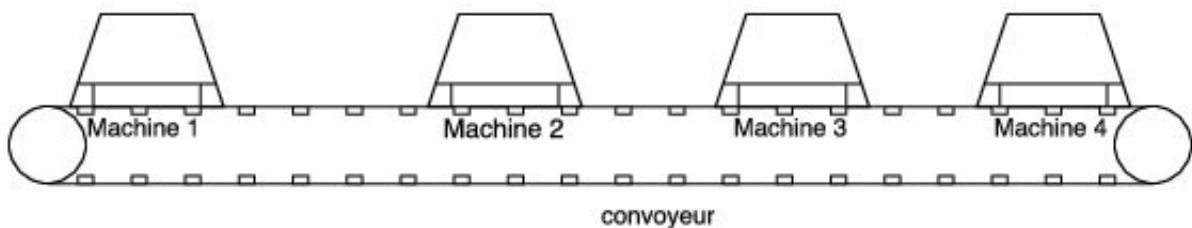


FIGURE 1.2 – Configuration en ligne de production

Sur la figure 1.2 nous observons que les machines sont reliées entre elles par un convoyeur. L'ordre des postes de travail ou des machines dépend de la précédence des opérations. Sur une ligne de production, les postes sont agencés de manière à minimiser les flux ré entrants, c'est à dire les retours en arrière des produits. Une ligne de production est bien sûr caractérisée par un sens de circulation des produits unique.

## La configuration en îlots ou en cellules de production

Une cellule de production a une configuration en atelier de production lorsque le transport entre les machines n'est pas assuré par un moyen de transfert mécanisé pouvant assurer un flux continu entre les pièces.

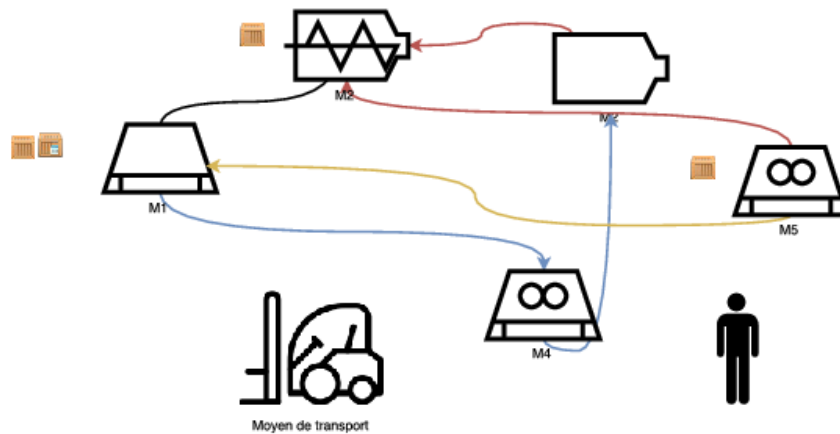


FIGURE 1.3 – Schéma d'un atelier de production en îlots de production

Sur la figure 1.3 on peut voir un atelier de production en îlots ou cellules de production. Les machines M1 à M5 ne sont pas reliées par un convoyeur comme cela était le cas pour la configuration en ligne de production (voir la figure 1.2). Ici le transport des produits s'effectue à l'aide d'un moyen de transport qui se trouve ici être un transpalette.

## La classification en fonction de la flexibilité

Dans le paragraphe précédent nous avons fait une classification des systèmes de production en fonction de leur configuration spatiale. A présent nous allons les classer en fonction de leur objectif de production et leur capacité d'évolution et d'adaptation. Sur la base de ces critères, [10] classe les systèmes de production en trois grands groupes : **les systèmes de production dédiés**, **les systèmes de production flexibles** et **les systèmes de production reconfigurables**

### 1.2.3.6.1 Les systèmes de production dédiés

D'après [11] ce sont des systèmes de production caractérisés par une automatisation fixe et dont le but principal est de fabriquer des produits de base ou les principaux éléments de base des produits d'une entreprise. Ces systèmes font le plus souvent de la production en grande série afin de diminuer les coûts unitaires de production. Les systèmes de production dédiés sont essentiellement des lignes de production. La notion de système de production dédié est liée à la mission de ce dernier à savoir la fabrication d'une pièce ou de famille de pièces en particulier. En règle générale la configuration en ligne de production se trouve privilégiée par ce qu'elle permet d'avoir un haut niveau de production. C'est pour cette raison que les systèmes de production sont souvent appelés lignes dédiées ou lignes de

transfert. En règle générale dans une entreprise qui utilise les lignes de transfert chaque ligne est dédiée à la fabrication d'un produit. Alors que les lignes de production dédiées ont pour avantage de permettre la production en grande série et donc de diminuer les coûts unitaires des produits, en revanche, elles permettent difficilement l'introduction de nouveaux produits dans le système de production. Cependant les systèmes de production flexibles permettent d'une certaine manière de répondre à cette difficulté.

#### 1.2.3.6.2 Les systèmes de production reconfigurables

Selon [10], Les systèmes de production reconfigurables sont des systèmes combinant le haut rendement des lignes de production dédiées et la flexibilité des ateliers de production flexibles mais ils doivent aussi être capables de s'adapter rapidement aux demandes du marché (variation des commandes, modification de la gamme de production).

[10] définit les caractéristiques d'un système de production reconfigurable à savoir :

- La modularité : cela signifie que toutes les composantes d'un système reconfigurables (machines, architecture physique et composantes logicielle) peuvent être ajustées, modifiées ou même supprimées ;
- L'intégrabilité : c'est la capacité à intégrer des modules avec rapidité et précision, et ce par des ensembles mécaniques, informationnels et des interfaces de contrôle facilitant l'intégration et la communication ;
- La personnalisation : l'auteur décline la personnalisation en deux types : une personnalisation de la flexibilité, et une personnalisation de la commande. La personnalisation de la flexibilité signifie pour [1] que les machines sont construites autour d'une famille de pièces qui sont en cours de fabrication et ne fournissent la flexibilité nécessaire que pour la production de ces dernières réduisant ainsi les coûts. Le même auteur complète en rajoutant que la personnalisation de la commande se réalise en intégrant des modules de commande à l'aide de la technologie d'architecture ouverte assurant ainsi les commandes exactes nécessaires. En d'autres termes la personnalisation de la commande consiste à adapter les modules de commandes du systèmes reconfigurables aux produits qui sont en cours de production.
- Convertibilité : c'est la capacité à transformer aisément les fonctionnalités d'un système existant pour qu'il réponde aux exigences d'un nouveau produit ;
- La diagnosticabilité : c'est la capacité de connaître de manière automatique l'état actuel d'un système et de détecter les causes et de diagnostiquer les problèmes qui surviennent sur des machines ou sur des produits et de les résoudre rapidement ;

### 1.2.3.6.3 Les ateliers de production flexibles

FMS (Flexible Manufacturing System) ou en français (Système de fabrication flexible) est une technologie qui vise à rendre flexible l'ensemble de l'outil de production. C'est-à-dire, le préparer à s'adapter aux divers changements de son environnement, sans qu'il y ait besoin d'engager de nouveaux investissements en biens d'équipement, ou d'engendrer de longues pertes de temps. Sa finalité est de réaliser une multitude d'opérations hétérogènes à partir d'un nombre très limité de ressources.

Un système de production flexible est un système qui se compose de machines à commande numérique, d'automates programmables et qui est capable de fabriquer des produits différents [11]. Les systèmes de production flexibles permettent de diversifier les produits mais le rendement est faible comparé à celui d'une ligne de production dédiée. Il faut bien préciser que comme dans le cas des systèmes de production flexibles et celui des systèmes des lignes de transfert, les positions des postes de travail sont fixes. Le fait que les positions des machines soient définitives empêche la configuration de l'atelier de s'adapter à l'évolution de la production. Dans de très nombreux articles, on remarque très souvent que la notion de flexibilité est abordée de manière différente. Les travaux de [12] et de [13] permettent de définir onze niveaux de flexibilités. On peut citer : la flexibilité des machines, la flexibilité des outils de manutention, la flexibilité des opérations, la flexibilité des processus de fabrication, la flexibilité du produit, la flexibilité du cheminement des produits, la flexibilité des volumes de produits, la flexibilité d'extension, la flexibilité des programmes de contrôle, la flexibilité de la production et la flexibilité face au marché. Dans les sections suivantes, nous allons détailler chacune des flexibilités déjà listées.

Les systèmes de productions flexibles présentent un certain nombre d'avantages selon les travaux de [1] :

- La production d'un nombre de produits variés grâce à la flexibilité des machines ;
- L'adaptation de la production face à la variation des volumes de production (réactivité face au marché) ;
- L'introduction d'un nouveau produit de la même famille est possible et ne nécessite pas un ré-aménagement de l'outil de production ;
- La flexibilité d'expansion d'un système de production flexible permet à celui de pouvoir évoluer contrairement à celui d'une ligne de production dédiée.

Le même auteur rappelle aussi les inconvénients de ces derniers.

- Un coût des machines à commande numériques très élevé
- Des machines à commande numérique ne pouvant effectuer qu'une seule opération
- Un rendement plus faible que celui des lignes de production dédiées du fait des temps de changements d'outillages.



Les types de systèmes de production que nous avons présentés dans ce paragraphe ne sont pas mis en opposition. Ils se complètent et surtout répondent chacun à des problématiques différentes. Le tableau 1.1 présente les principales caractéristiques des systèmes dédiés, flexibles et reconfigurables, notamment sur le plan de la structure des machines, sur les éléments sur lesquels ces systèmes sont focalisés, sur la capacité à évoluer, sur leur flexibilité et sur la versatilité

TABLE 1.1 – Tableau comparatif type de systèmes de production [1]

	<b>Systèmes de production dédiés</b>	<b>Systèmes de production reconfigurables</b>	<b>Systèmes de production flexibles</b>
Structure des machines	Machine fixe	Structure ajustable	Structure fixe
Focalisation du système	Focalisation sur les pièces	Focalisation sur une famille de pièces	Focalisation sur les machines
Evolutivité	Non	Oui	Oui
Flexibilité	non	Possibilité de personnalisation	Général
Utilisation simultanée d'outils	Oui	Oui	Non

On peut déjà observer à travers ce tableau que la structure des systèmes dédiés ainsi que des systèmes flexibles est généralement fixe comparée à celle des systèmes reconfigurables. Cela peut être un atout pour une entreprise qui doit réorganiser régulièrement son implantation, mais cela ne s'avère pas utile pour une entreprise qui ne fabrique qu'un seul produit avec très peu de variantes. Il faut tout de même reconnaître que la capacité à faire évoluer un système de production qui possède à la fois les caractéristiques d'un système de production flexible et reconfigurable les rend plus attrayant pour des entreprises ayant des fréquences de production variables et une forte diversification des produits. Enfin le tableau 1.1 nous montre aussi qu'en fonction de ceux sur quoi se focalise un système on peut être amené à se tourner vers un des trois types de systèmes de production. Sachant que les lignes de production dédiées sont centrées sur la pièce à fabriquer, les systèmes de production flexibles sont quant à eux centrés sur la machine, cela signifie que dans la mise en place d'un système de production l'accent est souvent mis sur la flexibilité de la machine et notamment le nombre d'outils que cette dernière possède ou encore ses fonctionnalités. Par contre les systèmes de production reconfigurables sont focalisés sur des familles de pièces.

# La Flexibilité dans un système de production

## La flexibilité des machines

La flexibilité d'une machine fait référence à la variété d'opérations que peut réaliser celle-ci [12]. Les avantages de la flexibilité des machines dans un système sont : la diminution de la taille des lots de transport, l'augmentation du taux d'utilisation des machines, la production de pièces complexes, la réduction du temps d'introduction de nouveaux produits dans le système de production, etc. Les moyens qui permettent la mise en œuvre de la flexibilité des machines sont : le contrôle numérique, la sophistication du chargement des pièces et du changement d'outillage. La formation du personnel pour leur permettre d'avoir plus d'autonomie en ce qui concerne la prise de décision sur l'arrêt des machines est aussi un moyen d'améliorer la flexibilité des machines. La mesure de la flexibilité d'une machine consiste en la détermination du nombre d'opération de la machine, par la mesure du temps de changement de fonctionnalité, ou encore par le nombre d'outils ou de programmes. Sur la figure 1.4 on peut voir deux machines l'une possédant plusieurs outils

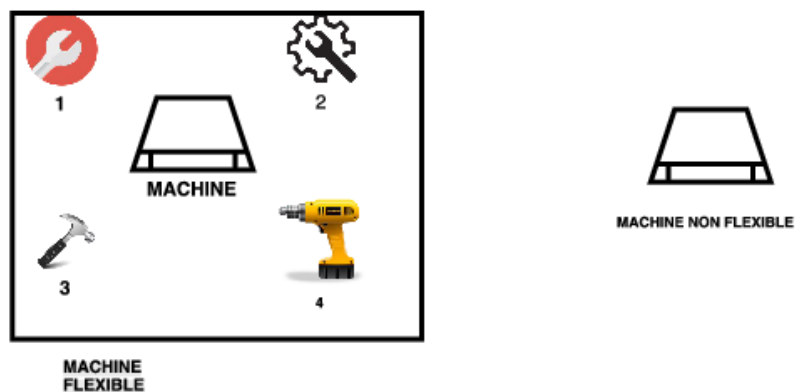


FIGURE 1.4 – Illustration de la flexibilité des machines

supplémentaires (machine de gauche) et l'autre n'en possédant pas (machine de droite). Alors la machine de gauche sera donc plus flexible que celle de droite dans la mesure où elle pourra effectuer plusieurs opérations.

## La flexibilité des outils de manutention

C'est la capacité d'un système de manutention à déplacer les différentes pièces de manière efficiente à travers le système de production lors d'une phase de production. Les finalités de la flexibilité des moyens de manutention sont l'augmentation de la disponibilité des machines, l'augmentation du taux d'utilisation des machines, la réduction des temps de production et par conséquent l'augmentation du rendement du système de production manufacturier. La mise en place de la flexibilité du système de manutention nécessitent des moyens de transports tels que les chariots élévateurs et des installations hautement spécialisées, la nature de l'implantation de l'outil de production, la propreté de l'environnement ainsi que l'amélioration de l'ergonomie sont des facteurs qui augmentent la flexibilité des systèmes de manutention d'un système de production. Pour mesurer la flexibilité des moyens de manutention, [12] définit le système de manutention idéal comme un système qui permet de relier toutes les machines du système de production deux à deux. La mesure de flexibilité du système de manutention est alors le ratio du nombre de chemins qu'un système de production peut supporter sur le nombre de chemins du système de manutention universel correspondant à ce système.

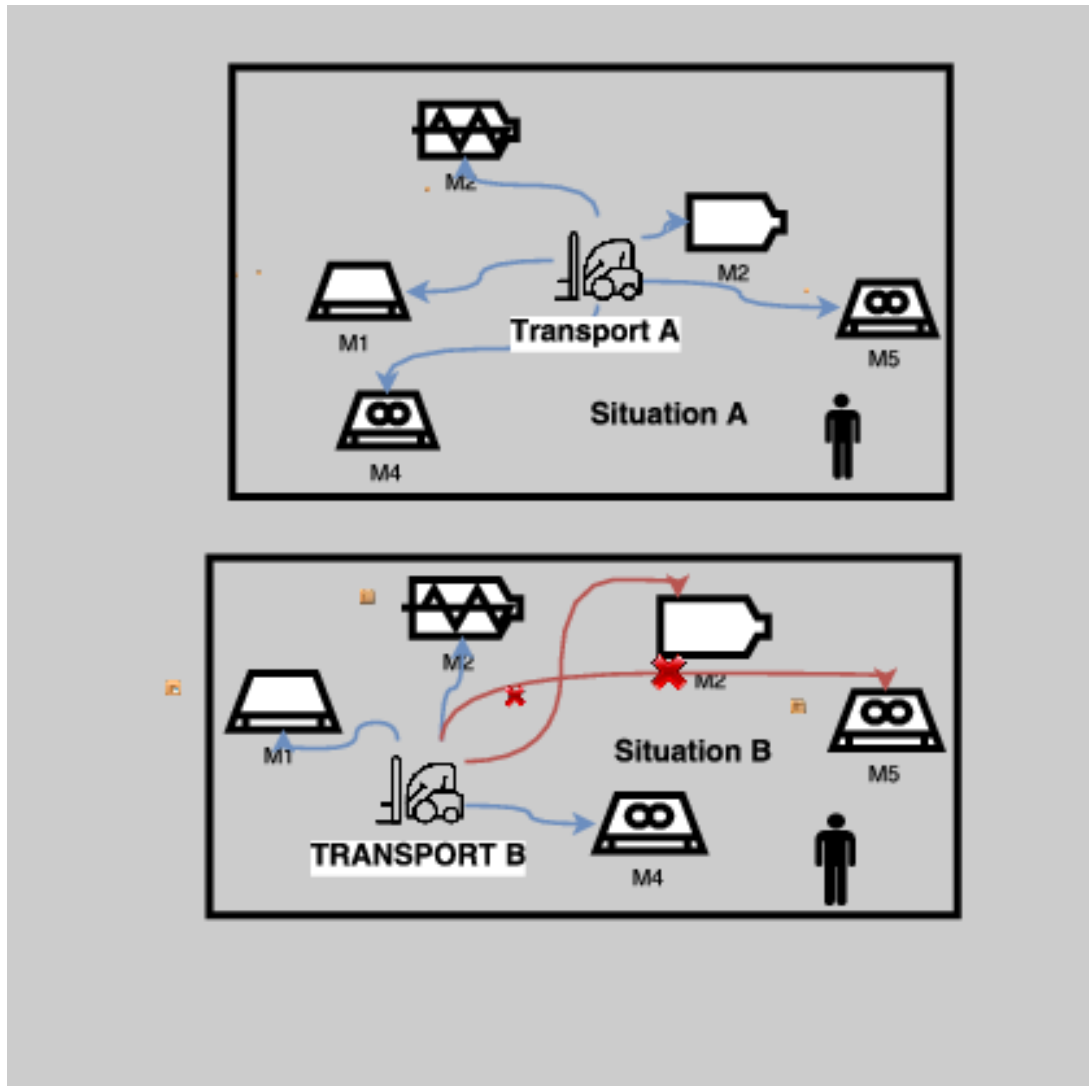


FIGURE 1.5 – Illustration de la flexibilité des moyens de maintenance : deux situations sont présentées. La situation A avec un transporteur A et la situation B avec un transport B.

La figure 1.5 présente dans un même atelier de production deux situations. Dans la situation A, il y a un transporteur (*transporteur A*) qui a la possibilité de se déplacer vers toutes les machines au cours de ses activités de transport. De l'autre côté nous avons un autre transporteur qui a des déplacements limités dans l'atelier de production. En effet le transporteur B ne peut pas se déplacer vers les machines M2 et M5. On dira alors que le système de production dans la situation A a une meilleure flexibilité des moyens de maintenance que le système dans la situation B. Dans la situation A le transporteur peut se déplacer vers toutes les machines du système on dira dans ce cas que le système de maintenance est universel.

## La flexibilité des opérations

C'est la capacité d'intervertir ou de remplacer les opérations qui permettent la fabrication d'une pièce. Dans ce cas de figure, cela signifie qu'il n'existe pas de contraintes de précédence entre toutes les opérations de fabrication d'une pièce. Les objectifs de la flexibilité des opérations sont l'amélioration de la disponibilité des machines, et de leur taux d'utilisation, la possibilité de poursuivre la production même si une machine est défaillante et la facilitation de l'ordonnancement en temps réel des pièces. La flexibilité des opérations dépend essentiellement de la conception et de la nature d'une pièce. La mesure de la flexibilité des opérations peut se mesurer en dénombrant pour une pièce le nombre de plans de fabrication possibles.

## La flexibilité des processus de fabrication

C'est l'ensemble des pièces qu'un système peut produire sans pour autant qu'on y opère des modifications majeures. L'objectif d'une telle flexibilité est la diminution de la taille des lots de production et des stocks, l'augmentation des ressources et la minimisation de la duplication de machines. Pour mettre en place la flexibilité des processus de fabrication, le système doit posséder une flexibilité des machines, une flexibilité des opérations significative, et une flexibilité des moyens de manutention. Il faut enfin noter que la présence de personnels possédant des compétences transverses permet d'améliorer la flexibilité des processus de fabrication. Une manière de mesurer la flexibilité des processus est de déterminer le volume des pièces produites sans changement.

## La flexibilité du produit

C'est la capacité à intégrer une nouvelle pièce dans un système de production tout en minimisant les temps de changement, mais aussi en minimisant les coûts. Cette flexibilité peut être atteinte grâce à une flexibilité des machines mais aussi par un système de planification et de contrôle efficient [12].

## La flexibilité d'acheminement des produits

C'est la capacité d'un système de production à prendre en compte les pannes des machines qui surviennent lors de la production et de continuer tout de même à produire. La flexibilité des produits est un atout pour les entreprises innovantes ou celles qui sont dans une phase de croissance car l'introduction d'un nouveau produit fait généralement suite à une innovation technologique ou à une stratégie de croissance de l'offre. Ce niveau de flexibilité ne peut être atteint que si l'on possède une certaine flexibilité des machines et des machines à commande numériques possédant plusieurs outils. Il est aussi important que le système de production possède une flexibilité des moyens de manutention, une flexibilité des opérations, une technologie de groupe, des routines de fabrication des pièces similaires, et aussi un logiciel de réarrangement dynamique de la production. Une manière de mesurer

la flexibilité d'acheminement d'un système de production est de déterminer le nombre de cheminements possibles pour la fabrication d'une pièce. Cette mesure est cependant propre à chaque produit ce qui permet d'avoir une vue partielle de la flexibilité d'acheminement. Une autre manière de mesurer la flexibilité d'acheminement est de calculer l'entropie d'acheminement. L'entropie d'acheminement est la mesure de l'entropie des informations contenues dans une liste d'opérations et de machines, à partir de laquelle l'opération suivante et la machine sur laquelle s'effectuera cette opération sera choisie [14]. Nous reviendrons avec un peu plus de détails sur la notion de flexibilité d'acheminement des produits dans le chapitre 3 lorsque nous aborderons la mesure de la flexibilité de routage d'un atelier de production .

## La flexibilité des volumes de produits

C'est la capacité d'un système de s'adapter à la variation des commandes. Cette adaptation peut se traduire par un ajout/suppression d'une(des) machine(s) en cas d'augmentation/diminution du volume de production soit par une modification de la configuration de l'atelier en fonction des processus de fabrication des pièces, ou en encore par une modification de l'ordonnancement. La flexibilité des volumes dépend des coûts de la variabilité des coûts de production. Un système de production qui a des coûts de production variables élevés par rapport à ces coûts fixes a plus de chances de mettre en place la flexibilité des volumes. Cependant un système de production qui possède une forte automatisation de ses machines, peut facilement s'adapter à la variation du volume de sa production. La mesure de la flexibilité des volumes de production peut se mesurer en prenant en compte les volumes de gammes de production profitables pour une entreprise [15] . On peut aussi la mesurer par le ratio de la variation moyenne de volume sur une période de temps donné par la capacité de production limite.

## La flexibilité d'extension

C'est la capacité d'expansion d'un système de production selon la volonté du concepteur. Pour atteindre ce niveau de flexibilité il est primordial de posséder une flexibilité au niveau des machines et une flexibilité au niveau des moyens de manutentions. La flexibilité d'extension d'un système de production concerne des entreprises qui connaissent une croissance d'activité. Selon [16] la flexibilité d'expansion aide à la réduction des coûts des temps d'implémentation et des coûts de nouveaux produits. De plus, pour atteindre ce niveau de flexibilité il faut éviter les processus de conception basés sur l'implantation des systèmes de production (par exemple les lignes dédiées de production), il est préférable d'avoir des cellules flexibles de production modulaires. De même la flexibilité d'expansion se met facilement en place si le système de production possède une flexibilité de son système de manutention, une flexibilité de ses machines (machines outils) ainsi qu'un haut niveau d'automatisation. La flexibilité d'expansion se mesure par l'effort global et le coût dont on a besoin en termes de temps pour augmenter la capacité de production d'un système [16] . De manière plus spécifique [13] mesure la flexibilité d'expansion par le ratio du coût du doublement de la production par le coût de production du système d'origine.

## La flexibilité des programmes de contrôle

Dans les usines où plusieurs équipes travaillent mais à des horaires décalés, il y a un moment où doit se faire le changement d'équipe. Cette période entraîne un ralentissement de la production. Une des solutions pour stabiliser la production pendant le changement d'équipe est l'augmentation de la flexibilité des programmes de contrôle du système de production. Ainsi, la flexibilité des programmes de contrôles est la capacité d'un système de production à fonctionner en friche pendant une période de temps relativement longue. Une telle flexibilité a pour objectif de : permettre le changement d'équipes, de réduire les temps de lancement de la production et par conséquent augmenter les rendements du système de production. Pour un système de production la capacité de travail en friche augmente la capacité effective du système de production[12]. La mise en place d'une telle flexibilité nécessite une flexibilité d'acheminement ainsi que des processus, la présence de capteurs pour le contrôle et la détection. La flexibilité des programmes de contrôle nécessite une compréhension accrue des processus de fabrication afin de mettre en place des procédures pour toutes les éventualités qui surviendraient sur le système. La flexibilité des programmes de contrôle exige aussi une expérimentation contrôlée pour l'apprentissage, la modification des connaissances accumulées sous forme de programmes informatiques et la capacité de transfert de ces connaissances pour des produits, des processus ou des procédures similaires.

## La flexibilité de la production

C'est l'ensemble des pièces que peut produire un système de production sans qu'on ne lui ajoute un autre équipement majeur. Cette caractéristique définit le potentiel de production. La flexibilité de la production permet aux entreprises qui se positionnent dans des secteurs de nouveaux produits sans cesse de rester compétitives. Elle permet d'éviter des coûts liés à la mise en place de nouveaux produits et permet aussi sur le plan opérationnel d'augmenter l'ensemble des familles de pièces que peut produire un système de production, ce qui permet à une entreprise de diversifier ses activités et donc de diminuer ses risques. La mise en place d'une telle flexibilité nécessite la présence d'une flexibilité au niveau des machines, mais aussi au niveau du système de manutention et du système de contrôle. La meilleure manière de mesurer la flexibilité de la production est de déterminer le nombre total de pièces fabriquées par ce dernier [13].

## La flexibilité face au marché

C'est la capacité d'un système de production à s'adapter au changement de l'environnement du marché. Ce type de flexibilité est très important pour la survie d'entreprise dans un environnement en constante mutation. Cette mutation pourrait être due à des innovations technologiques, au changement de goûts et comportements des consommateurs, à la courte durée de vie des produits ou encore à l'incertitude des sources d'approvisionnement. La flexibilité face au marché permet aux firmes de faire face à ces changements, de plus cette flexibilité est aussi un vrai atout pour une entreprise face à ses concurrents.

Pour assurer une flexibilité face au variation du marché, il est nécessaire que le système possède les flexibilités des produits, des volumes et aussi de l'expansion. En plus de cela, la mise en place d'une flexibilité face au marché nécessite aussi bien un bon système de prévision qu'une bonne collaboration entre une entreprise et ses fournisseurs. La flexibilité face au marché peut se mesurer comme la mesure pondérée de l'effort en termes de temps et de coût requis pour introduire un nouveau produit, ou encore pour augmenter ou baisser le volume de production d'une quantité déterminée et pour ajouter une unité de capacité [13].

Nous venons de présenter les trois catégories de systèmes de production que l'on retrouve dans la littérature à savoir les systèmes de productions dédiés, flexibles et reconfigurables. Il faut noter que en ce qui concerne les systèmes de production reconfigurables, nous n'avons pas trouvé de cas réel où ils sont implantés, ce qui nous laisse penser que leur implémentation au sein des entreprises est probablement en cours. Pour cette raison nous n'intégrerons pas les contraintes des systèmes de production reconfigurables dans l'élaboration de l'outil que nous souhaitons mettre en place.

D'autre part nous avons rappelé dans l'introduction notre intention de proposer un modèle de simulation des systèmes de production manufacturiers ; il est alors important pour nous de faire un point sur les types de modèles et les approches de modélisations qui sont adoptés dans les problématiques liées aux systèmes de production c'est l'objet de la prochaine partie.

## **Modélisation et simulation des systèmes de production**

Nous nous engageons dans la proposition d'un modèle de simulation des systèmes de production, alors il est important de rappeler quels sont les objectifs de la modélisation pour les systèmes de production, quels sont les principaux modèles utilisés, et aussi quelles sont les démarches et approches utilisées dans le cadre de cette modélisation. De cette manière nous pourrons choisir le modèle que nous allons utiliser dans le cadre de notre travail. Avant cela rappelons les objectif de la modélisation des systèmes de production.

### **Modèles pour les systèmes de production**

La modélisation des systèmes de production manufacturiers a trois objectifs principaux :

- L'analyse, la description ou la simulation : les outils d'analyse et de description permettent de définir les modèles des sous-systèmes d'un système de production tels que définis dans [3] à savoir les sous-systèmes physiques, d'information et décisionnels.
- L'évaluation : les modèles d'évaluation permettent de déterminer la performance d'un système de production à travers des indicateurs définis par le gestionnaire du système de production. Les modèles qui permettent l'évaluation d'un système de



production sont le plus souvent des modèles dits de simulation. Ces modèles de simulation se basent sur des équations mathématiques ou sur la théorie des graphes. La simulation n'est évidemment possible qu'à l'aide de logiciels de calcul ou de simulation adaptés.

- L'optimisation : elle se fait par les méthodes exactes ou des méthodes approchées (heuristiques et méta heuristiques), utilisant des modèles mathématiques ou des graphes.

Dans le domaine de la modélisation des activités on distingue principalement deux types de modèles, les modèles mathématiques, et les modèles à événements discrets. Parmi les modèles mathématiques nous avons les modèles linéaires, les modèles en nombre entiers, les modèles quadratiques. Et parmi les modèles à événements discrets nous distinguons les graphes disjonctifs-conjonctifs, les graphes de précedence, les automates, et les réseaux de Pétri.

## Les Modèles Mathématiques

Les modèles mathématiques sont utilisés pour formaliser des problèmes, définir des fonctions objectifs, mais aussi les contraintes d'un système de production. La plupart des problèmes liés aux ateliers et lignes de production ont toujours une formulation mathématique. Ainsi on retrouve dans de nombreux travaux tels que [17], [18], [19] des modèles mathématiques pour la résolution de problèmes d'ordonnancement d'atelier de production. On retrouve aussi des travaux sur la modélisation des problèmes d'équilibrage des lignes [20]. Les modèles mathématiques sont aussi couramment utilisés pour résoudre les problèmes d'implantation des systèmes de production. On peut citer quelques problèmes tels que le problème d'affectation quadratique, le problème d'affectation linéaire ou encore le problème de couverture d'ensemble. Parmi les modèles mathématiques on distingue les modèles linéaires, les modèles en nombre entiers et les modèles quadratiques. L'utilisation de ces modèles dépend du type de problème à solutionner, et aussi de la volonté du modélisateur.

L'utilisation des modèles mathématiques permet grâce aux solveurs de trouver des résultats exacts aux problèmes combinatoires de petites tailles. Ce pendant lorsque les problèmes ont de grandes tailles et notamment de nombreuses variables décisionnelles, les méthodes exactes ont des temps de calcul extrêmement élevés. C'est pourquoi des méthodes heuristiques sont souvent substituées aux solveurs. A titre d'exemple dans les travaux de [17], l'utilisation du modèle mathématique pour la résolution des problèmes de JOB SHOP avec des contraintes de transport donne des temps de calcul très importants. En outre, les modèles mathématiques présentent une faiblesse, c'est celle de ne pas permettre une analyse descriptive aisée du système, ce que les modèles basés sur la théorie des graphes font aisément.

## Les modèles basés sur la théorie des graphes

Parmi les modèles basés sur la théorie des graphes nous pouvons citer les graphes conjonctifs, les automates, et les réseaux de Pétri.

De manière générale, un graphe simple non orienté  $G = (X, U)$  est défini par :

- Un ensemble  $X = \{x_1, x_2, \dots, x_n\}$  dont les éléments sont appelés sommets ou nœud.
- $U = \{u_1, u_2, \dots, u_m\}$  est un ensemble inclus dans  $X \times X$ . Les éléments de  $U$  sont appelés arcs ou arêtes. Soit  $u$  un élément de  $U$  alors  $\exists(x, y) \in X \times X$   $u = (x, y)$ . Lorsque  $u = (x, y)$  on dit que les sommets  $x$  et  $y$  sont adjacents ou voisins. Deux arêtes sont dites incidentes si elles ont un sommet en commun. Le nombre d'arêtes incidentes à un sommet  $x$  est appelé degré de  $x$ . C'est aussi le nombre de voisins de  $x$ .

Les principaux graphes qui sont utilisés dans un cadre de modélisation des systèmes de production manufacturiers sont les graphes disjonctifs ou conjonctifs, les graphes de précedence, les automates et les réseaux de Pétri. Nous présentons dans les paragraphes qui suivent quelques travaux utilisant ces différents graphes.

### 1.4.1.2.1 les graphes conjonctifs et disjonctifs

[21] définit un graphe conjonctif comme "un graphe valué  $G = (X, U)$  comportant une seule entrée et une seule sortie  $n$  tel qu'il existe un chemin de valeur positive entre l'entrée 1 et tout autre sommet, et, entre tout sommet différent de la sortie  $n$ , et cette sortie".

Dans un graphe disjonctif, les sommets représentent les tâches effectuées et les arcs correspondent aux contraintes de potentielles qui peuvent exister entre les tâches. L'obtention d'un graphe disjonctif se fait en complétant l'ensemble des tâches par deux tâches fictives, 1 et  $n$ .

Les modèles basés sur les graphes disjonctifs sont utilisés pour résoudre des problèmes d'ordonnancement en général, et en particulier des problèmes où les ressources sont déjà dédiées à des tâches. Nous pouvons citer les travaux de [22] [17], dans lequel l'auteur présente un modèle basé sur les graphes disjonctifs-conjonctifs pour résoudre les problèmes d'ordonnancement des Job Shop avec transport.

### 1.4.1.2.2 Les machines à états

Une machines à état, encore appelé automate est une entité ayant un état courant et d'autres états par lesquels elle est déjà passée ou par lesquels elle passera (éventuellement) dans le futur. Le changement d'état s'appelle transition.

De manière formelle, un automate est un quadruplet  $A = (S, S_0, E, T)$  où :

- $S$  un ensemble d'états
- $S_0$  un ensemble d'états initiaux de l'automate

- $E$  : ensemble fini des étiquettes des transitions
- $T$  : ensemble des transitions  $T \subseteq S \times E \times S$

Les réseaux d'automates sont utilisés dans le cadre de résolution de problèmes d'ordonnement dans les travaux de [23]. Le formalisme des machines à états est très peu utilisé dans le cadre de la modélisation des systèmes de production.

#### 1.4.1.2.3 Les réseaux de pétri

Un Réseau de Pétri (RdP) est un quadruplet  $Q = (P, T, Pré, Post)$  avec :

$P = \{P_1, P_2, \dots, P_n\}$  est un ensemble fini de places

$T = \{T_1, T_2, \dots, T_n\}$  est un ensemble fini de transitions

$Pré : P \times T \rightarrow N$  l'application d'incidence avant

$Post : P \times T \rightarrow N$  application incidence arrière

$Pré(P_i, T_j)$  et  $Post(P_i, T_j)$  représente respectivement les poids de l'arc reliant  $P_i$  à  $T_j$  et le poids de l'arc reliant  $T_j$  à  $P_i$ .

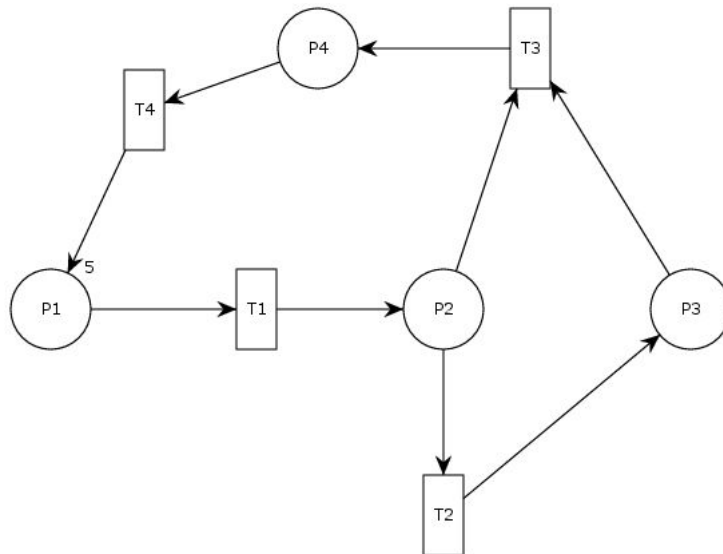


FIGURE 1.6 – Un exemple de Réseau de Pétri

Les réseaux de Pétri sont utilisés tout comme les machines à états pour modéliser des systèmes à événements discrets. Ils sont utilisés dans le cadre de la modélisation des activités de production pour analyser, modéliser et simuler les systèmes de production. En fonction des applications, on distingue plusieurs types de réseaux de Pétri :

- Les réseaux de pétri temporisés [24, 25];
- Les réseaux de Pétri colorés [26, 27];
- Les réseaux de Pétri flous [28]

- Les réseaux de Pétri déterministes et stochastiques [29]
- les réseaux de Pétri de haut niveau[30]

Les matrices de transition des modèles basés sur la théorie des graphes permettent d'obtenir un modèle mathématique du système que l'on souhaite analyser, simuler ou même optimiser. Dans ce cas de figure la recherche des invariants permet de déterminer certaines propriétés du système. Cependant l'utilisation exclusive de la théorie des graphes peut poser quelques problèmes :

- Le premier problème est celui de la définition de la fonction objectif si l'on se trouve dans le cas d'un problème d'optimisation. En effet l'utilisation des graphes couplée à un simulateur permet d'obtenir des résultats par simulation mais ne permet pas toujours d'avoir l'expression mathématiques de la fonction objectif.
- Le deuxième problème est celui de la détermination de la matrice de transition. lorsque l'on a affaire à un réseau de Pétri simple l'expression de la matrice de transition est relativement simple dans la mesure où le poids des arcs est une fonction de  $N \rightarrow N$  alors que dans le cas des réseaux de Pétri colorés ou de haut niveau, la fonction située au niveau du poids des arcs est un morphisme. Dans ce cas la matrice de transition devient une matrice de morphismes et la détermination de l'invariant dans ce cas de figure devient moins aisée.

Le choix d'un type de modèle dépend du problème auquel on est confronté, et de la capacité du modèle à intégrer les contraintes du système réel, et aussi de l'objectif de la modélisation. Et, ce qui nous concerne, les objectifs de l'outil que nous souhaitons mettre en place sont :

- Représenter la structure d'un système de production :
- modéliser les interactions entre les composantes du système
- Gérer des événements (début de production, fin d'une opération de fabrication ou de transport)
- Observer l'évolution au cours du temps à travers des simulations

Ces objectifs rendent le choix des modèles analytiques inopérant dans la mesure où ces derniers ne permettent pas une gestion aisée des événements. De plus les modèles analytiques ne sont pas en mesure d'assurer une traçabilité et une extraction de données au cours d'une simulation. De plus comme nous ne sommes pas dans une démarche de recherche d'une solution optimale l'utilisation de modèle mathématiques parait moins appropriée. En revanche, les modèles basés sur la théorie des graphes en donne la possibilité. Parmi ceux-ci nous travaillerons avec les réseaux de Pétri plutôt que des graphes ordinaires ou encore des machines à états. Ce choix est justifié par le fait que la sémantique et la syntaxe des réseaux de Pétri est déjà clairement définie alors que opter pour des graphes disjonctifs (ou d'autres types de graphes) impliquerait la définition de la sémantique et la syntaxe de ce dernier (définition des nœuds, définition des arcs, définition des transitions), sans parler des outils de vérification du modèle.

## Les RdP et la gestion de production

Dans la littérature [31],[32], [33] : les RdP sont utilisés dans le cadre de la modélisation des systèmes de production manufacturiers pour décrire le fonctionnement du système, d'étudier ses caractéristiques et de le simuler et surtout pour vérifier le fonctionnement des modèles développés grâce à la vérification de la vivacité du réseau de pétri [34].

Plusieurs travaux se sont intéressés à l'utilisation des RdP pour la modélisation et la simulation des systèmes de production quel que soit leur type : ligne de production [35], ateliers en job-shop [26, 36], [25] [37], systèmes de production flexibles [38], [27] et systèmes de productions reconfigurables [39]. Les RdP ont aussi été exploités pour l'amélioration des performances des systèmes de production [40], où plusieurs travaux ont utilisé des modèles de simulation basés sur les RdP tels que [41] qui a utilisé des RdP temporisés pour la détermination d'un ordonnancement de production optimal, [27] qui propose une méthode pour la minimisation de la somme des retards dans un atelier de production flexible en utilisant des RdP combiné avec des Algorithmes génétiques ou les travaux de [38] qui ou l'on utilise les RdPC ainsi que les algorithmes génétiques pour la réduction des coûts de production et l'amélioration de la flexibilité du système de production flexible, ou pour augmenter leur robustesse [42]. Dans le Tableau 1.2 nous avons résumé quelques travaux qui ont utilisé des RdP pour les systèmes de production.

Référence et date	types de RdP	Description
[43] 1989	RdP-Temporisés	Modélisation des Systèmes de production flexibles en utilisant des RdP temporisé.
[38] 2002	RDPC	Optimisation multi objectif pour la minimisation des coûts de la production sous multiples plans de production et minimisation des coûts de reconfiguration. les RdPC sont couplés aux algorithmes génétiques ainsi qu'à la règle de priorité SIO (Shortest Imminent Operation time)
[41] 2003	RdP-Temporisés	Optimisation de l'ordonnancement à l'aide de l'algorithme A* (recherche du plus petit chemin) et les graphes de marquage afin de minimiser la productivité (Makespan) des ateliers de production flexibles
[44] 2003	RdPC temporisés	L'auteur utilise les RdPC pour modéliser et analyser un atelier de fonderie de type Job-Shop sous contraintes de périssabilité des ressources. L'objectif de cette analyse est de maximiser le nombre de produits par unité de temps.
[45] 2008	RdP de Haut niveau	L'objet de cet travail est l'utilisation des RdP pour la modélisation et l'évaluation des Systèmes de production flexibles. L'auteur propose aussi une vision objet des réseaux Pétri, où les jetons sont désormais considérés comme des objets.

TABLE 1.2 – Liste des articles ( suite ).

Référence et date	Type de RdP	Description
[46] 2009	RdP Colorés	Modélisation et maximisation de la performance du système de transport constitué de véhicules à guidage automatique. L'activité d'une machine est représentée par une transition.
[39] 2010	RdPC temporisés orientés objet	L'auteur utilise les RdP colorés temporisés avec une approche objet pour modéliser les systèmes de production reconfigurables. Quatre classes ont été développées : postes de travail, stocks, véhicules autoguidés et robots. Dans cet article, un accent est mis sur la communication dans les systèmes de production reconfigurable ; l'auteur met en avant des outils de production qui communiquent entre eux. Cependant cet article ne mentionne pas les indicateurs de performance.
[26] 2011	RdP Colorés	Utilisation des graphes d'état temporisés pour modéliser le traitement d'un Job-Shop à deux machines.
[47] 2011	Les Réseau de Pétri Stochastiques	Dans ce travail, les auteurs présentent, une méthodologie pour l'analyse des systèmes de production flexibles. Cette méthodologie est basée sur les réseaux de Petri stochastiques et modulaires. Les auteurs présentent dans leur travail les modèles des éléments de base d'un système de production flexible à savoir : les machines, les ressources de transport, les produits. Les auteurs développent aussi un modèle stochastique afin de modéliser la sûreté de fonctionnement.
[48] 2012	RdP Colorés Stochastiques	Cet article traite de l'analyse et de la modélisation d'ateliers de production . Dans celui-ci, l'objectif est de déterminer le taux d'occupation des équipements de l'atelier de production. Dans le cas présenté un opérateur assure la production sur deux postes de travail en tenant compte des problèmes d'approvisionnement sans considération de la préemption des tâches.
[49] 2012	RdPC	L'auteur utilise les RdPC (pour la modélisation et la simulation avec le logiciel de simulation CPN Tools) couplé avec un algorithme GSA (Gravitational Search Algorithm). L'objectif est de minimiser le temps total d'écoulement des produits (Makespan) dans un atelier flexible avec un ordonnancement type Job-Shop.

TABLE 1.2 – Liste des articles ( suite ).

Référence et date	Type de RdP	Description
[27] 2013	RdP colorés	Minimisation de la somme des retards pondérés à l'aide des RdP et des algorithmes génétiques : application à un Job Shop Flexibles
[25] 2013	RdP temporisés	Modélisation et simulation de l'ordonnancement des Job-Shop à l'aide des RdPC. Les modèles des machines qui sont présentés prennent en compte la disponibilité de la machine mais ne semble pas prendre en compte les différents outillage. Ce type de modèles ne pourrait pas être très efficace si on veut modéliser des Job-Shop flexibles. Ce type de modèle permet néanmoins de vérifier les propriétés des RdP.
[29] 2014	RdP déterministes et Stochastiques	Les modèles sont utilisés pour évaluer la performance des Systèmes de production manufacturiers. Les modèles de machines utilisés prennent en compte le problème de disponibilité par le changement d'outillage. Ce travail présente une méthodologie de conception des systèmes de production modulaire avec prise en compte des temps de transport, des pannes des machines, et des tâches de maintenance.
[28] 2014	RdP flous	Planification de la maintenance des systèmes manufacturiers de production sous contraintes de temps en se basant sur les RdP flous.
[50] 2016	RdP place-transition	Développement d'une méthodologie permettant l'augmentation de la flexibilité de planification dans les systèmes de production orienté services.

Cet aperçu synthétique des travaux sur les systèmes de production flexibles en utilisant les réseaux de pétri de divers types montre que, les RdP ont été utilisés depuis plus de 40 ans pour la modélisation et la vérification de différents aspects d'un système de production flexible [51], [52]. Nous avons aussi remarqué que peu de travaux ont considéré la modélisation et l'optimisation de la tâche de transport à l'intérieur de l'atelier de production et la flexibilité de routage.

Nous avons choisi d'utiliser les réseaux de Petri Colorés hiérarchiques dans le cadre de notre travail de modélisation. Dans la prochaine section nous allons les présenter de manière formelle.

## Les Réseaux de Pétri colorés Hiérarchiques

Dans la section 1.4, nous avons justifié le choix des réseaux de Pétri, pour la mise en place de notre outil d'aide à la conception des ateliers de production flexibles. Dans cette section, nous présentons les réseaux de pétri colorés Hiérarchiques que nous allons utiliser dans le reste de notre travail. La notion de réseaux de pétri colorés hiérarchiques repose sur deux concepts que nous présentons dans ce travail à savoir les réseaux de Pétri colorés et les réseaux de Pétri colorés modulaires. A la fin de ce travail, nous présentons un ensemble de travaux de recherche dans lesquels les réseaux de pétri sont utilisés afin de répondre à diverses problématiques de la gestion de production.

### Les réseaux de Pétri Colorés

Dans [53], Un RdP coloré "Colored Peri Net (CPN)" est défini comme un neuf-uplet :  $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  où :

1.  $P$  est un ensemble fini de places
2.  $T$  est un ensemble fini de transitions tel que  $P \cap T = \emptyset$ .
3.  $A \subseteq P \times T \cup T \times P$  est un ensemble d'arcs orientés.
4.  $\Sigma$  est un ensemble fini non vide d'ensembles de couleurs.
5.  $V$  est un ensemble fini de variable typées tel que  $Type[v] \in \Sigma$  pour tout  $v \in V$ .
6.  $C : P \rightarrow \Sigma$  est une fonction d'ensemble coloré qui assigne une couleur à chaque place.
7.  $G : T \rightarrow EXPR_V$  est une fonction de garde qui assigne une garde à chaque transition  $t$  de manière à ce que  $Type[G(t)] = Bool$  où  $Bool$  est une variable booléenne.
8.  $E : A \rightarrow EXPR_V$  est une fonction d'expression d'arc qui assigne une expression à chaque arc de manière à ce que  $Type[E(a)] = C(p)_MS$  où  $p$  désigne la place connectée à l'arc  $a$
9.  $I : P \rightarrow EXPR_\emptyset$  est une fonction d'initialisation qui associe à chaque place  $p$  une expression d'initialisation de telle manière que  $Type[I(p)] = C(p)_MS$

Les réseaux de Pétri colorés permettent de modéliser l'information dans un système de production. Contrairement aux réseaux de Pétri places-transition il est possible d'avoir dans une même place des jetons de nature différente. Si une place représente un stock, avec les réseaux de Pétri colorés il est possible de mettre dans cette place deux jetons de différentes natures représentant des produits différents alors qu'avec les réseaux de pétri simples cela n'est pas possible car il sera impossible de distinguer les deux jetons.



Dans un système de production flexible, les produits peuvent être de nature diverse c'est pour cela que l'utilisation des réseaux de pétri colorés s'avère nécessaire. De plus les couleurs d'un réseau de Pétri peuvent représenter n'importe quel type de données (réels, entiers, chaîne de caractères, structure etc). Cela signifie que nous pourrions intégrer n'importe quel type d'information dans nos modèles. Cependant les réseaux de Pétri colorés présentent un inconvénient : ils ne permettent pas la modularité. Cela signifie qu'il n'est pas possible de définir un réseau de Pétri comme un module avec des entrées et des sorties, et qu'il n'est pas possible de le réutiliser ou le dupliquer. Imaginons un atelier de production qui possède plusieurs machines identiques, avec les réseaux de pétri colorés il faudra refaire pour chaque machine le même modèle de réseau de Pétri, ce qui peut prendre beaucoup de temps et en plus de cela engendrer des erreurs de conception. Pour résoudre ce problème les réseaux de Pétri colorés Modulaires et Hiérarchiques ont été développés. Nous allons les présenter dans le prochain paragraphe.

## Les RdPC Modulaires et Hiérarchiques

Les RdP hiérarchiques ont été formalisés dans le but de manipuler les réseaux de pétri et de les utiliser comme des objets ou des modules. Les réseaux de Pétri Colorés Hiérarchiques (RdPCH) permettent une manipulation modulaire d'un RdP colorés. Ainsi un réseau de Pétri Coloré hiérarchique est un ensemble de réseaux de Pétri Modulaires. Commençons tout d'abord par définir le formalisme des Réseaux de Pétri Coloré Modulaires (RdPCM).

Dans [53], un RdPM est défini comme un quadruple.

$CPN_M = (CPN, T_{sub}, P_{port}, PT)$  avec :

- $CPN = (P, T, A, \Sigma, V, C, G, E, I)$  un réseau de Pétri coloré.
- $T_{sub} \subseteq T$  un ensemble de transitions de substitution.
- $P_{port} \subseteq P$  est un ensemble de ports d'entrée ou de sorties de places.
- $PT : P_{port} \rightarrow \{IN, OUT, I/O\}$  est une fonction d'assignation de type de port qui associe à chaque port de place un type.

Un RdPCM permet de transformer un RdP Coloré en un système d'entrées et de sortie. De cette manière le réseau pétri se comportera comme une boîte noire avec des entrées et des sorties qui seront des places. Lorsque les places jouent le rôle d'entrée ou de sortie elles ont une fonction de ports d'entrée et/ou de sortie. Prenons pour exemple le Réseau de Petri de la figure 1.7. Nous allons le transformer en un Réseau de Petri Modulaire.

Nous allons considérer la places A comme une place d'entrée et la place D comme une place de sortie. La figure 1.8 est une illustration de ce choix. Cela signifie que les places autres que A et D (les places B,C et E seront considérées comme intérieures au réseau de pétri modulaire) Les places qui ne sont pas considérées comme des ports d'entrée ou de sortie du Réseau de pétri modulaires n'apparaîtront pas dans la représentation du réseau de Petri modulaire. De cette manière le réseau de pétri sera réduit de telle sorte que l'on

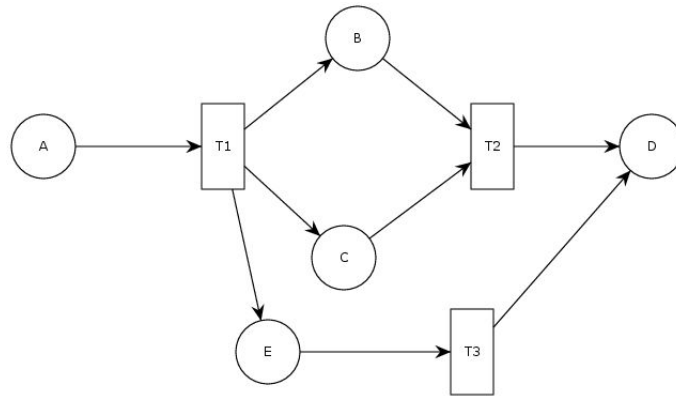


FIGURE 1.7 – Exemple d'un RdP coloré pour illustrer les RdPCM

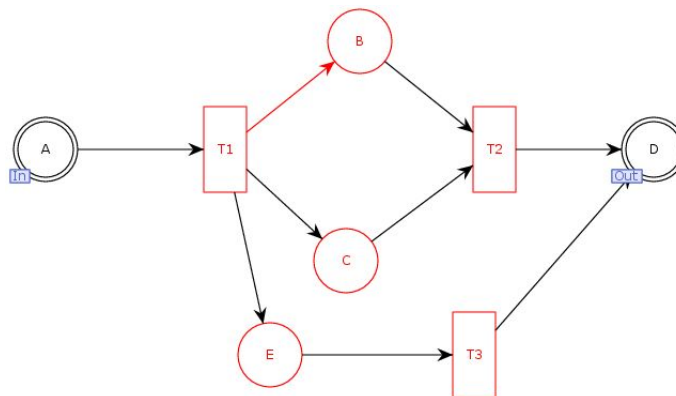


FIGURE 1.8 – Exemple d'un RdP coloré pour illustrer les RdPCM

obtienne la figure 1.9. Sur la Figure 1.9 on ne peut voir que des places n'ayant pas de fonction de port ainsi que des transitions du réseau qui ont été réduites à la transition M. Cette dernière est appelée transition de substitution.

Lorsque l'on observe les Figures 1.7 et 1.9 on observe que ces deux réseaux sont identiques du point de vue fonctionnel et même des propriétés liées au RdPC, la seule différence réside au niveau de la vision que l'on a de ces deux réseaux, en effet la Figure 1.7 présente une vue détaillée du RdPC alors que la Figure 1.9 présente une vue hiérarchisée du RdPC. C'est l'atout majeur des RdPCM, car il permettent d'avoir une vue hiérarchisée d'un RdPC. De plus un RdPCM permet de réutiliser plusieurs fois un module qui a déjà été prédéfini. Supposons que le RdPC de la Figure 1.7 représente une unité de production et que les places  $A$  et  $D$  représente les entrées et les sorties des produits de cette unité de production. Si l'on souhaite rajouter une seconde unité de production identique à la première, deux possibilités se présentent :

- Reproduire le réseau de la Figure 1.7
- Utiliser le RdPCM développé et présenter à la Figure 1.9

Ces deux solutions sont équivalentes mais l'utilisation de la deuxième solution rend le réseau moins encombré et plus lisible. les réseaux de Pétri Colorés Modulaires permettent

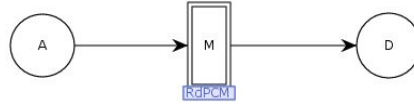


FIGURE 1.9 – le RdPCM obtenu à partir du RdPC de la figure 1.7

de construire les réseaux de Pétri colorés Hiérarchiques. En effet un réseau de pétri Colorés Hiérarchiques est un réseau constitué d'au moins un réseau de Pétri Modulaires. L'utilisation dans un RdPC de plusieurs RdPCM a conduit à la définition des RdPC Hiérarchiques

Un RdPC Hiérarchique [53] est un quadruple.

$CPN_H = (S, SM, PS, FS)$  où :

- $S$  est un ensemble de modules. Chaque module est un RdPCM

$$s = ((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I), T_{sub}^s, P_{port}^s, PT^s)$$

tel que :

$$(P^{s_1} \cup T^{s_2}) \cap (P^{s_2} \cup T^{s_1}) = \emptyset \text{ pour tout } s_1, s_2 \in S \text{ tel que } s_1 \neq s_2$$

- $SM : T_{sub} \rightarrow S$  est une fonction de sous-module qui à un sous-module associe une transition de substitution.
- $PS$  c'est la fonction de relation port-connecteur

Plus simplement nous pouvons dire que un RdPC hiérarchique est un réseau dans lequel les transitions sont des Réseaux de Pétri colorés modulaires. Si nous avons choisi d'utiliser les Réseaux de pétri Hiérarchiques, c'est parce que nous avons l'intention de définir une collection d'objets que nous allons pouvoir ré-utiliser et agréger afin de modéliser les différentes composantes d'un système de production flexible, afin de simuler le fonctionnement global d'un système.

Dans la mesure où nous allons devoir développer des modèles de simulation, il est important pour nous d'aborder la problématique de la simulation des systèmes de production flexibles et de présenter quelques plateformes de modélisation et de simulation des réseaux de Pétri.

## La simulation des systèmes de production

*La simulation d'un système réel devient utile dès lors que la réalisation d'un modèle mathématique de ce système est, soit trop coûteuse en temps de conception et de calcul, soit trop simplificatrice de la réalité, de sorte que le modèle obtenu ne permet plus d'appréhender les informations utiles sur le système à étudier [54]* Les forces de la simulation des systèmes de production [55] sont :

- La compression temporelle : en effet, il est possible de simuler la production en plusieurs jours d'un système en quelques minutes voire même des secondes ;
- Intégration de composants : la capacité d'intégrer des composants de systèmes complexes d'étudier leurs interactions ;
- Limitation des risques : les systèmes potentiellement dangereux peuvent être étudiés sans les risques financiers ou physiques qui peuvent être impliqués dans la construction et l'étude de tels systèmes réels ;
- La mise à l'échelle physique : c'est la possibilité d'étudier des versions beaucoup plus grandes ou petites d'un système ;
- La répétabilité : c'est la possibilité d'étudier des différents systèmes dans des environnements identiques ou le même système dans des environnements différents ;
- Le contrôle : tout dans un environnement simulé peut être contrôlé avec précision et exactitude.

La simulation est couramment utilisée en gestion de production comme solution aux problématiques suivantes :

- L'ordonnancement des ateliers de production [56, 57, 58, 59, 60]
- L'évaluation des performances [61, 62, 63]
- simulation et analyse des systèmes de production [64, 65]

Le processus de simulation d'un système de production comprend trois grandes étapes, ces étapes sont expliquées dans les travaux de [2], les auteurs cités divisent ce processus en trois parties :

- La phase de conception : c'est la phase au cours de laquelle les besoins et les contraintes du système de production sont identifiés. Les besoins en terme de modélisation à savoir qu'est-ce que l'on souhaite modéliser ou observer. En ce qui concerne les contraintes nous pouvons citer par exemple les contraintes liées aux processus de fabrication des produits. La phase de conception est aussi l'occasion de déterminer la nature des modèles qui seront utilisés.

- La phase de développement : cette étape intervient après la phase de conception elle comprend le choix de l'approche de modélisation, la construction du modèle, ainsi que les étapes de vérification et de validation. En ce qui concerne le choix de l'approche de modélisation il faut noter que dans le cadre de la simulation des systèmes de production il existe deux visions en termes de modélisation et de simulation :
  - Une vision centrée sur le produit. Encore appelée "*Job-driven view*" la vue centrée sur le produit considère au cours de la simulation le produit comme un objet actif alors que les ressources elles sont considérées comme passives (machines). Cette vision permet de suivre tout au long de la simulation le produit. Ainsi les produits s'emparent des ressources dont ils ont besoin et les libèrent lorsqu'ils ont achevé leur process.
  - Une vision centrée sur les ressources appelée "*resource driven view*", les ressources sont considérées actives et les produits eux passifs. Dans cette configuration les produits sont déplacés et transformés par les ressources.

Les deux vues que nous venons d'énoncer ont leur avantage et leur inconvénients. Dans [66], l'auteur après avoir fait la comparaison de deux modèles basés l'un sur une vision centrée sur le produit et l'autre sur une vision centrée sur les ressources arrive, aux observations suivantes :

- La simulation focalisée sur les ressources est beaucoup plus rapide que celle sur les produits.
  - La simulation centrée sur les produits permet de suivre avec précisions les étapes de fabrication d'un produit tout au long de production, cependant cela demande aussi pour le système d'enregistrer toutes les données relatives à un produit et ce tout au long de la simulation. Cela a pour conséquence de ralentir la simulation au cas où dans un système de production il y a beaucoup de produits.
  - La simulation centrée sur les ressources est plus rapide que celle qui est centrée sur les produits, mais elle devient difficile à mettre en œuvre dans le cas d'une forte personnalisation des produits, c'est pourquoi l'auteur rappelle d'ailleurs qu'il est préférable d'avoir une approche de simulation centrée sur les produits si l'on souhaite simuler un système de production à faible volume de production et ayant un haut niveau de mixité des produits.
- .
- La phase de déploiement : La phase de déploiement consiste à l'expérimentation du modèle développé sur un cas réel. Cette phase a pour rôle d'extraire des données du modèle de simulation afin de prendre des décisions. C'est une phase qui peut être plus ou moins longue en temps de simulation. Cela est dû au choix de l'approche de simulation (centrée sur le produit ou sur les ressources), mais aussi de la complexité du système en lui même.

Le tableau 1.3 récapitule les étapes de la simulation d'un système de production.

Étapes du processus de simulation	Sous-étapes du processus de simulation
La conception	<ul style="list-style-type: none"> <li>• Identification des besoins et des contraintes</li> <li>• Développement du modèle conceptuel</li> <li>• Planification du projet</li> </ul>
Le développement du modèle	<ul style="list-style-type: none"> <li>• Choix d'une approche de modélisation ;</li> <li>• Construction et test du modèle ;</li> <li>• Vérification et validation du modèle</li> </ul>
Le déploiement du modèle	<ul style="list-style-type: none"> <li>• Expérimentation du modèle sur un cas réel ;</li> <li>• Analyse des résultats ;</li> <li>• Implémentation des résultats pour la prise de décision</li> </ul>

TABLE 1.3 – Les étapes du processus de simulation des systèmes de production [2]

On peut y voir les différentes étapes de la simulation d'un système de production que nous n'avons pas expliqué en détail dans cette section. Les travaux de [2] donnent plus d'explications sur ces phases. Nous allons présenter la plateforme de simulation que nous avons utilisée pour la modélisation et la simulation de nos modèles.

## Les plateformes de simulation des réseaux de pétri

Lorsque l'on a un RdP qui a une très faible complexité c'est à dire un faible nombre de places, de transition et de nœuds du graphe de marquage qui peuvent être déterminés manuellement par une personne<sup>2</sup>, l'utilisation d'un outil informatique peut ne pas avoir d'intérêt. Par contre, lorsque le réseau devient complexe les plateformes de simulation deviennent indispensables pour les raisons suivantes :

- Vérification des propriétés des RdP que nous avons évoquées jusqu'ici (bornitude, vivacité) : la vérification permet l'analyse du réseau et peut prévenir certains blocages du réseau.
- Simulation et suivi de l'évolution du RdP et extraction des informations du modèle.
- Détermination des graphes de marquage des RdP à forte complexité.
- Test de la stabilité du RdP en modifiant le marquage initial.

<sup>2</sup>. La notion de complexité étant relative, l'on peut dire que la capacité à déterminer manuellement le graphe de marquage d'un RdP dépend aussi du temps passé et la capacité de la personne, même si l'on peut raisonnablement penser qu'il puisse être difficile de déterminer manuellement les graphes de marquage d'un réseau qui comporte plus de 1000 nœuds.

Plusieurs logiciels de simulation de RdP ont été développés pour permettre de répondre aux besoins exprimés auparavant. Vu la spécification des applications, l'absence de logiciel universel de simulation des RdP et l'accès relativement simple au développement informatique encourage les utilisateurs à développer leur propre simulateur de Réseau de Pétri. Il y en a quelques uns dans les travaux de [67] et de [68]. D'autres logiciels de modélisation et simulation basés sur les RdP sont utilisés dans la recherche [69],[70] et surtout dans le monde industriel [71].

Dans ce paragraphe nous allons lister quelques plateformes de modélisation et de simulation basées sur les RdP, La table 1.4 présente une liste non exhaustive des plateformes de simulation des RdP et les classe par type de RdP qu'elles supporte. [72]

TABLE 1.4 – Liste de plateformes de modélisation et de simulation des RdP

Outils de modélisation	RdP pris en charge
LoLA ; JFern ; JARP ; IOPT-Tools ; Income Suite ; INA ; HISIm ; GreatSPN ; Geist3D ; GHENeSys ; GDToolkit ; FLOWer ; ARP ; Artifex ; CPN-AMI ; Disc Software Platform ; ePNK ; ExSpect ;	RdP places-transition
JFern ; Income Suite ; INA ; HISIm ; GreatSPN ; Geist3D ; GHENeSys ; F-net ; ALPHA/Sim ; ARP ; CPN Tools ; ExSpect ;	RdPtemporisés
LoLA ; JFern ; Income Suite ; INA ; HISIm ; HiQPN-Tool ; Helena ; GreatSPN ; Geist3D ; GDToolkit ; ALPHA/Sim ; ALPiNA, Artifex ; CoopnBuilder ; COSA BPM ; CPN-AMI ; CPN Tools ; ePNK ; ExSpect ; GDTToolkit ; GHENeSys ;	RdP de haut niveau
QPME ; PROTOS ; Predator ; PACE ; ORIS ; MISS-RdP ; HISIm ; HiQPN-Tool ; GreatSPN ; F-net ; ExSpect ;	RdP Stochastiques

En complément de la Table 1.4, une liste encore plus détaillée est donnée par sur le site [72]. Le choix d'une plateforme de simulation par rapport à une autre dépend, hormis le coût, d'autres facteurs tel que :

- **Le type de réseau pris en charge** : en fonction de la spécificité du réseau utilisé pour optimiser le problème, on peut être amené à utiliser des réseaux de Petri places-transition ou encore des réseaux de pétri de haut niveau. Il faut dans ce cas choisir une plateforme qui peut prendre en charge le type de réseau que l'on va utiliser.
- **La capacité d'analyse** : la capacité de la plateforme de modélisation à déterminer les propriétés du réseau est très importante, cela permet d'assurer le bon fonctionnement du réseau lors de la simulation et éviter des blocages. Les outils qui permettent une extraction plus facile des résultats et un affichage plus clair sont souvent plus utilisés.

- **Le contrôle du réseau de pétri lors de la simulation** : une des problématiques récurrentes avec les RdP étant la franchissabilité simultanée des transitions, une plateforme de simulation de réseau devrait permettre, dans ce type de situation, de mettre en place une stratégie afin de définir la priorité de déclenchement lorsque deux transitions sont franchissables.
- **Une communauté d'utilisateurs et de développeurs active** : cela permet d'avoir d'une part un logiciel souvent mis à jour, des versions stables, et aussi de pouvoir trouver de l'aide en ligne en cas de difficulté à travers des tutoriels, vidéos et forum.

Les critères que nous venons de citer ainsi que les recommandations des chercheurs collectés lors de notre état de l'art, nous ont permis de faire le choix de CPN TOOLS comme outil de simulation. La raison qui a confirmé notre choix était le fait qu'en 2012 au moment où nos travaux de thèse ont débuté CPN TOOLS existait déjà depuis plusieurs années et était encore en développement par un laboratoire de recherche. De plus les chercheurs du CPN Group de l'Université d'Aarhus en plus du site en ligne qu'ils ont développé, ont très souvent répondu à nos sollicitations quant à la prise en main de la plateforme de simulation CPN TOOLS. Nous la présentons brièvement dans la section suivante.

## Présentation de CPN Tools

CPN TOOLS est un outil de modélisation et de simulation des systèmes à événements discrets modélisés avec les RdP. CPN Tools a été développé par le *CPN Group* de l'Université d'Aarhus entre 2000 et 2010. Les principaux chercheurs qui l'ont développés sont Kurt Jensen, Soren Christensen, Lars M. Kristensen, et Michael W. [73]

C'est un outil qui permet de représenter les différentes familles de RdPC que nous avons décrit tout au long de ce chapitre à savoir les RdP simples, les RdP temporisés, les RdPC et aussi les RdP Stochastiques. De plus, CPN Tools permet aussi de simuler les RdPC Hiérarchiques. Le logiciel CPN TOOLS s'appuie sur la programmation fonctionnelle à l'aide du langage *SML/NJ* pour définir les ensembles colorés, les fonctions des arcs ; les fonctions de garde et les fonctions d'action des transitions. En plus des raisons que nous avons évoqué précédemment, nous avons choisi d'utiliser CPN TOOLS pour l'implémentation et la simulation des modèles développés dans le cadre de cette thèse parce qu'il permet les fonctionnalités suivantes :

- Définition libre des couleurs à travers la définition d'ensembles colorés : En Effet, CPN TOOLS permet de définir librement à l'aide de langage SML des structure de données complexes. Cependant il faut relever que la définition de structure de données complexes, impose aussi d'écrire des fonctions d'accessibilité et de modification de ces dernières.
- CPN TOOLS permet de définir tous les types de réseaux de Pétri que nous avons définis jusqu'ici à savoir les réseaux de Pétri simples, temporels, colorés, et stochastiques et hiérarchiques.



- CPN TOOLS propose une méthode pour la gestion de la franchissabilité simultanée des transitions. En effet, la plateforme de simulation permet lors de la définition d'une transition de lui attribuer un niveau de priorité allant de 100 pour la plus prioritaire à 1000 pour la moins prioritaire. La mise en place des priorités entre les transitions impose une analyse approfondie afin de classer les transitions en fonction de leur ordre de priorité
- Analyse des propriétés d'un RdPC à l'aide de graphes d'états et des propriétés telles que la vivacité ou la bornitude.

Cependant il faut tout de même noter que le problème de gestion des transitions franchissables en un même instant reste difficile à gérer parce qu'il nécessite que l'on hiérarchise les priorités dans tous les cas de figures. Cela signifie que dans certains cas de figure on peut être amené à mettre en place des stratégies de gestions des transitions. Dans la mesure où nous allons utiliser CPN TOOLS pour présenter nos modèles notamment dans les chapitres 2 et 3, nous allons présenter dans le prochain paragraphe la plate forme CPN TOOLS ainsi que son fonctionnement à travers des exemples simples.

La modélisation et la simulation avec la plateforme CPN Tools nécessitent la connaissance des règles de programmation fonctionnelle du standard SML (Standard Meta Language), cela permet de comprendre la définition des ensembles colorés, ainsi que celle des fonctions associés aux arcs et aux gardes. Le langage SML/NJ (Standard ML of New Jersey) est un compilateur et un environnement de programmation pour le langage de programmation Standard ML [74]. Le Standard ML est un langage de programmation fonctionnelle et impérative dans le sens où la plupart des fonctions mathématiques y est présent [75]. Le langage SML/NJ est un langage fortement typé.

Nous n'allons pas présenter dans ce paragraphe le langage SML mais uniquement quelques éléments de sémantiques afin de pouvoir créer un RdP. Un réseau étant constitué de places et de transitions, nous allons expliquer comment définir à l'aide de CPN TOOL non seulement les places et les transitions mais aussi les fonctions des arcs, les fonctions de garde, et les fonctions d'action des transition.

#### 1.5.4.1.1 Les types de données

CPN tools emprunte au Standard ML ses types de données de bases à savoir l'ensemble des réels, des entiers naturels, des chaînes de caractère, du temps, des booléens , des entiers infinis et enfin l'ensemble unité ( l'ensemble unité est un multi-ensemble construit à partir de l'ensemble  $\{1\}$ . (une liste avec des explications quand on a besoin sera plus approprié ici)

#### 1.5.4.1.2 Création d'une place

La Figure 1.10 représente une place définie sur CPN TOOLS. On peut y voir les caractéristiques de cet élément qui sont : le nom de la place, le type de jetons contenus dans la place, ainsi que le marquage initial de la place. Le nom de la place ne doit pas contenir de caractères spéciaux. Le type de jetons que doit contenir la place représentée ici par

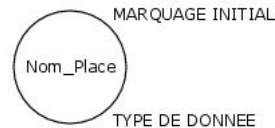


FIGURE 1.10 – Illustration d'une place sur CPN TOOLS

**TYPE DE DONNEE** n'est donné ici qu'à titre illustratif . CPN TOOLS possède des types de données standards à savoir le type entier, réel, booléen.

#### 1.5.4.1.3 Définition des ensembles colorés

Lorsque l'on utilise CPN Tools la définition d'un ensemble coloré se fait par la syntaxe suivante :

`Colset , <nom de l'ensemble coloré>= <type de donnée ou structure de donnée>;`

Le terme `Colset` est un mot clé obligatoire prédéfini et interprété de manière particulière par le compilateur, le `type de donnée` ou la `structure de donnée` doit être défini avant la définition de l'ensemble coloré. Les structures de données quant à elles peuvent être déjà définies par un code SML/NJ ou par l'utilisateur. . CPN TOOLS possèdent des types de données simples déjà prédéfinis, mais l'utilisateur peut aussi décider de créer ses propres structures de données.

#### 1.5.4.1.4 La définition de quelques structures de données

Les structures de données de base de CPN Tools sont celles issues de SML/NJ à savoir : les types de données dits simples à savoir : entiers, réels, booléens, les chaînes de caractères et les unités de temps. Le tableau 1.5 donne la syntaxe des types de données simples à l'aide de CPN TOOLS.

TABLE 1.5 – structure de données simples sur CPN TOOLS

structure de données	syntaxe CPN TOOLS
entiers naturels	INT
réels	REAL
Temps	TIME
Chaîne de caractère	STRING
valeur booléenne	BOOL

Ces types de données sont définis sous CPN TOOLS de la manière suivante :

```
colset UNIT = unit;
colset BOOL = bool;
colset INT = int;
colset INTINF = intinf;
colset TIME = time;
```

```
colset REAL = real;
colset STRING = string;
```

Si l'utilisateur souhaite utiliser des structures de données plus complexes, ce dernier se doit d'utiliser des structures complexes issus du langage SML/NJ. On peut citer :

#### 1.5.4.1.5 Le tuple

Il représente le produit cartésien de plusieurs ensembles colorés. Pour définir un ensemble coloré comme un produit cartésien de plusieurs ensembles colorés il suffit d'utiliser comme structure de donnée le type `product`.

La syntaxe de la représentation d'un ensemble coloré avec ce type de structure de donnée est :

```
colset Nom_ensemble_colore = product type_donnee1 * ... * type_donneeN
```

Cela signifie que nous avons défini un N-uplets constitué de N types de donnée différents.

Par exemple définissons l'ensemble coloré des couples de réels de la forme  $(a, b)$  avec  $a, b \in \mathbb{R}$  sera :

```
colset RxR= product REAL*REAL;
```

. Cet ensemble coloré est construit à partir de l'ensemble coloré des réels défini plus haut

Pour accéder à une composante d'une variable de type `product` il faut utiliser le caractère `"#"` suivi du numéro de la position de cette composante dans le tuple. A titre d'exemple supposons une variable

```
a= (1,3,4,5)
```

si l'on souhaite accéder à la troisième composante de la valeur `a` c'est à dire `4`, il suffit de faire :

```
#3 a
```

#### 1.5.4.1.6 Le type record

Le type record peut être compris comme le type tuple dans la mesure où il est permet de représenter le produit cartésien de plusieurs ensemble. La seule différence est dans la manière de l'exprimer et d'accéder à la donnée.

La syntaxe de la déclaration d'un ensemble coloré de type record est :

```
colset Nom = record nom1:type_donnee1 * ... * nomN:type_donneeN
```

Cette syntaxe nous montre que à la différence de la structure `product` il est nécessaire de donner un nom à chaque champ de la structure. Le nom qui est donné va permettre d'accéder à chaque champ de la structure en utilisant le signe `"#"` suivant du nom du

champ. Si l'on souhaite accéder à la valeur contenue dans `nom1` il faut utiliser la syntaxe `(#nom1, Nom)` Pour reprendre notre exemple précédent on pourrait redéfinir l'ensemble coloré `RxR` de la manière suivante :

```
colset RxR1= record reel1: REAL* reel2:REAL;
```

### 1.5.4.1.7 L'union

Cette structure de donnée représente l'union d'un ou plusieurs multi ensembles colorés. Pour définir un ensemble coloré comme l'union d'un ou plusieurs ensembles colorés l'on utilise la structure de donnée `union`

### Les jetons

Les jetons sont les éléments contenus dans les places. L'utilisation des RdP colorés impose la définition de multi ensemble, cela signifie que dans une même place on peut retrouver deux jetons identiques sans que cela ne pose un problème sémantique. Soit une place **P**, le nombre de jetons dans cette place sera :

$$m(P) = n(p_1)'p_1 + +n(p_2)'p_2 + + \dots + +n(p_i)'p_i + + \dots n(p_N)'p_N \quad (1.1)$$

$n(p_i)'p_i$  est le nombre de jetons  $p_i$  contenu dans la place **P**. Le signe "++" est une loi du multi ensemble définie telle que

$$\begin{aligned} a'x + +b'y &= (a + b)'z \Rightarrow x = y = z \\ a'x + +b'y &= a'x + +b'y \text{ si } x \neq y \end{aligned}$$

Le nombre de jetons contenus dans la place **P** sera donc :

$$\sum_{i=1}^N n(p_i)$$

Nous pouvons voir sur la Figure 1.11, le nombre de jetons d'une place sous CPN TOOLS. Sur la figure nous pouvons voir que la place **A** qui est de type entier contient 1 jeton de

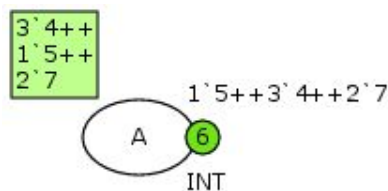


FIGURE 1.11 – Nombre de jetons d'une place sous CPN TOOLS

type 5, 3 de type 4 et 2 de type 7. Le nombre de jetons de la place **A** est de 6. Nous avons à faire ici à une place de type entier mais cette définition s'étend au place de type `STRING`, ou même de type plus complexe. Sur la Figure 1.12, on peut voir que le place **B** contient 6 jetons de type chaîne de caractères à savoir le jeton "bonjour", deux jetons "Madame" et 3 jetons "Monsieur".

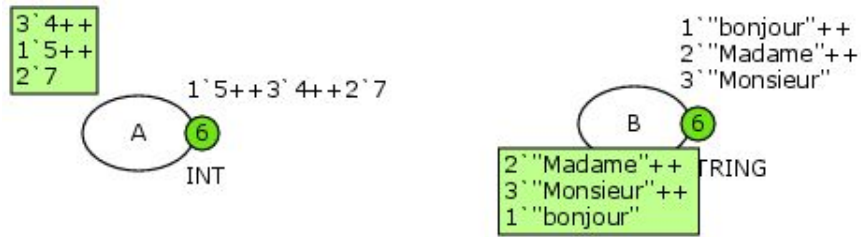


FIGURE 1.12 – Jetons de types entiers et de type chaîne de caractère sous CPN TOOLS

### Définition des variables et des fonctions

La définition des variables permet uniquement de pouvoir pondérer les arcs comme nous allons le voir dans le prochain paragraphe. Les variables sont des éléments d'ensembles colorés qui ont été définis au préalable. La syntaxe est la suivante :

```
var Nom_variable: Colset_Type;
```

le mot `var` est un mot clé, `Nom_variable` est le nom de la variable et `Colset_Type` un ensemble coloré précédemment défini. La définition d'une variable `r` réelle sera

```
var r: REAL;
```

La définition des fonctions a été emprunté au langage SML/NJ et se traduit par la syntaxe suivante :

```
fun Nom_Fonction (var1:Colset_Type1,...,varN: Colset_TypeN):Colset_Result=
{ Instructions}
```

`var1, ..., varN` sont les `N` variables de type respectivement `Colset_type1, ..., Colset_typeN`. la fonction `Nom_Fonction` renvoie une variable de type `Colset_result`. L'utilisation des fonctions et des variables permet de définir les poids des arcs lors de leur création.

### La création des arcs

Un arc relie une transition à une place ou inversement. Cela suppose qu'avant de créer un arc il faudrait déjà avoir fait de même pour les places et les transitions. Nous allons volontairement expliquer la définition des arcs avant celle des transitions. Nous allons notamment nous attarder sur la définition des fonctions des arcs. Avec le formalisme des RdP colorés les poids des arcs sont remplacés par des fonctions, il en est de même sous CPN TOOLS, où le poids de chaque arc est une fonction écrite sous le langage SML. Cependant, l'utilisation de fonctions au sens mathématique, c'est à dire, avec des variables de départ et d'arrivée n'est pas un impératif, on peut se contenter d'assigner une variable à chaque arc comme cela est présenté sur la Figure 1.13.

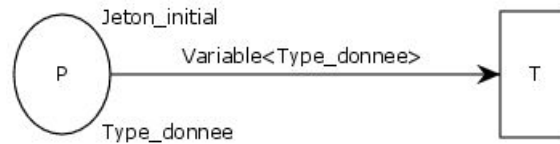


FIGURE 1.13 – Description d'un arc sous CPN TOOLS

La figure 1.13 représente un arc reliant une place amont de type **Type\_donnee** à une transition T. Le poids de l'arc la variable de nom **Variable** et de type **Type\_donnee**. Dans le cas où le poids de l'arc est une variable qui doit être obligatoirement du même type que la place. Dans le cas où le poids de l'arc sera une fonction cette fonction doit nécessairement avoir des variables du même type que la place amont.

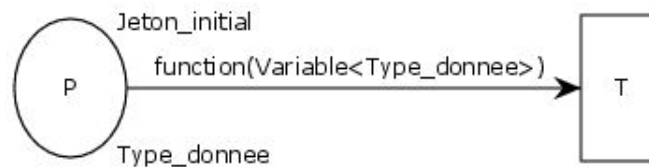


FIGURE 1.14 – illustration d'un arc pondéré par une fonction sous CPN TOOLS

Sur la Figure 1.14 l'arc est pondéré par une fonction de nom **fonction** et qui prend en entrée la variable **Variable** de type **Type\_donnee**. Ce que renvoie cette fonction n'a pas vraiment d'importance pour le compilateur, du moment que la variable **Variable** est de type **Type\_donnee**. La sémantique de pondération d'un arc reliant une place amont à une transition diffère un peu de celle reliant une transition à une place aval. En effet, le poids d'un arc reliant une transition T à une place aval P reste toujours une variable ou une fonction mais sous deux conditions :

- Dans le cas où l'arc est pondéré par une variable, cette dernière doit être du type de la place aval(cf. exemple de la figure 1.15).
- Dans le cas où l'arc est pondéré par une fonction, peu importe le type des variables en entrée de la fonction, cette dernière devra toujours renvoyer une valeur qui est du type de la place aval(Cf. exemple de la figure 1.16).

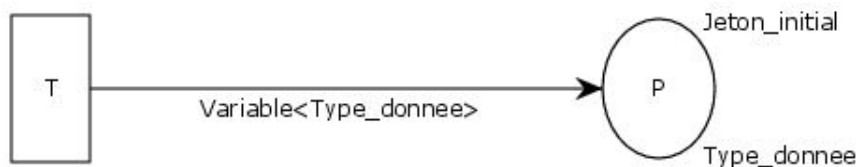


FIGURE 1.15 – Description d'un arc reliant une transition à une place aval et pondéré par une variable.

Il est important de noter que lorsqu'une variable est utilisée pour pondérer (soit de manière directe, soit par le biais d'une fonction), cette variable doit nécessairement pondérer un arc entrant à la transition. Si l'on essaye d'implémenter sur CPN TOOLS le réseau

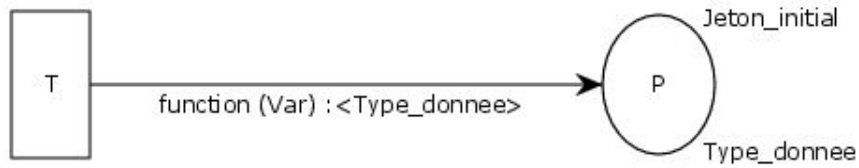


FIGURE 1.16 – illustration d'un arc pondéré par une fonction sous CPN TOOLS

que nous avons défini aux Figures 1.15 et 1.16, le compilateur renverra une erreur en nous expliquant que la variable **Variable** ou même **Var** n'est liée à aucune transition. Pour cela il faudra donc ajouter une place amont à cette transition et on obtiendra les réseaux que nous présentons à la Figure 1.18. On peut voir que toutes les variables utilisées au niveau des arcs sortant des transitions sont déjà définies au niveau des arcs entrant dans la transition T .

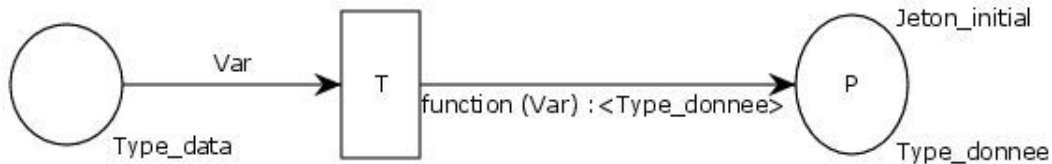


FIGURE 1.17 – Illustration des règles d'utilisation des variables au niveau des arcs sortant des transitions lorsque la place est pondéré par une fonction

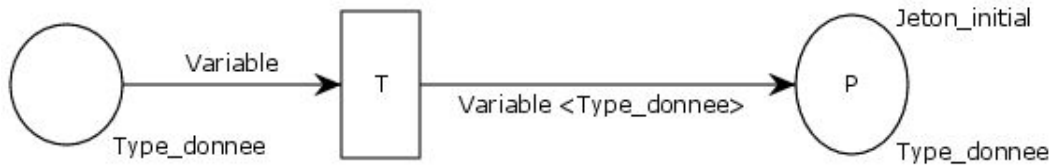


FIGURE 1.18 – Illustration de la condition d'utilisation des variables sur les arcs sortant d'une transition

### Création d'une transition

CPN TOOLS a été développé dans l'optique de modéliser et simuler les RDdPC avec une contraintes de temps. La Figure 1.19 présente les caractéristiques d'une transition sous CPN TOOLS à savoir : le nom de la transition, la fonction de garde, la fonction de temporisation, la fonction de hiérarchisation des priorités des transitions et la fonction de transfert de la transition. Toutes ses caractéristiques sont optionnelles. Le nom de la transition obéit aux mêmes règles que celle du nom des places.

- *La fonction de garde* : est une fonction qui renvoie vrai ou faux. Lorsque cette dernière est vraie alors la transition est franchissable. Évidemment elle ne l'est pas dans le cas contraire même si les conditions liées aux jetons et aux arcs sont remplies. La fonction de garde prend en entrée les variables liées aux arcs telles que nous les avons définies à la section 1.5.4.4.

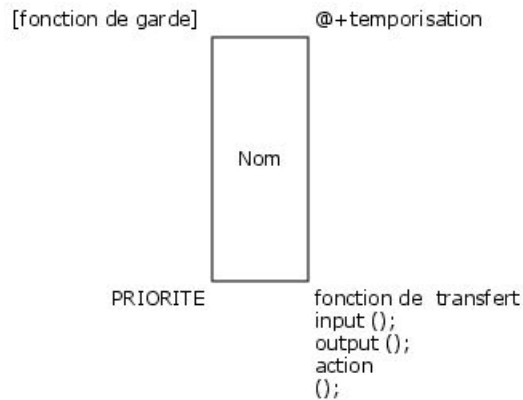


FIGURE 1.19 – Les caractéristiques d’une transition sur la plateforme CPN TOOLS

- *La fonction de temporisation* : est une fonction qui rajoute une valeur de temps **temporisation** à un jeton.
- *le paramètre d’assignation de l’ordre de priorité* : Ce dernier peut prendre des valeurs comprises entre 100(pour la transition la plus prioritaire) et 10 000(pour la transition la moins prioritaire).
- *La fonction de transfert* : A chaque transition est liée une fonction de transfert qui prend en entrée des variables liées aux arcs entrant dans la transition et qui renvoie en sortie des variables qui seront liées aux arcs de sortie. Prenons l’exemple du réseau de la Figure 1.20, on a deux places de types entier et on souhaite faire la somme des jetons de la place **X** et de la place **Y** dans la place **Z**. Nous allons utiliser la fonction de transfert de la transition T pour effectuer le calcul et renvoyer directement à travers la variable **z** la somme des jetons des places **X** et **Y** La

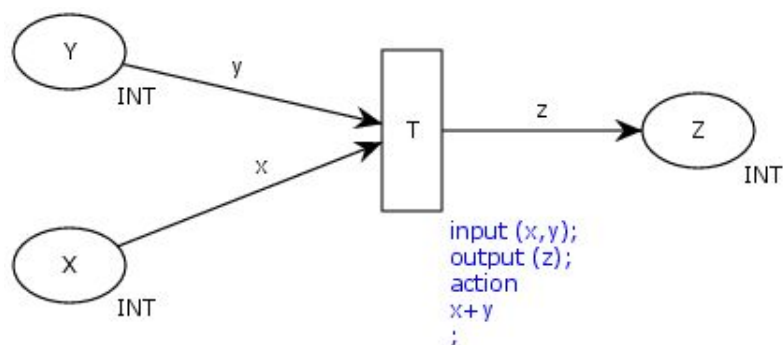


FIGURE 1.20 – Illustration de l’utilisation de la fonction de transfert d’une transition sous CPN TOOLS

fonction de transfert prend en entrée les variables **X** et **Y** et en variable sortie **Z** le mot clé **action** marque le début des opérations que vont subir les variables **X** et **Y** et qui vont permettre de stocker la somme dans la variable **Z**. L’utilisation d’une fonction de transfert peut inclure des algorithmes complexes avec de la lecture et de l’écriture dans des fichiers etc. Dans le cadre de nos travaux nous allons utiliser les fonctions de transitions notamment pour la modélisation des activités de transport.



## Conclusion

Dans ce chapitre nous avons posé les bases de notre travail en définissant dans un premier temps les systèmes de production. Nous avons par la suite présenté plusieurs classifications de ceux-ci. Nous avons accordé une attention particulière aux systèmes de production flexibles et à la notion de flexibilité qui sont au cœur de nos préoccupations. Par la suite nous avons fait le choix des réseaux de Pétri comme outil de modélisation. En effet contrairement aux modèles analytiques, les réseaux de Pétri offrent les avantages suivants :

- Une meilleure analyse des processus du fait de sa représentation graphique ;
- Une possibilité d'intégration dans un simulateur
- L'utilisation des propriétés des réseaux de Pétri (bornitude, exclusion mutuelle, vivacité) assure une représentation des problématiques de partage des ressources, mais aussi d'assurer le non blocage du modèle de simulation.

Néanmoins l'abandon des modèles mathématiques entraîne de manière inéluctable la possibilité de traiter des problèmes d'optimisation avec des méthodes exactes. Nous avons choisi comme outil de simulation des réseaux de Pétri, CPN TOOLS que nous présentons brièvement en section 1.5.4.1. CPN TOOLS nous permettra de développer nos modèles, de les vérifier et de pouvoir simuler leur fonctionnement. Dans le prochain chapitre, nous présentons les modèles que nous avons développés afin de construire notre outil d'aide à la décision à savoir les modèles des éléments d'un système de production (modèles de machines, de produits, de stocks etc.) mais aussi les modèles des activités d'un système de production (production, transport etc.).

# Chapitre 2

## Modélisation des systèmes de production flexibles à l'aide des réseaux de pétri colorés hiérarchiques

### Introduction

Dans le chapitre 1, nous avons opté pour les Réseaux de Pétri Colorés Hiérarchiques pour la modélisation de notre outil d'aide à la conception. Dans ce chapitre nous allons présenter un modèle d'atelier flexible. Celui-ci sera constitué d'un ensemble de modèles représentant les différents éléments d'un atelier flexible à savoir les machines, les produits, les stocks et les ressources de transports. Ces modèles seront présentés à la section 2.3. Nous mettons un accent particulier sur le développement des modèles de machines flexibles en y intégrant le multi-outillage (Flexibilité de la machine), et sur le développement du modèle des produits. Nous ajouterons en plus des modèles des entités du système qui représentent l'aspect structurel du système, modéliser aussi le déroulement des activités de production (assemblage, usinage, transport etc.). Parmi ces dernières, nous insisterons sur la modélisation du transport au sein de l'atelier de production. C'est ainsi que nous avons modélisé l'activité de transport à la section 2.4 en proposant deux modèles :

- Un modèle de sélection des produits à transporter ;
- Un modèle représentant le transport proprement dit.

Dans la section 2.5, nous présentons l'étape de validation de nos modèles à travers la modélisation d'une chaîne de production réelle constituée de six postes de travail, reliés entre eux par un convoyeur.

Mais avant de nous lancer, dans la modélisation à l'aide des Réseaux de Pétri Colorés Hiérarchiques, nous avons choisi de présenter en section les modèles qui ont été développés

sous le formalisme UML. Nous résumons notre démarche de travail par le schéma de la figure 2.1

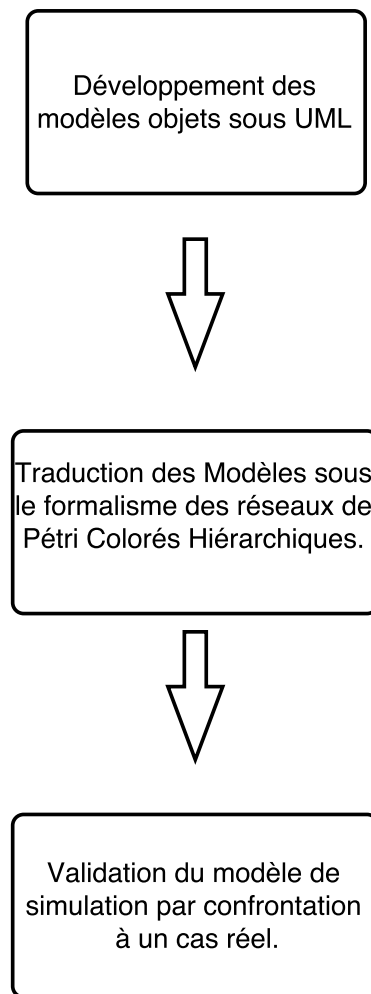


FIGURE 2.1 – Notre Démarche de travail

Le choix de présenter les modèles sous un formalisme UML permet de les rendre exploitables avec des outils de simulation autre que les Réseaux de Pétri.

## Modèles objets d'un atelier de production flexible

L'objectif de cette section est de présenter les modèles objet d'atelier flexible de production que nous avons développés à l'aide des Réseaux de Pétri Colorés Hiérarchiques. Du point de vue structurel, nous avons considéré que le système de production manufacturier est composé de machines ou postes de travail, de ressources de transport, des stocks, et des produits.

Si on adopte cette vision d'un système de production on peut développer une collection d'objets permettant de représenter la structure d'un système de production (c.f.

Figure 2.2). Les attributs et les méthodes de ces objets ont été définis pour que les modèles développés puissent représenter des systèmes de production flexibles. Nous allons présenter dans la suite les principaux éléments qui constituent la structure physique d'un système de production flexible à savoir : les produits, les stocks, les machines et les ressources de transports. Dans cette partie nous n'aborderons pas la manière dont l'organisation de la production est mise en place. Nous nous limiterons donc à un diagramme de classes, et à une représentation de celles-ci.

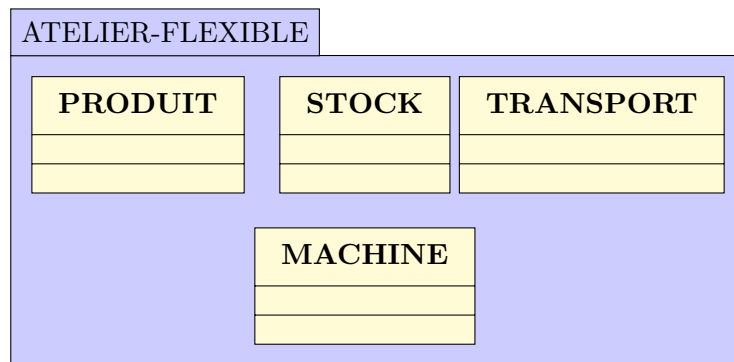


FIGURE 2.2 – Diagramme des classes d'un atelier de production flexible

Le modèle de la Figure 2.3 présente la composition d'un atelier de production flexible. Il met en évidence les relations existant entre les classes. On peut voir qu'un atelier de production flexible est composé de stocks, de machines, de produits et de ressources de transport. Un produit est constitué d'un ensemble d'opérations qui doivent être exécutées par les machines. Les machines quant à elles sont constituées d'outils et de stocks. Dans les prochains paragraphes nous allons présenter la classe produit, machines, stocks, ressources de transport, outils et opérations machines.

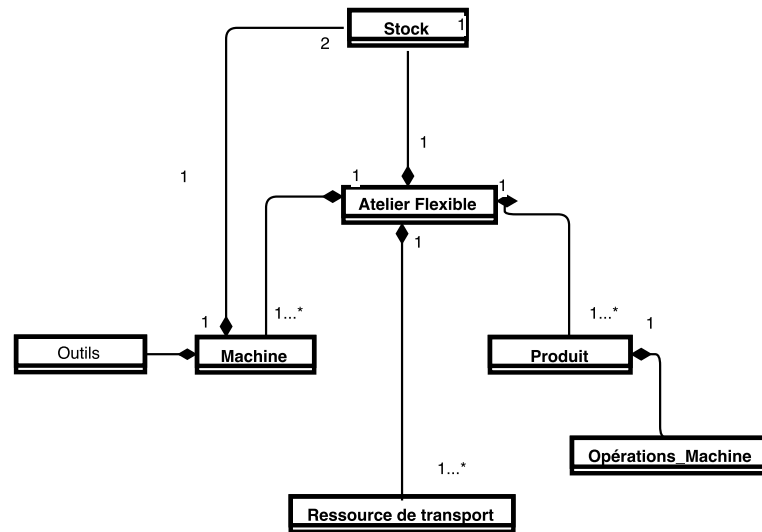


FIGURE 2.3 – Diagramme de composition d'un atelier de production flexible

## La modélisation des produits

Un produit est le résultat final d'un ensemble de processus nécessitant des ressources consommables (matières premières), des machines et des moyens de transport. Le modèle d'un produit doit permettre de reproduire le comportement de celui-ci et respecter les contraintes qui lui sont imposées par le processus de fabrication et le système de production.

Dans un système de production nous appelons " produit " les pièces finies ainsi que celles en cours de production. La fabrication d'un produit se fait par une succession d'opérations au niveau des machines ou des postes de travail. Ainsi, un produit peut être considéré comme une entité qui subit plusieurs transformations et additions ( en respectant ou non des contraintes de précédences) toute au long du processus de fabrication. La classe produit (cf. Figure 2.4) que nous présentons dans ce paragraphe doit nous permettre de répondre aux exigences suivantes en termes de traçabilité :

- Pouvoir identifier à quelle famille de produits appartient un produit.
- Différencier un produit des autres même s'ils sont de la même famille.
- Déterminer à tout moment à quelle phase de sa fabrication se trouve un produit et quelles sont les étapes qui lui reste encore à parcourir pour achever sa confection.

Afin de pouvoir identifier les familles de produits d'un système de production, nous avons décidé de classer celles-ci en "type" de produits. Ainsi dans un atelier constitué de cinq familles de produits, nous aurons les produits de type 1, 2,3,4 et 5. De cette manière nous créons un attribut "Type\_produit" à la classe "produit" qui prendra des valeurs entières. L'attribut permettra de différencier deux produits de familles différentes. Cependant, il ne permettra pas de distinguer deux produits de même famille au cours de la production. Ce qui nous ramène à notre deuxième exigence à savoir différencier deux produits même s'ils sont de la même famille.

Pour y répondre, nous allons choisir un numéro d'identification (id) unique pour chaque produit qui rentre indépendamment de la famille de produits à laquelle celui-ci appartient.

La troisième exigence qui est de pouvoir suivre l'évolution de la production étape par étape de chaque produit nous a poussé à inclure dans le modèle de Produit un attribut de List\_operations, ce dernier contiendra la liste des opérations restantes à effectuer pour achever la fabrication du produit.

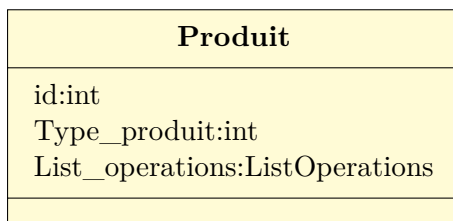


FIGURE 2.4 – Classe produit

Le modèle de produit que nous développons doit répondre à deux exigences :

- La première exigence est celle de l'utilisation de ce modèle objet dans un processus d'ordonnancement statique ou dynamique. Pour cela nous allons inclure des variables temporelles telles que la date de début de tâche, la date de fin et le retard.
- La deuxième exigence est aussi de permettre de suivre l'évolution du produit à travers l'atelier de production.

Pour répondre à la première exigence il est important de définir la vue que nous avons d'un produit. Nous allons considérer un produit comme un ensemble d'opérations nécessitant des ressources disponibles. Cette vision est cohérente dans la mesure où un produit possède une gamme de production c'est à dire un ensemble d'étapes, de procédés ou d'opérations qui concourent à sa fabrication. Si nous considérons un produit comme un travail qui peut se diviser en plusieurs opérations alors les notions telles que les dates de début, de fin et même de retard peuvent aussi être liées à la notion de produit. C'est ce qui nous conduit à améliorer le modèle de la Figure 2.4 et à proposer celui de la Figure 2.5. Le modèle du produit est enrichi en rajoutant les caractéristiques suivantes au produit :

- Date de début de fabrication du produit : c'est la date d'entrée d'un produit dans l'atelier de production.
- Date de fin souhaitée de fabrication du produit.
- Date de fin réelle

## La modélisation d'une opération

Pour modéliser les opérations que subissent les produits nous utilisons la classe d'association **Opération**. Cette dernière **Opération** permet de définir les opérations qui peuvent être effectuées par une machine sur un produit.

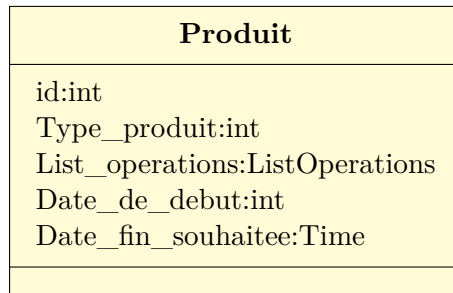


FIGURE 2.5 – Classe produit avec des variables pour les problèmes d'ordonnancement

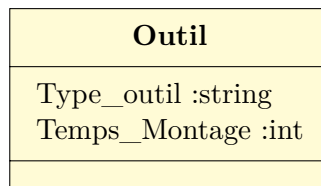


FIGURE 2.6 – classe outil

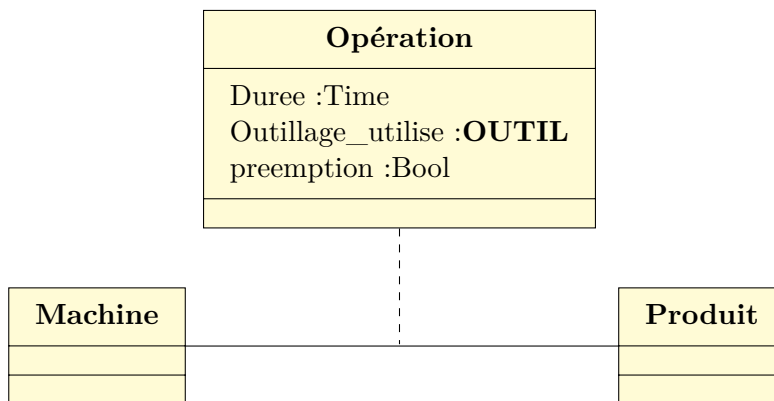


FIGURE 2.7 – Classe opération

La classe **Opération** telle qu'elle se présente signifie que chaque opération nécessite une machine et un produit. Elle n'indique pas comment les opérations sont affectées aux machines. Cela nous permet de pouvoir représenter le cas de figure d'un système de production où les opérations sont affectées aux machines avant le lancement de la production ; comme c'est le cas pour un atelier de type JOBSHOP. Cette représentation permet aussi de modéliser un système de production où les opérations ne sont pas par avance affectées aux machines comme c'est le cas des JOBSHOP flexibles ou des OPENSHOP. De plus, la classe **Opération** possède trois attributs :

- *Duree* : qui contient la durée de l'opération
- *Outillage\_utilise* qui contient l'outil utilisé pour réaliser l'opération.
- *preemption* : c'est une variable de type booléenne qui permet de définir si l'opération peut être interrompue.

## La modélisation des stocks

Dans un atelier de production flexible, il existe des stocks en entrées et en sorties des postes de travail, mais aussi en entrée et en sortie de l'atelier. Le modèle d'un stock doit traduire les caractéristiques et les fonctions élémentaires d'un stock réel, notamment le stockage et l'ordonnancement des produits, la position du stock, sa taille, sa nature, etc. Nous allons détailler dans la suite de ce paragraphe chacune des caractéristiques et fonctionnalités du modèle du stock : La Figure 2.8 présente la classe Stock avec les plus importants attributs et méthodes. Les caractéristiques du modèle STOCK sont :

<b>STOCK</b>
Nom_Stock : String Liste_produits :list <b>PRODUIT</b> Position_Stock : real $\times$ real Type_stock :{In, Out} capacite_maximale :int Capacite_previsionnelle :int Disponibilite :{0,1} Nom_machine :String
Position_stock() :Real $\times$ Real Nom_stock() :String Taille_stock() :int Distnace_de(s :STOCK) : Real Ordonnance_stock(ORD)

FIGURE 2.8 – Classe STOCK

- *Nom* : chaque stock est identifié par un attribut *Nom\_Stock* qui correspond au stock de la machine à laquelle le stock est rattaché.
- *Contenu* : Un stock contient un ou plusieurs produits. Nous avons défini la variable *Liste\_produits* de type liste permettant de contenir plusieurs identifiant de "produit".
- *Position* : Afin de gérer les temps et distances de transport, il est primordial de connaître la position d'un stock dans l'atelier de production. Nous avons rajouté un attribut définissant la position du stock dans l'atelier (attribut *Position\_stock* dans la Figure 2.8). Nous allons considérer qu'un stock a la même position que sa machine.
- *Type* : Nous avons distingué quatre types de stocks, des stocks d'entrée machine, des stocks de sortie machine, les stock d'entrée de l'atelier et les stocks en sortie atelier. Afin de différencier les types de stocks, nous avons défini l'attribut *Type\_stock* pour distinguer les stocks d'entrée et de sortie. Les stocks d'entrée et de sortie atelier sont considérés comme les autres stocks sans être liés à des machines.



- *Machine* : Chaque stock est lié à une machine à l'exception des stocks entrée et sortie de l'atelier. Nous avons donc défini l'attribut *Nom\_Machine* qui contient le nom de la machine liée à ce stock. Dans notre modèle d'atelier on trouve toujours deux stocks avec le même attribut *Nom\_Machine* (un stock d'entrée et un stock de sortie). Les stocks d'entrée et de sortie de l'atelier ont un attribut *Nom\_Machine* avec les valeurs "Entrée" et "Sortie" respectivement.
- *Taille* : Chaque stock peut contenir un nombre limité de produits. L'attribut *Capacité\_maximale* est un entier positif qui représente la taille maximale du stock. La taille instantanée du stock peut être définie à travers le calcul de la taille de la liste *Liste\_produits*. Afin d'éviter des conflits de remplissage des stocks dans le cas de plusieurs transporteurs, nous avons attribué à chaque stock une variable nommée *Capacite\_Previsionnelle* indiquant la taille du stock à sa prochaine mise à jour.
- *Disponibilité* : Un stock n'est pas toujours disponible, il peut-être rempli ou inutile lorsque sa machine est en panne par exemple. L'attribut binaire *Disponibilite* est mis à 1 lorsque le stock est disponible i.e. utilisable et possède assez de place pour contenir d'autres produits, et à 0 lorsqu'il est indisponible .

Les fonctionnalités du modèle STOCK sont :

- *Position* : La fonction *Position\_stock* permet de récupérer la position du stock. Elle est utile lorsqu'on veut placer des stocks dans l'atelier ou lorsqu'on traite le problème d'agencement spatial des machines.
- *Nom* : La fonction *Nom\_stock* permet de récupérer l'identifiant du stock qui répond à un certain nombre de conditions liés à sa position ou son contenu.
- *Taille* : La fonction *Taille\_stock* compte le nombre d'éléments dans de la liste *Liste\_produits* qui correspond
- *Distance* : La fonction *Distance\_de(s1 : STOCK)* retourne la distance qui sépare le stock concerné d'un autre stock *s1*.
- *Ordonnement* : La fonction *Ordonnance\_stock(ORD)* permet de classer les produits présents dans le stock selon une règle d'ordonnement *ORD* bien défini tel que FIFO LIFO ou autre.

## La modélisation des machines

La classe *Machine* (voir la Figure 2.9) représente un poste de travail qui peut être manuel, semi-automatique ou totalement automatique, permettant d'effectuer un ou plusieurs types d'opérations sur un produit avec des capacités *C<sub>pk</sub>* et des temps de processus différents. Nous allons considérer que chaque opération que la machine peut effectuer est liée à un outil présent dans la machine. Chaque machine possède aussi un stock d'entrée et un stock de sortie permettant de stocker les produits en attente de fabrication ou déjà transformés. un modèle d'une machine est défini par agrégation de deux classes *STOCK*

et de une ou plusieurs classes Outils. Les deux classes STOCK représentent les stocks en entrée et en sortie de la machine. Les classes outils permettent de modéliser les différents outils que possède la machine. Nous allons détailler dans la suite de ce paragraphe chacune des caractéristiques et fonctionnalités du modèle d'une machine.

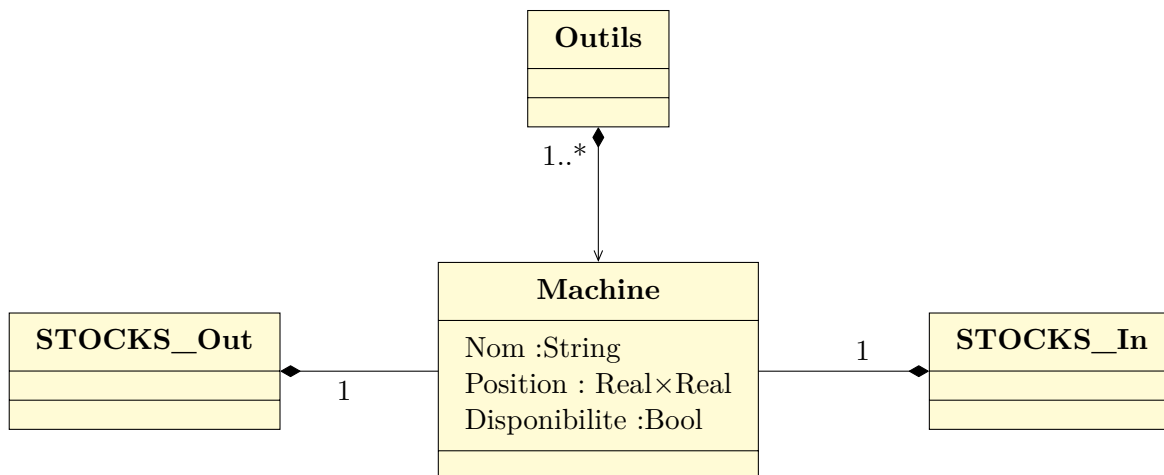


FIGURE 2.9 – classe machine

Une machine peut se trouver dans plusieurs états, comme représenté sur la Figure 2.10. En effet une machine peut se retrouver dans deux états : "Disponible" ou "Indisponible" :

- *Disponible* : cet état correspond à la situation où la machine est en attente d'une pièce pour y effectuer une opération. On suppose que la machine est fonctionnelle dans cet état.
- *Indisponible* : La machine est indisponible si on n'est pas en mesure d'effectuer une opération qu'elle est normalement capable de faire. Ceci correspond à deux situations :
  - Quand elle est déjà en train d'effectuer une opération sur un produit.
  - Quand elle est en panne ou en train de subir une opération de maintenance.

Le fonctionnement ordinaire d'une machine est qu'elle alterne entre "Libre" et "Occupée". Une machine peut passer de la situation libre vers la situation "Panne ou maintenance" quand une panne arrive pendant l'arrêt ou lorsqu'une maintenance est programmée pendant que la machine est en attente. On suppose que la machine passe toujours par la situation "Libre" après une maintenance. La machine peut aussi interrompre son fonctionnement et passer dans la situation "Panne ou maintenance" lorsqu'une panne arrive pendant le fonctionnement ou lorsqu'une maintenance urgente est programmée pour une opération interruptible.

## La Modélisation des ressources de transport

La modélisation des ressources de transport dans un atelier de production comprend à la fois la modélisation des dispositifs de transport et la modélisation de l'activité de

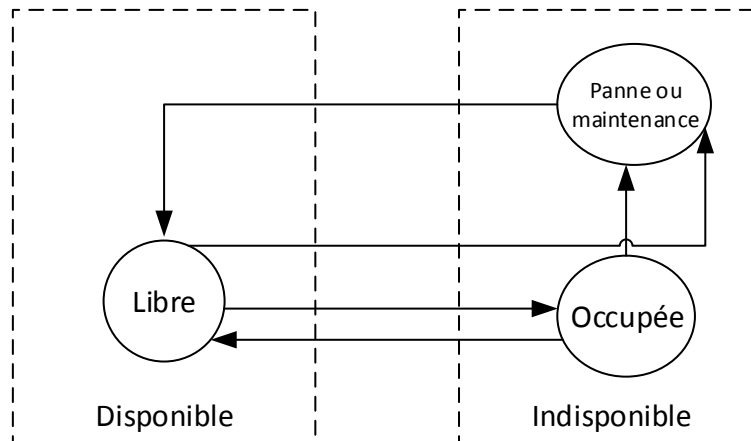


FIGURE 2.10 – États de la machine

transport des produits. Les équipements de transport sont caractérisés par leur vitesse de déplacement, leur temps de chargement/déchargement des produits et leur de chargement . nous avons rajouté d'autres indicateurs tels que la distance parcourue afin de pouvoir évaluer le coût lié au transport dans l'atelier de production flexible. L'objectif de cette partie sera dans un premier temps de présenter ce que nous entendons par dispositif de transport et aussi comment nous allons le modéliser. Nous présentons un dispositif de transport comme une classe contenant les attributs et méthodes suivants :

- *idv* : Un numéro d'identification unique qui permet de différencier deux moyens de transport.
- *position* : Représente la position instantané du transporteur dans le plan de l'atelier. Cette position va permettre de calculer la distance parcourue par le transporteur, ainsi que les temps de transport.
- *vitesse\_moyenne* : La vitesse moyenne de déplacement est évidemment indispensable pour déterminer le temps mis pour parcourir une distance. Nous avons supposé que le transporteur se déplace
- *Temps chargement/déchargement* : Les temps de chargement et de déchargement des produits dans le moyen de transport respectivement dans le stock d'entrée des machines. Ce temps dépend du transporteur, de la machine et du produit.
- *Date d'arrivée* : C'est la date à laquelle la transporteur terminera son opération de déchargement. Dans le cas où il ne transporte rien cette date est mise à zéro.

Le modèle objet d'une ressource de transport est représenté à la figure 2.11.

Une opération de transport est caractérisée par une classe d'association (cf. Figure 2.12) constituée de la classe Produit et de la classe Transport. La variable *PositionDepart*

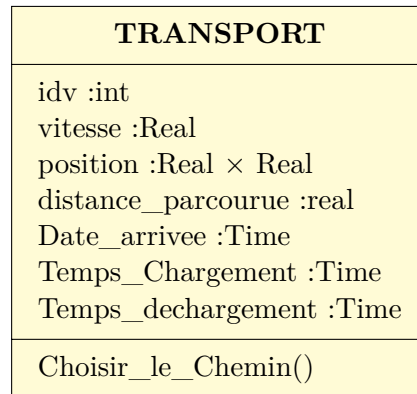


FIGURE 2.11 – Classe Transport

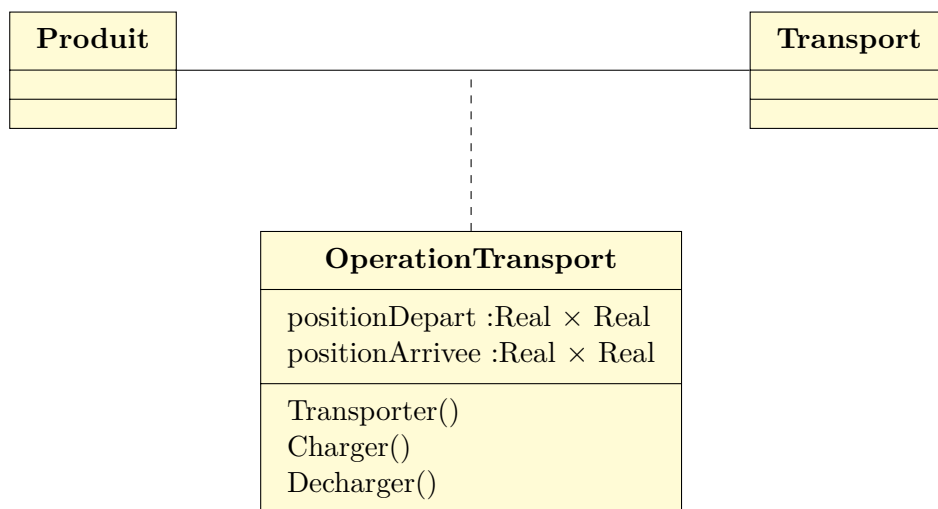


FIGURE 2.12 – Classe d'association d'une opération de transport

représente la position dans l'atelier d'une zone de chargement (stock en sortie de machine ou stock en entrée de l'atelier), et *positionArrivee* une zone de déchargement'stock en entrée des machines ou stock de sortie de l'atelier). Nous venons de présenter les modèles objets des entités d'un système de production flexible à l'aide du formalisme UML. A présent nous allons traduire ces modèles sous forme de réseaux de pétri Hiérarchiques

## Modélisation d'un système de production flexible à l'aide des Réseaux de Pétri Colorés Temporisés Hiérarchiques (RdPCTH)

L'objectif de cette section est de proposer un modèle de simulation d'un atelier de production flexible. Ce modèle sera basé sur les Réseaux de Pétri Colorés Temporisés Hiérarchiques. Nous allons nous appuyer sur les modèles objets que nous avons présentés dans la section 2.2. De cette manière nous allons faire une transcription de ces modèles objets UML en modèles Réseaux de Pétri Colorés Temporisés Hiérarchiques. Toutes les

classes objet à l'exception de la classe machine seront représentées par des multi-ensembles colorés. Nous avons expliqué au premier chapitre qu'une place était un multi-ensemble coloré. Cela signifie que dans un réseau de Pétri, une place de type  $x^1$  représentera la classe objet  $x$ . Lorsqu'une place  $x$  représente une classe objet  $x$ , alors la présence d'un jeton dans cette dernière signifie l'instanciation un objet de type  $x$ .

De cette manière, la classe **Produit** sera modélisée par l'ensemble coloré **Product**, la classe **Operation** par l'ensemble coloré **Operation**(opération machine), la classe **Stock** par l'ensemble coloré **STOCK\_IO**, et les ressources de transport par les ensembles colorés **TRANSPORT**. Le modèle d'une machine sera représenté par un réseau de pétri modulaire, parce qu'une machine est une composition de stocks (entrée et sortie), mais aussi d'outils. Nous commencerons par présenter le modèle d'un outil, ensuite d'une opération machine, après celui d'un produit, d'un stock et d'une machine flexible et enfin la modélisation d'une activité de transport.

## La modélisation d'un outil

Un outil permettra à une machine de transformer un produit en cours de production. La classe **Outil** sera traduite par l'ensemble coloré **TOOL** défini ci-dessous :

```
colset TOOL = record TypeTool :STRING*TimeOfChange :INT;
```

La définition de l'ensemble coloré **TOOL** nous montre qu'un outil aura principalement deux caractéristiques :

- La nature de l'outil représentée par la variable *TypeTool* qui définit le type d'outil, c'est à dire l'usage que l'on peut faire de ce dernier.
- Le temps de montage de l'outil représenté par la variable *TimeOfChange*.

## La modélisation des opérations machine

Pour modéliser une opération machine nous avons besoin de connaître :

- La machine sur laquelle l'opération doit avoir lieu ;
- La durée de cette opération sur la machine où elle se réalise ;
- L'outil utilisé afin de réaliser cette opération ;
- La nature interruptive de la dite opération.

Ce sont ces quatre caractéristiques que nous retrouvons dans la définition du modèle d'une opération machine représenté par l'ensemble coloré **Operation** défini de la manière suivante :

---

1. le type d'une place P renseigne le multi-ensemble qui a servi à définir la définir et par conséquent la nature des jetons qu'elle contient

```
colset Operation=product STRING*INT*TOOL*BOOL;
```

L'ensemble coloré est un quadruple d'ensembles colorés à savoir :

- **STRING** : L'ensemble coloré **STRING** correspond au nom de la machine où l'opération s'effectuera.
- **INT** contient la durée de l'opération au niveau de la machine.
- l'ensemble **TOOL** définit l'outil qui sera utilisé pour cette opération.
- **BOOL** permet de savoir si l'opération machine peut être interrompue pendant son exécution.

## La Modélisation des produits

Nous construirons un modèle de produit en nous basant sur le diagramme UML de la Figure 2.5. Pour définir un produit nous décrirons l'ensemble qui contient ces derniers. La classe objet d'un produit sera modélisée par le multi-ensemble **ITEM** défini de la manière suivante :

```
colset ITEM=record id:INT* TypeItem:INT*  
state:A* release:INT* due:TIME* Routing:ListOperations*  
Date_debut:TIME* Date_fin:TIME*  
Retard:TIME timed;
```

On observe qu'un élément de l'ensemble **ITEM** est un 9-uplets. Autrement dit l'ensemble **ITEM** est un produit cartésien de neuf ensembles. Cela signifie qu'un élément de **ITEM** que nous appellerons produit doit avoir obligatoirement neuf attributs. Ces derniers sont représentés par les variables *id*, *TypeItem*, *state*, *release*, *due*, *Routing*, *Date\_debut*, *date\_fin*, *Retard* qui stocke chacune une information importante sur le produit. .

- *id* : l'identifiant du produit ;
- *TypeItem* : le type de produit dans le cas où le produit appartient à une famille de produits donnée ;
- *state* : cette variable renseigne à quelle étape de production se trouve un produit ;
- *release* : cette variable contient la date de lancement prévisionnelle du début de la fabrication d'un produit.
- la variable *due* appartient à l'ensemble coloré **TIME** et permet de connaître à chaque étape de fabrication d'un produit la date de fin prévisionnelle de l'opération en cours.

- La variable *Routing* appartient à l'ensemble **Listoperations**(l'ensemble des listes opérations machines d'un atelier de production flexible). cette variable contient la liste des opérations machines en cours pour un produit. Lorsque cette liste est vide alors la fabrication du produit est alors achevée.
- La variable *Date\_de\_debut* permet de stocker la date de début de l'opération machine en cours ;
- La variable *Date\_de\_fin* permet de stocker la date de fin effective de l'opération en cours ;
- La variable *Retard* appartient à l'ensemble coloré **TIME** et permet de contenir le retard cumulé par un produit au cours de la production.

De cette manière, une place de type **ITEM** contiendra des jetons qui représenteront des produits. Cependant, il faut tout de même relever qu'une place de type **ITEM** sera rarement utilisée pour représenter la réalité dans un système de production flexible. En effet dans un système de production réelle les entités qui contiennent des produits sont appelés des stocks. Le fait de créer l'ensemble **ITEM** permet uniquement de décrire un produit mais on ne saurait se servir de celui-ci pour modéliser un stock. En effet les stocks contiennent certes des produits tout comme l'ensemble **ITEM**, mais ils possèdent d'autres caractéristiques. C'est ce que nous allons définir dans le prochain paragraphe.

## La modélisation d'un stock

Un stock est un jeton de l'ensemble coloré **STOCK\_IO** qui représente la classe objet d'un stock. Cet ensemble est défini de la manière suivante :

```
colset STOCK_IO= product STOCK*STRING;
```

Autrement dit, l'ensemble coloré **STOCK\_IO** est le produit cartésien des ensembles colorés **STOCK** et **STRING**. L'ensemble coloré **STRING** indique la position du stock par rapport au flux de production, en effet si le stock est en entrée du système de production cette variable prendra la valeur "In" et si il est position de sortie la valeur prise sera alors "Out" L'ensemble coloré **STOCK** est une structure de données particulière qui permet de représenter le nombre de produits dans le stock, ou encore qui permet de stocker d'autres types d'informations. Regardons plutôt sa définition :

```
colset STOCK=product ListItem*DonneeSTock;
```

L'ensemble coloré **STOCK** est le produit cartésien des ensembles colorés **ListItem** qui représente la liste des produits dans le stock et de l'ensemble coloré **DonneeSTock**, qui permet la collecte des informations relatives à la capacité du stock et de sa position dans l'atelier. Il est défini de la manière suivante :

```
colset DonneeSTock=product STRING*RxR*INT*INT;
```

L'ensemble coloré **DonneeSTock** est le produit cartésien de quatre ensembles à savoir : **STRING**, **RxR**, **INT** et **INT**. Cela signifie qu'un élément de ce multi-ensemble coloré contient quatre types de données :

- une chaîne de caractères qui représente la machine à laquelle le stock est rattaché ;
- un couple de réels qui représente la position du stock dans l'atelier ;
- deux entiers naturels qui représentent respectivement la capacité maximale et la capacité prévisionnelle d'un stock ;

```
colset TOOL=record TypeTool:STRING*TimeOfChange:INT;  
colset Operation=product STRING*RxR*INT*TOOL*BOOL;  
colset ListOperations= list Operation ;
```

## La modélisation des ressources de transport

Nous avons considéré le matériel de manutention comme une ressource ; par conséquent nous avons représenté le matériel de transport par un jeton coloré de type *Transport*. Cela signifie qu'un matériel de manutention est un élément de l'ensemble coloré *Transport* défini comme suit :

```
colset Transport= record idv:INT*velocity:REAL* position:RxR*  
distance:REAL* ArriveAt: INT* LoadTime:INT* UnloadTime:INT  
timed ;
```

## La modélisation d'une machine flexible

Reprenons la classe machine que nous avons définie précédemment et qui est présenté en figure 2.13. Nous nous servons de ce modèle objet pour définir le modèle de machine basé sur les Réseaux de Pétri. Nous avons justifié au début de notre travail l'utilisation des Réseaux de Pétri Colorés Hiérarchiques.



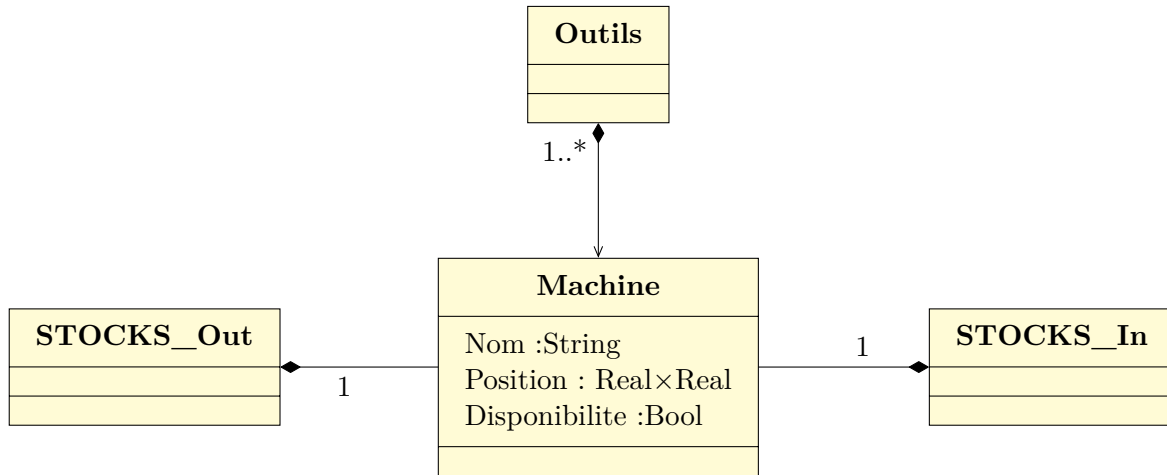


FIGURE 2.13 – classe machine

Comme nous l’avons évoqué en 2.3, le modèle de la machine sera représenté par un Réseau de Pétri modulaire. Ainsi, à la figure 2.14 il est présenté une vue hiérarchique d’une machine flexible.

Le modèle de la figure 2.14 présente quatre places qui sont :

1. **DATA\_INIT** : elle a pour rôle d’initialiser (d’instancier) la classe machine, c’est une place qui contient les informations sur la position de la machine, sur le nombre d’outils qu’elle possède etc. L’ensemble coloré qui a permis de définir cette place est **InitMachine** défini de la manière suivante :

```

colset InitMachine=record
Name:STRING*
Position:RxR* Outils:ListTool*
CapaciteStock:NxN;
  
```

Dans cet ensemble coloré, les variables *Name*, *Position* et *CapaciteStock* correspondent respectivement au nom de la machine, à la liste des outils de cette dernière, et à la capacité maximale des stocks d’entrée et de sortie de la machine (Nous avons supposé que les stocks d’entrée et de sortie ont la même capacité).

2. **Stck\_I** :Elle représente le stock en entrée de la machine ;
3. **Stck\_S** : Elle représente le stock de sortie de la machine ;
4. La place **DTB** nous permet de stocker au cours de la simulation des données sur chaque machine.

La transition **M** est dite de substitution. C’est à l’intérieur de celle-ci que se produisent les interactions entre la machine et les produits.

Pour prendre connaissance des différentes fonctionnalités et interactions qui sont présentes dans la transition **M**, nous allons nous servir du modèle détaillé de la machine qui

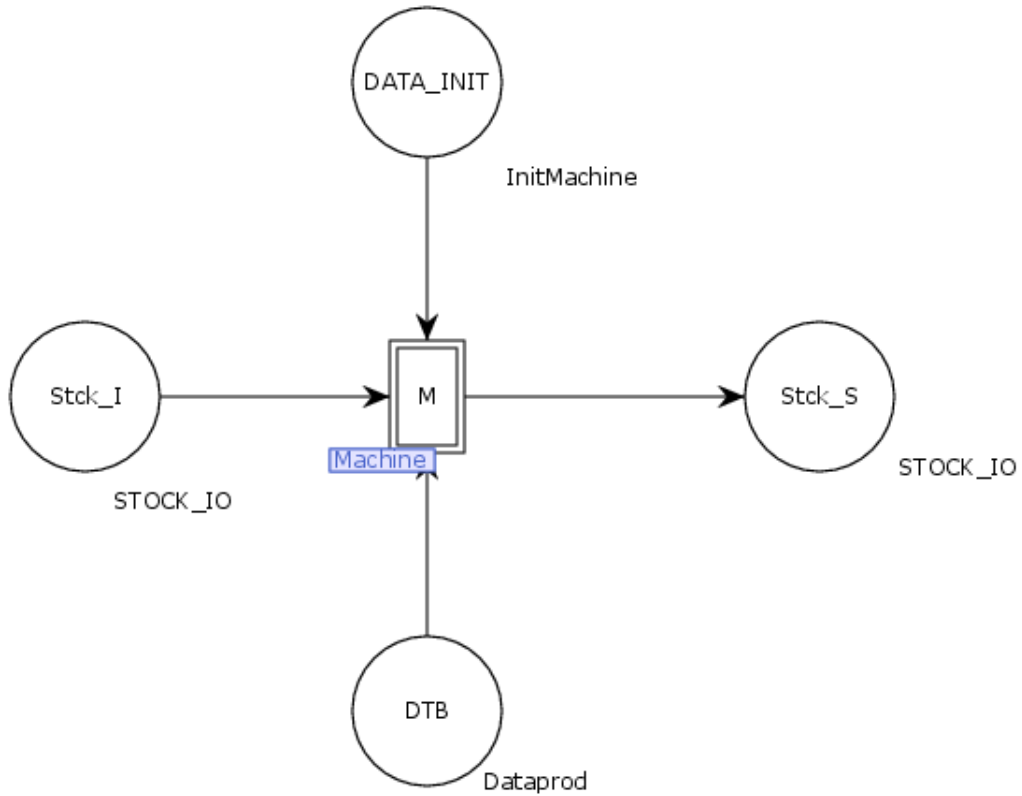


FIGURE 2.14 – Représentation d’un RdP modulaire d’une machine.

est présenté en figure 2.15. Dans le modèle de machine, représenté dans la Figure 2.15 on distinguera les places d’entrée/sortie du Réseau de Pétri Modulaire de la machine et les places internes à celui-ci. Les places d’entrée du RdP modulaire :

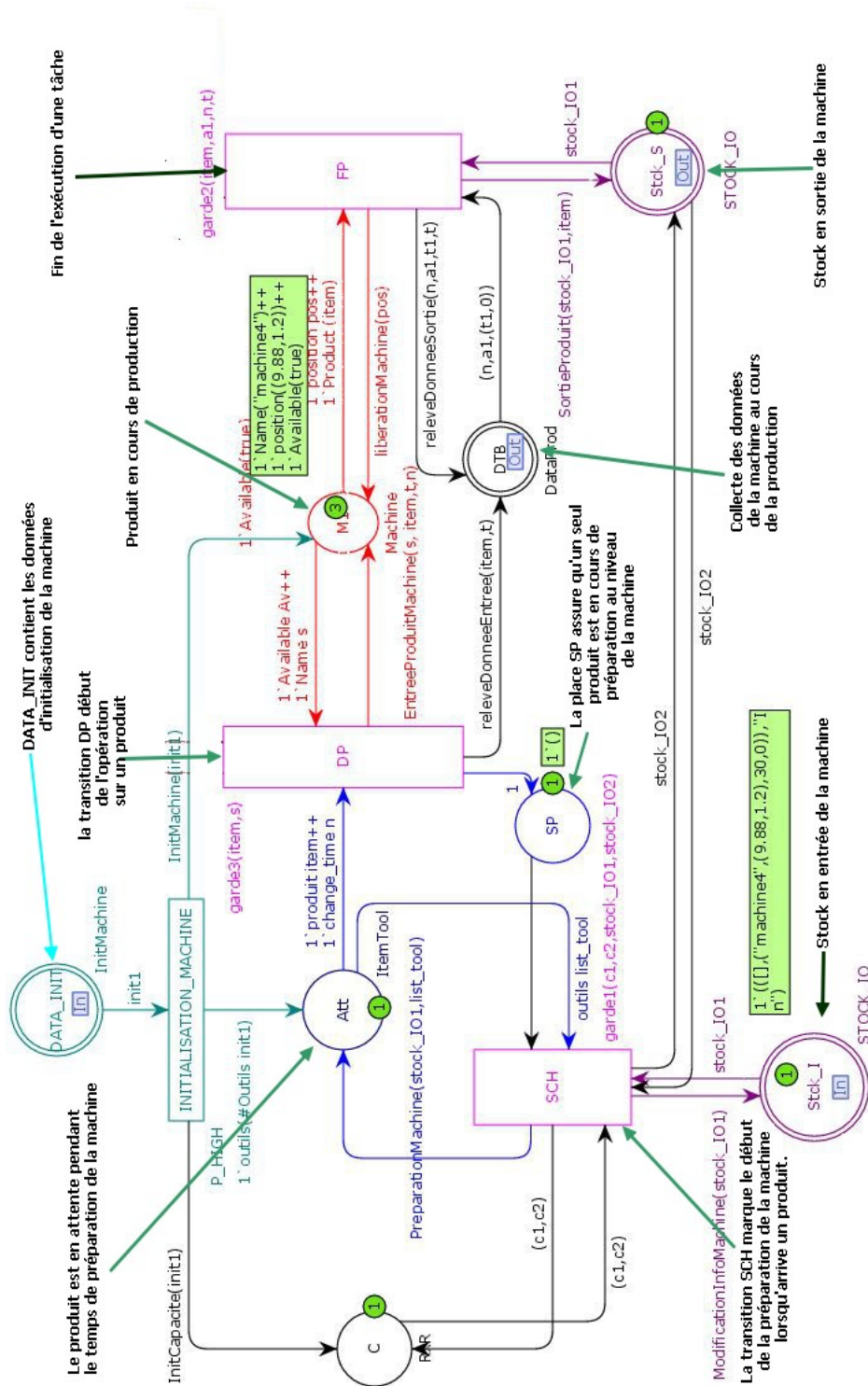


FIGURE 2.15 – Réseau de Pétri Coloré Modulaire d'une machine

### Description des places du modèle de machine

On retrouve sur cette figure les mêmes places **DATA\_INIT**, **DTB**, **Stck\_I** et **Stck\_S** qui sont déjà présentes sur la figure 2.14 et dont nous connaissons déjà les

rôles. Nous décrirons uniquement les places et transitions qui ne l'étaient pas au niveau du modèle hiérarchique (parce que pas visibles) de la figure 2.14. Ainsi nous avons :

1. **C** : La place **C** est de type **RxR**, elle contient un couple de réels (a,b) qui représente la capacité maximale des stocks d'entrée et de sortie de la machine. La première coordonnées correspond à la capacité maximale du stock d'entrée tandis que la deuxième correspond à celle du stock de sortie.
2. **Att** Cette place de type **ItemTool** contient des informations permettant la préparation d'un produit avant qu'il ne rentre dans la machine. Cette place contient la liste des outils de la machine, ainsi que le temps de changement d'outillage. en effet la place **ItemTool** peut contenir à la fois un produit, une liste d'outils (celle que possède la machine) ainsi que le temps de changement d'outillage de l'outil courant. Le produit situé en tête de liste est le produit courant c'est à dire celui qui est utilisé actuellement par la machine.
3. **SP** : Cette place contient un jeton de type **UNIT**. L'absence de jeton dans cette place signifie qu'un produit est dans la zone de préparation de la machine. La place **SP** est toujours initialisé avec un jeton.
4. **M1** : **M1** permet de représenter le fait qu'un produit est en train de subir une opération au sein d'une machine. C'est une place qui se définit à partir de l'ensemble coloré **Machine** suivant :

```
colset Machine =
union Name:STRING+
position:RxR+
Available:BOOL+
Product:ITEM;
```

La définition du multi-ensemble **Machine** permet de comprendre que la place **M1** de type de **Machine** peut contenir la position, le nom et la disponibilité de la machine ainsi que le produit qui est en cours de transformation.

## Les transitions du RdP Modulaire

Pour chaque transition du réseau modulaire nous présenterons, les conditions de franchissabilité, les arcs entrants et les arcs sortant de celle-ci.

### 2.3.6.2.1 La transition SCH

Elle marque le début de la préparation de la machine lorsqu'un produit y arrive. La condition de franchissabilité de **SCH** est la présence de jetons dans les places **C**, **Stck\_I**, **Att**, **SP**, et la vérification de la fonction de garde *garde1*

La fonction de garde *garde1* assure que la transition **SCH** ne sera franchissable que si la taille du stock en entrée est non vide et inférieure à sa valeur maximale, si la capacité

du stock en sortie de la machine est inférieure à sa capacité maximale. Le code source de la fonction *garde1* est donné à dans l'Annexe 3.8.1

### 2.3.6.2.1.1 Les fonctions liées aux arcs de la transition SCH

- $Fe(C \rightarrow SCH) = (c1, c2)$  ; La place contient les capacités maximales des stocks d'entrée et sortie de la machine. Le couple  $(c1, c2)$  lie la valeur de cette place à la transition **SCH**. Cette variable permet une évaluation de la capacité de stockage par rapport à leur valeur maximale et ce à travers la fonction de garde *garde1*
- $Fe(Stck\_I \rightarrow SCH) = stock\_IO1$  : La variable *stock\_IO1* est de type **STOCK\_IO** et contient les données contenues dans la place *Stck\_I*.
- $Fe(Stck\_S \rightarrow SCH) = stock\_IO2$  : La variable *stock\_IO2* est de type **STOCK\_IO** et contient les données contenues dans la place *Stck\_2*.
- $Fe(Att \rightarrow SCH) = outils\ list\_tool$  : La variable *list\_tool* est de type **ListTOOL**, elle permet de lier la liste des outils de la machine à la transition **SCH** afin de préparer la machine lors de l'arrivée d'un produit. La place **Att** est une place de type union c'est ce qui explique le fait qu'on a rajouté l'index *outils* qui permet d'accéder aux éléments de type **ListTOOL** de l'ensemble coloré **ItemTOOL**.
- $Fe(SP \rightarrow SCH) = 1$

### 2.3.6.2.1.2 Les arcs sortant à la transition SCH

- $Fs(SCH \rightarrow C) = (c1, c2)$  : cet arc permet la rétroaction entre la transition **SCH** et la place **C** afin de conserver l'information sur la capacité des stocks une fois que la transition sera franchie ;
- $Fs(SCH \rightarrow Att) = PreparationMachine(stock\_IO1, list\_tool)$  Cette fonction d'arc renvoie dans la place **Att** le produit en tête de la file d'attente du stock d'entrée, ainsi que le temps de changement d'outillage nécessaire dans le cas où l'outil courant de la machine ne serait pas celui qu'il faille utiliser pour assurer l'opération sur le produit issu du stock d'entrée. La fonction *PreparationMachine* est définie en Annexe 3.8.3 La fonction *PreparationMachine* prend ainsi en entrée le stock d'entrée et la liste des outils de la machine et renvoie le produit en tête de liste, la liste des outils (afin de conserver l'information) en mettant l'outil courant en tête de liste et aussi le temps de changement d'outillage afin que celui-ci soit comptabilisé au moment du franchissement de la transition **DP** ;
- $Fs(SCH \rightarrow Stck\_I) = ModificationInfoMachine(stock\_IO1)$  cette fonction permet de mettre à jour les informations du stock d'entrée notamment en modifiant le nombre de produits restant dans la file d'attente. La fonction *ModificationInfoMachine* est définie en Annexe 3.8.9. La fonction *ModificationInfoMachine* prend en entrée un stock et permet de mettre à jour la liste des produits grâce à la fonction *ReturnStock* et aussi la capacité prévisionnelle grâce à la fonction *Stock\_prev\_change*. Ces deux fonctions sont définies en Annexes 3.8.1 et 3.8.2.

- $F_s(SCH \rightarrow Stck\_S) = stock\_IO2$  cette fonction permet de restituer à la place **Stck\_S** le jeton qui lui a été emprunté lors du franchissement de la transition **SCH**;

### 2.3.6.2.2 La transition DP

Elle marque le début d'une opération effectuée par la machine. La condition de franchissabilité de **DP** est la présence d'au moins un jeton dans les places **Att**, **TIME** et **M1**. La fonction de garde *garde3* complète les conditions de franchissabilité de la transition **DP** elle est définie de la manière suivante :

```
fun garde3 (item : ITEM, nomMachine : STRING) =
  [ CheckMachine (item , nomMachine) = true ]
```

La fonction *CheckMachine* est une fonction qui permet de vérifier qu'une machine est bien capable de transformer un produit. Elle prend en entrée un produit, le nom de la machine et vérifie que la prochaine opération qui doit être effectuée sur le produit a été affecté à la machine dont le nom est passé en entrée de la fonction. Elle est définie de la manière suivante :

```
fun CheckMachine (item : ITEM, s : STRING) : BOOL =
  let
    val a = (#Routing item)
    val b = hd a
    val c = (#1 b)
    val d = s
  in
    if (d = c)
    then true
    else false
  end;
```

Les arcs entrants à la transition **DP** sont :

- $f_e(Att \rightarrow DP) = 1'produititem + +1'change\_itemn$  cette fonction permet de lier le produit qui est dans la zone de préparation ainsi que le temps de changement d'outillage à la transition **DP**
- $f_e(M1 \rightarrow DP) = 1'Available(Av) ++ 1'Name(s)$  : cette fonction lie le nom et la disponibilité de la machine.

### Les arcs sortants de la transition **DP** sont :

- $F_s(DP \rightarrow M1) = \text{EntreeProduitMachine}(s, \text{item}, t, n)$  : traduit l'entrée d'un produit dans la machine avec la production d'un jeton de type **Machine** dans la place **M1**. La fonction *EntreeProduitMachine* est définie à l'Annexe 3.8.6 La fonction *EntreeProduitMachine* prend en entrée le produit qui était dans la place **Att**, le nom de la machine, le temps et l'identifiant du produit et renvoie dans la place **M1** le produit en modifiant son temps de fin d'opération en fonction de son instant d'entrée. De plus, la fonction *EntreeProduitMachine* renvoie aussi le nom de la machine contenu dans la variable *s*.
- $F_s(DP \rightarrow SP) = 1$  lorsqu'une opération débute c'est-à-dire lorsque la transition **DP** est franchie, un jeton est renvoyé dans la place **SP** ce qui permet de libérer la zone de préparation d'un produit.

#### 2.3.6.2.3 La transition **FP**

La transition **FP** marque la fin d'une opération effectuée par la machine sur un produit. Cette transition est franchissable lorsqu'il y a des jetons dans les places **DTB**, **M1**, et **Stck\_S** et lorsque la fonction de garde *garde2* est vérifiée.

La fonction de garde *garde2* impose que la transition **FP** soit franchie si les trois conditions suivantes sont vérifiées : la valeur de la variable **t** (temps) de l'horloge correspond à celle de la fin de tâche (vérification de la date de fin de tâche), si l'identifiant du produit est contenu dans la place **DTB**, si l'état du produit en sortie correspond à un état contenu dans la place **DTB**. Les deux dernières conditions permettent de remplir correctement la place **DTB** en vérifiant que l'identifiant du produit sortant avec celui qui est rentré lorsque la transition **DP** a été franchie. La fonction de garde *garde2* est définie comme suit :

```
fun garde2(item:ITEM, a1:A, n:INT, t:TIME)=
[t=(#due item), (#id item)=n, (#state item)=a1]
```

### Les fonctions des arcs entrants à **FP**

- $F_e(M1 \rightarrow FP) = 1'$  : position pos ++ 1'Product item : elle permet de lier la position de la machine et le produit en cours de fabrication à la transition **FP** ;
- $F_e(TIME \rightarrow FP) = t$  : elle lie l'instant *t* à la transition **FP**
- $F_e(DTB \rightarrow FP) = (n, a1, (t1, 0))$  : la variable *n* est l'identifiant du produit en cours de production dans la machine. Cet identifiant a été relevé lorsque le produit est entrée dans la machine (avec le franchissement de la transition **DP**) par la fonction d'arc  $FA(DP \rightarrow DTB)$ . *a1* représente l'état du produit en cours de processus, ceci a été relevé de la même manière que la variable *n* ; et il en est de même pour la

variable  $t1$  qui représente la date d'entrée du produit en cours de production dans la machine. Le couple  $(t1,0)$  doit représenté la date d'entrée et de sortie du produit de la machine. Cependant tant que le produit n'est pas encore sortie de la machine nous avons initialisé les coordonnées de ce couple à 0.

- $Fe(Stck\_S \rightarrow FP)=stock\_IO1$  : cette fonction permet de lier les informations du stock en sortie de la machine à celui de la transition **FP**. Cela afin de pouvoir modifier le stock de sortie (place **Stck\_S**) lorsque le produit sortira de la machine.

### Les fonctions des arcs sortants à la transition **FP**

- $Fs(FP \rightarrow M1)=liberationMAchine(pos)$  : cette fonction permet d'envoyer une information traduisant la disponibilité de la machine dans la place **M1**.

```

fun liberationMachine ( pos : RxR) =
  1` Available true++
  1` position pos

```

- $Fs(FP \rightarrow Stok\_S)SortieProduit(stock\_IO1, item)$  : cette fonction permet de rajouter le produit  $item$  dans le stock  $stock\_IO1$ . La fonction  $SortieProduit$  est définie de la manière suivante :

```

fun SortieProduit ( stock_IO1 : STOCK_IO, item : ITEM) =
  ( Ajouter_produit_stock ( item , (#1 stock_IO1) ) , "Out" )

```



La fonction *Ajouter\_produit\_stock* est définie de la manière suivante :

```

fun Ajouter_produit_stock (i :ITEM, s :STOCK) :STOCK=
let
  val Nom_stock= #1 (#2 s)
  val position_stock= #2 (#2 s)
  val capacite_stock =(#3 (#2 s))
  val produit_en_stock= #1 s
in
  ([i]^~^produit_en_stock, (Nom_stock, position_stock,
    capacite_stock))
end

```

- $Fs(FP \rightarrow DTB) = releveDonneeSortie(n, a1, t1, t)$  : cette fonction permet de relever la date de sortie du produit *item* dont l'identifiant ici est représenté par la variable *n*. *t1* est la date d'entrée du produit dans la machine et *t* celle de sa sortie.

```

  releveDonneeSortie (n:INT, a1:A, t1:TIME, t:TIME)=(n, a1
    ,( t1 , t))

```

#### 2.3.6.2.4 La transition INITIALISATION\_MACHINE

Cette transition n'a pas de rôle dans le fonctionnement du modèle de la machine, elle sert à initialiser les caractéristiques de chaque machine au début de simulation. Elle est active au début de la simulation lorsqu'il y a des jetons dans la place **DATA\_INIT** qui lui envoie son seul arc entrant :

- $Fe(DATA\_INIT \rightarrow INITIALISATION\_MACHINE) = init1$  : la variable *init1* est une variable de type **InitMachine**. Elle permet de lier les données d'initialisation de la machine à la transition **INITIALISATION\_MACHINE**.

Trois arcs sortent de la transition **INITIALISATION\_MACHINE** pour la relier aux places **Att**, **C** et **M1**, les fonctions respectives de ces arcs sont :

- $Fs(INITIALISATION\_MACHINE \rightarrow Att) = 1'outils(\#Outils\ init1)$  : cette fonction permet d'initialiser la liste d'outils de la machine avant le début de la simulation. la variable *init1* est de type **InitMachine**
- $Fs(INITIALISATION\_MACHINE \rightarrow C) = IntCapacite(init1)$  : cette fonction permet d'initialiser la place **C** qui contient la capacité des stocks d'entrée et de sortie de la machine.
- $Fs(INITIALISATION\_MACHINE \rightarrow M1) = InitMachine(init1)$  : cette fonction permet d'initialiser la place **M1** qui contient le nom de la machine ainsi la variable indiquant la disponibilité de la machine.

Les fonctions *IntCapacite* et *InitMachine* sont définies de la manière suivante :

```

fun InitMachine (init1 : InitMachine) =
  1`Name (#Name init1) ++
  1`position (#Position init1)
fun IntCapacite (init1 : InitMachine) =
  (Real.fromInt(#1 (#CapaciteStock init1)), (Real.fromInt(#2 (#
    CapaciteStock init1))))

```

Nous venons de présenter les entités physiques qui constituent un système de production. Nous modéliserons par la suite les activités de transport dans un atelier de production flexible.

## Modélisation du transport à l'aide des réseaux de Pétri Colorés hiérarchiques

Nous avons présenté jusqu'ici des modèles structurels comme ceux des machines, des produits, des stocks, des ressources de transport ou encore des stocks. Il est aussi important de présenter comment les matières et les produits circulent à l'intérieur de l'atelier de production, c'est pour cela que nous présentons dans cette section une modélisation de l'activité de transport dans un atelier de production. L'activité de transport constitue l'ensemble des opérations de transport réalisées au cours de la production. Du point de vue de la ressource de transport une opération de transport consiste à déplacer un produit d'un stock de sortie de machine (zone de chargement) vers un stock soit en entrée de machine, soit en sortie d'atelier (zones de déchargement). On peut décomposer une opération de transport en deux sous-opérations :

- Le choix de (des) ressources de transport qui vont assurer le déplacement du (des) produit (s), le choix du (des) produit(s) qui seront transportés d'une zone de chargement vers une zone de déchargement, le choix des zone de déchargement dans le cas où des dernières ne sont pas clairement définies.
- Le transport proprement dit de la machine de départ vers celle de destination.

Ainsi, nous avons divisé à l'aide des RPCH, notre modèle en deux RPCM (Réseaux de Pétri colorés modulaire). Un Réseau de Pétri Coloré Modulaire permettant l'évaluation de l'activité de transport et la sélection d'une activité de transport et un RPCM modélisant le déplacement des produits proprement dit.

### Modélisation du processus de sélection des produits

Soit un atelier de production de  $m$  machines,  $p$  stock de sortie et  $q$  stock d'entrée de l'atelier de production. La première activité à savoir celle du choix des machines de départ

et d'arrivée sera modélisée par un RPCM qui sera constitué d'une seule transition  $t$  et d'un nombre de places  $P$  défini par :

$$P = 2m + p + q + 3 \quad (2.1)$$

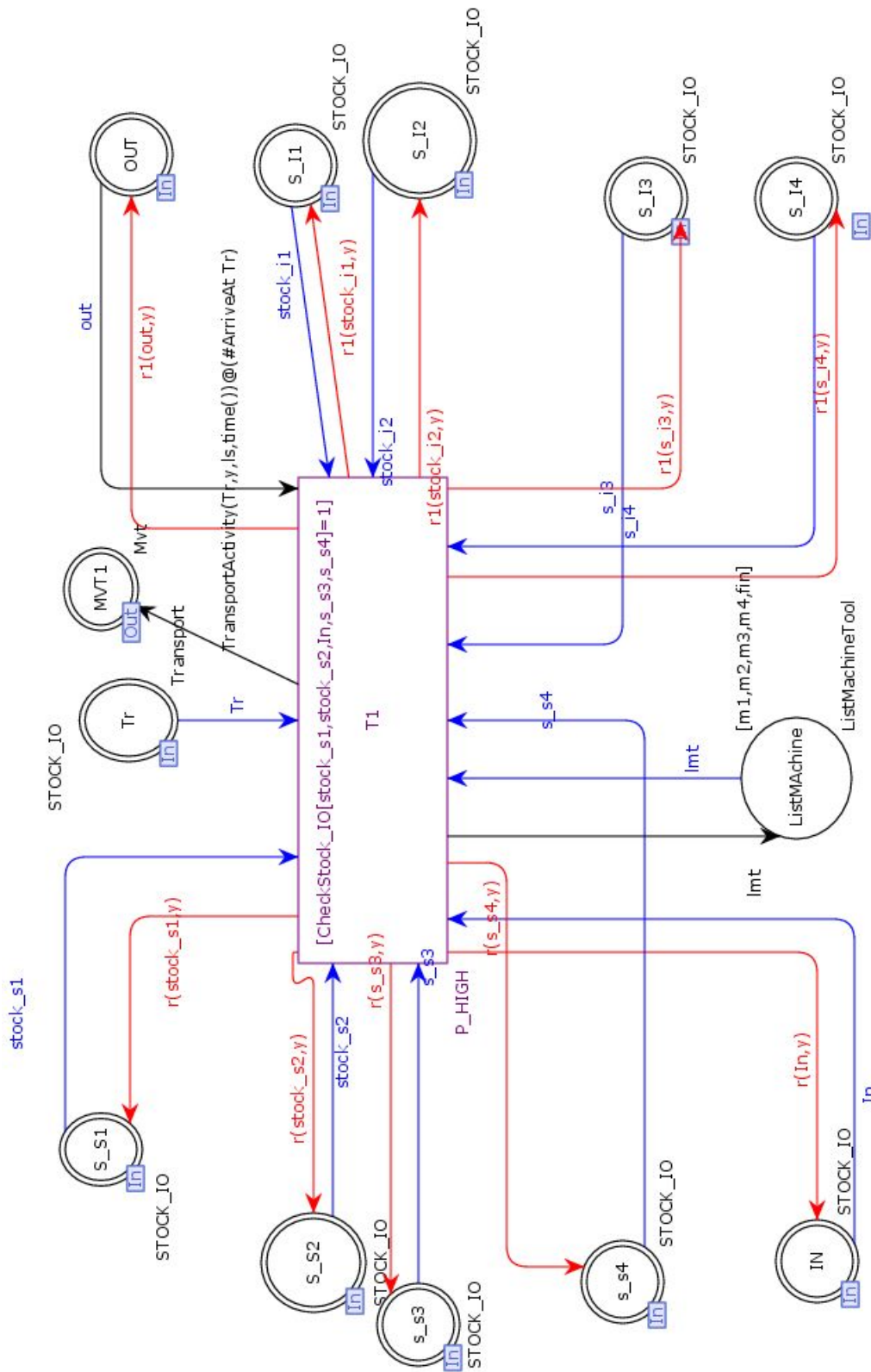
Il faut donc rajouter au nombre de machines et de stocks d'entrée et de sortie de l'atelier ;,

- une place qui contient des informations sur les machines de l'atelier de production.
- Une place contenant l'opération de transport qui sera sélectionnée à l'issu du processus de décision.
- Une place qui représente les ressources de transport disponibles de l'atelier.

Chacune de ces places sera reliée à la transition  $t$  grâce à des arcs en formation une boucle de rétroaction.

Parmi les  $P$  places, il en existe  $2m$  qui représentent les stocks d'entrée de sorties des machines de l'atelier de production. Soit  $SE_i$  la place représentant le stock d'entrée de la machine  $i$  et  $SS_i$  celle représentant le stock de sortie de la machine  $i$ . Ainsi soit  $SE$  l'ensemble des places d'entrée des machines alors  $SE = \{SE_1, SE_2, \dots, SE_m\}$ . De même  $SS = \{SS_1, SS_2, \dots, SS_m\}$  sera l'ensemble des stocks en sortie des machines.

Le RPCM est comprend  $2m + p + q + 2$  places d'entrée et d'une seule de sortie. La place de sortie contiendra le résultat de la décision du transport. La figure 2.16 illustre le RPCM d'un atelier constitué de quatre machines, d'un stock d'entrée et d'un stock de sortie de l'atelier.



Table\_de\_priorites(ls1,Tr)

FIGURE 2.16 – Réseau de pétri Coloré modulaire de l'activité de choix de la machine d'arrivée et de départ par la ressource de transport dans un atelier contenant trois machines, un stock d'entrée et un stock de sortie des produits

Comme on peut le voir sur la figure 2.16, le RPCM comprend treize places. Pour déterminer le nombre de places il suffit de remplacer le triplet  $(m = 4, p = 1, q = 1)$  dans la relation 2.1.

La figure 2.16 donne un exemple de la modélisation de l'évaluation d'une activité de transport. Les places **SS\_1**, **SS\_2**, **SS\_3**, **SS\_4** représentent les stocks en sortie des machines. Les stocks de sortie des machines constituent les zones de chargement. Les places **SE\_1**, **SE\_2**, **SE\_3**, **SE\_4** représentent les stocks en entrée des machines. Les stocks d'entrée des machines constituent des zones de déchargement. Les places **IN** et **OUT** représentent respectivement les stocks en entrée et en sortie de l'atelier. Les stock en entrée de l'atelier de production constituent des zones de chargement et les zones en sortie de ce dernier des zones de déchargement. La place **Tr** représentent les transporteurs qui sont disponibles. La place **Mvt** représente l'opération de transport qui a été sélectionnée. Cette place contient donc le produit qui doit être transporté, ainsi que le transporteur qui assure le déplacement. La transition **T1** est reliée à toutes les places que nous venons de citer.

### Conditions de franchissabilité de la transition T1

La transition **T1** permet l'évaluation et la sélection d'une opération de transport cela signifie que le tirage de cette dernière doit entraîner le déplacement d'un produit d'une des zones de chargement (les places **SS\_1**, **SS\_2**, **SS\_3**, **SS\_4** et **IN**) vers une zone de déchargement (les places **SE\_1**, **SE\_2**, **SE\_3**, **SE\_4** et **OUT**). Ce déplacement doit se faire à l'aide d'une ressource de transport (place **Tr**). L'une des conditions pour qu'il y ait une évaluation de l'activité de transport est qu'il ait au moins un produit à transporter c'est à dire au moins un produit dans les stocks de sortie des machines ou en entrée de l'atelier. Cependant le modèle que nous avons présenté porte une particularité toutes les places sont reliées à la même transition **T1**. Cela signifie que **T1** ne sera franchissable que si les places **STOCK\_IO** contiennent au moins un jeton. Nous avons expliqué au chapitre 3 que la place **STOCK\_IO** est une place qui ne contient qu'un seul jeton. En effet un jeton de type **STOCK\_IO** est un ensemble de données qui comprend la liste des produits contenus dans le stock, la position du stock, ou encore sa capacité (la définition complète du type **STOCK\_IO** a été faite au chapitre 2). Ainsi lorsqu'un produit arrive dans un stock c'est la valeur de la liste des produits dans le stock qui est d'une certaine manière mise cela ne change donc pas le nombre de jeton de la place de type **STOCK\_IO** en question. Pour que la transition **T1** soit franchissable nous devons donc initialiser toutes les places de notre modèle. Les places de type de **STOCK\_IO** seront initialisées en prenant soin de mettre une liste de produit vide comme dans l'exemple ci dessous : On

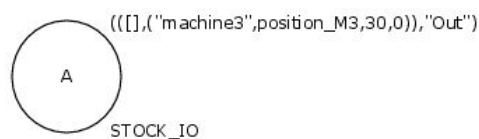


FIGURE 2.17 – Place de type **STOCK\_IO** représentant un stock vide

observe bien que la place A ne contient aucun produit vu que la première coordonnée du

triplet représentant la liste des produits dans le stock est représentée par le symbole  $[ ]$ . Nous pouvons donc interpréter la figure 2.17, comme une place  $A$  de type  $STOCK\_IO$  ne contenant aucun produit (présence d'une liste vide  $[ ]$ ), reliée à la machine "machine3", située à une position  $position\_M3$  dans l'atelier de production, et servant de stock sortie à la machine "machine3"

De cette manière le marquage des places de types de  $STOCK\_IO$  sera initialisé de afin que la liste des produits que contient chaque stock soit vide. Toutefois si la présence d'un jeton de type stock vide est nécessaire pour que la transition  $T1$  soit franchissable dans la réalité elle ne devrait être suffisante. En effet la franchissabilité de la transition  $T1$  signifie que les conditions (ressources disponibles, et présence de(s) produit(s) dans les stocks de sortie des machines, ou dans le stock d'entrée de l'atelier) sont réunies pour qu'une ou plusieurs opérations de transport soient effectuées. Cela signifie que pour que la transition  $T1$  soit franchissable il faut que :

1. La place  $Tr$  contient au moins un jeton. Cela signifie qu'il ya au moins une ressource de transport disponible ;
2. Les places de type  $STOCK\_IO$  reliées à la transition  $T2$  contiennent un jeton de stock vide ;
3. Les places représentant les stocks de sortie de machines ou de le stock d'entrée de l'atelier contiennent au moins un produit dans la sa liste de produits.

Pour s'assurer de cette dernière condition nous avons rajouté une fonction de garde au niveau de la transition  $T1$ . Cette fonction est située au niveau de la transition  $T1$  sur la figure 2.16. La fonction garde est

$CheckStock\_IO [stock\_s1 , stock\_s2 , In , s\_s3 , s\_s4]=1$

La fonction  $CheckStock\_IO$  prend en entrée la liste des stocks en sortie et en entrée de l'atelier et qui renvoie 1 si au moins un produit est contenu dans un stock contient un produit. Le code source de la fonction de garde  $CheckStock\_IO$  est donnée en Annexe 3.8.5.

La fonction  $CheckStocks$  prend en entrée la liste des stocks de sortie des machines et les stocks d'entrée de l'atelier et qui renvoie 0 si ces stocks sont vides et 1 dans le cas contraire. La franchissabilité de la transition  $T1$  entraîne la génération d'une table d'évaluation du transport que nous avons définie dans le paragraphe 3.3.5. L'élaboration de la table se fait par l'utilisation des fonction d'entrée sortie liées au transition dans un RdPC.

Lorsque un produit arrive dans une des places représentant les zones de chargement (stock en sortie des machines, stock en entrée de l'atelier) la transition devient franchissable et le produit passe de la zone de chargement à la zone de déchargement en attente d'une expédition ou d'une opération sur une machine. A titre d'exemple si l'on observe le réseau de la figure 2.16 l'activité de transport se déclenche lorsque qu'il y a au moins un jeton dans l'une des places  $SS\_i, i \in \{1, 2, 3, 4\}$  ou  $E$  et le franchissement de la transition  $T1$  entraîne le passage d'un jeton de la place  $SS\_i, i \in \{1, 2, 3, 4\}$  à une place

$SE\_j, j \in \{1, 2, 3, 4\}$

### Définition des fonctions des arcs

Dans ce paragraphe nous allons présenter les fonctions et variables liées aux arcs de la figure 2.18.

Nous utilisons trois fonctions afin de modéliser l'évaluation et la sélection des opérations de transport. Sur la figure 2.18, Nous pouvons voir qu'il existe trois fonctions liées aux arcs à savoir : les  $r$ ,  $r1$  et `TransportActivity`. La fonction  $r$  permet de mettre à jour les stocks de sortie des machines à l'issue de la sélection d'une opération de transport. La sélection d'une opération de transport implique aussi celle d'un stock de sortie de machine, d'un stock d'entrée et d'une ressource de transport, car une opération de transport nécessite un stock de départ, un stock d'entrée et une ressource de transport. La fonction  $r$  prend en entrée deux stocks : Le premier est celui en sortie d'une machine et le second stock est celui qui est sélectionné à travers un algorithme de sélection des opérations de transport que nous définirons dans le chapitre 3 . Pour une bonne compréhension supposons que le processus de sélection d'une opération de transport conduisent à retenir  $O$  , une opération de transport qui se fera grace à une ressource  $x$  qui déplacera le produit d'un stock de sortie représenté par une variable  $s$  vers un stock d'entrée  $e$ . soient les variables  $s1,s2,S3,s4$  les variables des stocks de sortie des machines de l'atelier de production. La fonction  $r$  permet une fois de vérifier une fois le processus de sélection achevée que :

- Les stocks de sortie des machines qui n'ont pas été sélectionnés restent inchangés, c'est à dire que le nombre de produits dans le stock ne change pas ;
- Le stock qui a été sélectionné par le processus de sélection verra son nombre de produits diminuer de 1

La fonction  $r1$  quant à elle joue le même rôle pour les stocks d'entrée des machines que la fonction  $r$  pour les stocks de sortie des machines. En effet une fois qu'une opération de transport est sélectionnée un stock d'entrée est aussi sélectionné. La fonction  $r1$  permet de mettre à jour les stocks en entrée des machines. La fonction  $r1$  est définie en Annexe3.9.

La dernière fonction d'arc que nous pouvons observer sur la Figure 2.18 est la fonction `TransportActivity`. Cette fonction est liée à l'arc reliant la transition **T1** à la place **Mvt**. La fonction `TransportActivity` permet de mettre dans la place **Mvt** le produit et la ressource de transport qui ont été sélectionnés après le processus de sélection lors du franchissement de la transition **T1**) pour dire qu'une opération de transport est en cours. Elle permet de mettre à jour les informations concernant le produit et la ressource de transport. `TransportActivity` modifie par exemple la position courante de la ressource de transport, son temps d'arrivée vers la zone de chargement, la date de début de la prochaine opération et affecte la machine destination au produit.

## Modélisation du déplacement des produits

Dans ce paragraphe nous présentons la modélisation de l'acheminement des produits. Le modèle de RdPCH de la Figure 2.18 est celui de l'acheminement des produits d'un atelier à quatre machines et possédant une seule zone de stockage en sortie de l'atelier.

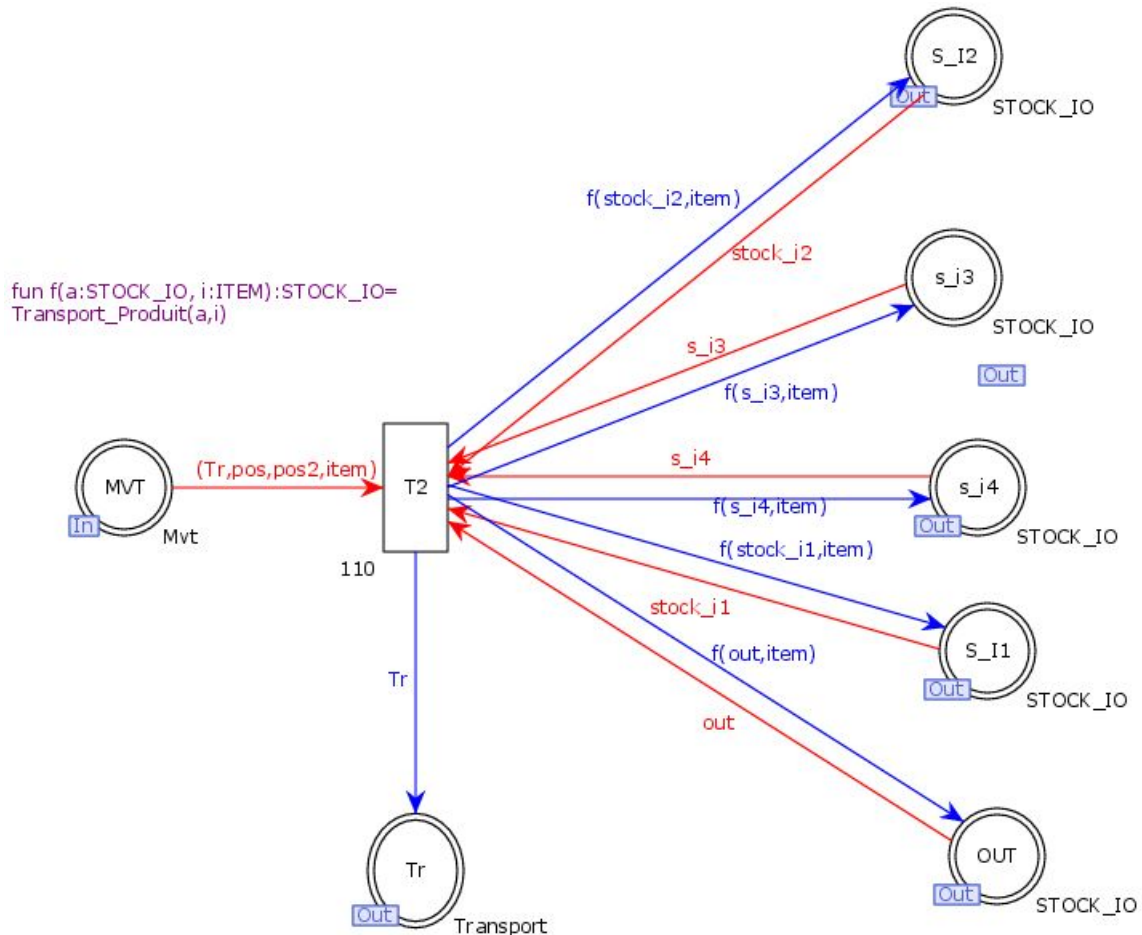


FIGURE 2.18 – Modélisation de l'opération de transport dans un atelier de production à quatre machines possédant chacune un stock d'entrée et un stock de sortie. L'atelier de production lui même possède une seule zone de sortie des produits.

Pour réaliser la modélisation de l'acheminement des produits il faut relier en boucle de rétroaction les places désignant les stocks en entrée et sorties des machines, la place contenant la ressource de transport et le produit à transporter à une même transition. C'est ce que nous pouvons observer sur la Figure 2.18.



## Description des places

La place  $MVT$  représente l'opération de transport en cours, en effet c'est elle qui contient le produit qui est en cours de transport ainsi que la ressource de transport qui assure le déplacement vers l'un des stocks d'entrée des machines (représenté ici par les places  $SI_1, SI_2, SI_3$  et  $SI_4$ ) ou vers la sortie de l'atelier représenté par la place  $OUT$ . La place  $Tr$  est la place qui contient toutes les ressources de transport disponibles de l'atelier de production. La transition marque la fin du transport.

## Description des transitions

La transition  $T2$  marque le début du transport du produit proprement dit. Pour que la transition  $T2$  soit franchie, il faut que il y ait au moins un jeton dans les places  $Mvt$  et que les places  $SI_1, SI_2, SI_3, SI_4$  et  $OUT$  soient non vides. Une fois que ces conditions sont réunies la transition  $T2$  est franchissable et à ce moment là le jeton de type produit contenu dans la place  $Mvt$  se retrouve dans l'une des places avalées  $SI_1, SI_2, SI_3, SI_4$  ou  $OUT$ . Le franchissement de  $T2$  entraîne aussi la disparition du jeton transport de la place  $Mvt$  et l'apparition de ce dernier dans la place  $Tr$  ce qui signifie que l'opération de transport est terminée et que la ressource de transport est de nouveau disponible pour effectuer une nouvelle opération.

## Description des fonctions des arcs

Si l'on observe la figure 2.18, on peut distinguer trois types d'arcs : les arcs sortant de  $T2$  vers les zones de déchargement, les arcs entrant dans  $T2$  et provenant des zones de déchargement, l'arc provenant de la place  $Mvt$ , l'arc sortant de  $T2$  et reliant la place  $Tr$ . A présent, définissons les fonctions liées à ces arcs.

- $Fe(MVT \rightarrow T2) = (Tr, pos, pos2, item)$  cette fonction lie à la transition  $T2$  la ressource de transport, la position de la machine de départ (zone de chargement), la position de la zone déchargement, et le produit qui est transporté.
- $Fe(S_I1 \rightarrow T2) = Fe(S_I2 \rightarrow T2) = Fe(S_I3 \rightarrow T2) = Fe(S_I4 \rightarrow T2) = Fe(OUT \rightarrow T2)$  transmet les informations à la transition  $T2$  concernant les zones de chargement. C'est une fonction identité. Ces informations permettent une évaluation de tous les stocks lors du franchissement de la transition  $T2$
- La fonction de l'arc sortant à la transition  $T2$  et reliant les zones de chargement est la fonction  $f(stock, produit)$  est une fonction qui prend en entrée une zone de déchargement et le produit à transporté, vérifie si la zone de déchargement est bien la zone de destination du produit, dépose le produit si c'est la cas et dans le cas contraire ne le dépose pas.
- La fonction de l'arc ( $T2 \rightarrow Tr$ ) est une fonction qui renvoie la ressource de transport qui était contenue avant le franchissement de  $T2$  dans la place  $MVT$

# Validation des modèles développés

## Définition du problème

Les modèles de simulation des systèmes de production sont utilisés pour plusieurs raisons telles que la conception et l'évaluation des systèmes de production, le développement de nouvelles politiques de gestion de production [76], et aussi pour comprendre les problématiques de liées à ce genre de systèmes[77].

Au moment où les modèles des systèmes physiques passent obligatoirement par une phase de validation [78], la modélisation des systèmes discrets, notamment des systèmes de production est développée sur la base d'observations de fonctionnement des composantes élémentaires du système sans passer par une étape de validation du système dans sa globalité [79]. Ce problème peut être surmonté avec le rajout des étapes de vérification et de validation des modèles au processus de modélisation du système de production. En effet la vérification d'un modèle est défini comme le fait d'assurer le programme informatique et l'algorithme sont implémentés correctement. Tant dis que la validation du modèle est définie comme la justification que le modèle assure un niveau satisfaisant de précision quand il est exécuté dans son domaine d'application comparé au résultat d'un système réel [80].

Le CESI de Rouen dispose d'un atelier de production didactique permettant l'assemblage de maquettes de téléphone. Nous nous sommes basés sur cet atelier pour valider les modèles développés parce qu'il est à notre disposition, ce qui nous a permis de faire plusieurs campagnes de mesures.

## La présentation de l'atelier de production didactique du CESI

La Plateforme de recherche Performance Industrielle et Innovation du CESI est composée d'un atelier d'assemblage à usage pédagogique et de recherche. Sur celui-ci, les étudiants peuvent se former sur la résolution de problèmes d'amélioration de performances de systèmes de productions de nouvelle génération. Les équipes du laboratoire de recherche travaillent aussi sur cet atelier pour tester leurs méthodes et développements tels que l'utilisation de la réalité augmentée, l'ordonnancement dynamique ou la robotique mobile pour le transport des pièces. Le transport sur la ligne de production est assuré par des palettes équipées d'un système RFID circulant sur un convoyeur circulaire. Pour modéliser cet atelier nous allons étudier les caractéristiques des produits, des postes de travail ainsi que des moyens de transport.

### Le produit

Le produit assemblé dans cet atelier de production est une maquette de téléphone portable composée des éléments suivants :

- Une coque constituée de deux couvercles : un couvercle supérieur et un couvercle inférieur.
- Une carte électronique ou circuit imprimé avec des emplacements pour deux fusibles.
- Un ou deux fusibles pourront être installés sur la carte électronique.

Ces éléments rentrant dans le processus d'assemblage téléphones portables sont visibles sur la Figure 2.19.



FIGURE 2.19 – Composition du produit

### Les postes de travail

La chaîne de production comprend six postes de travail répartis sur une ligne de production de forme circulaire comme le montre la Figure 2.20. Ces postes de travail sont :



FIGURE 2.20 – L'atelier d'assemblage flexible du laboratoire IRISE du CESI

- *Le poste ROBOT1* : il est composé d'un bras manipulateur robotisé qui assure la récupération du premier couvercle à partir du stock d'entrée pour le mettre sur le tapis roulant et la récupération du produit fini pour le mettre dans le stock de sortie.

Les couvercles sont stockés dans 16 emplacements rectangulaires. Lorsque le support arrive au niveau du poste de travail, le bras manipulateur prend le couvercle du premier emplacement non vide du stock d'entrée et le dépose sur la palette. Lorsque l'assemblage de la pièce est terminé, le bras manipulateur récupère le produit sur de la palette puis le dépose sur un des 16 emplacements du stock de sortie. Ce robot suit des trajectoires différentes selon la position de la pièce dans le stock. La Figure 2.21 montre un exemple de deux trajectoires que le robot peut prendre ainsi que les positions des différentes composantes de ce poste de travail.

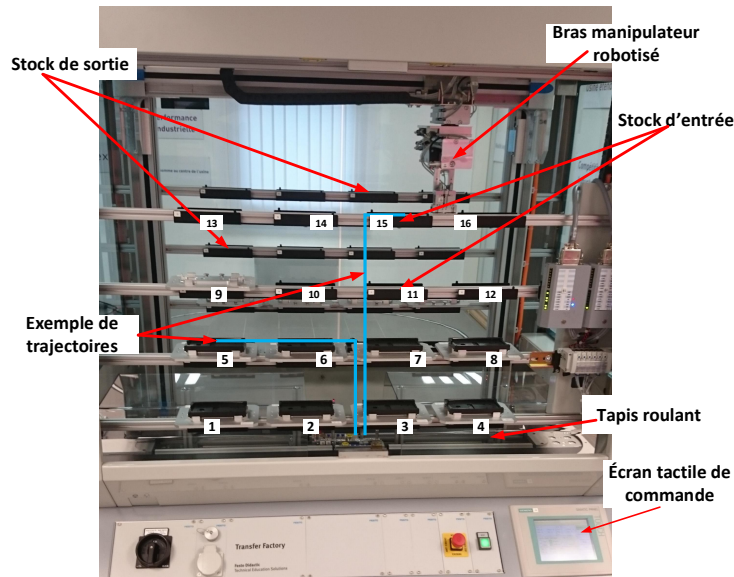


FIGURE 2.21 – Le *ROBOT1* de déchargement et chargement des stocks d'entrée et de sortie respectivement.

- *Le poste ROBOT2* : il est composé d'un bras manipulateur robotisé à six degrés de liberté et 3 types de pinces. Il assure les opérations de fixation des circuits imprimés dans le couvercle inférieur de la coque du téléphone. Le bras articulé permet aussi de fixer les fusibles sur le circuit imprimé une fois que celui-ci est fixé sur le couvercle inférieur de la coque. Le poste de travail *ROBOT2* est équipé d'une zone de stockage d'alimentation des circuits imprimés. Cette zone de stockage est un plateau roulant rectangulaire divisé en 12 positions dans lesquelles sont placées les cartes électroniques qui y sont posés. Ces emplacements permettent au bras de retrouver les circuits électroniques qui y sont posés. Si le premier emplacement est vide le bras va chercher le circuit électronique sur le deuxième emplacement s'il est présent, sinon il va continuer de scanner les emplacements jusqu'à ce qu'il trouve un emplacement contenant une carte électronique. Ce poste est aussi équipé d'un système de vision artificielle permettant de vérifier l'orientation du couvercle inférieur avant de procéder à l'installation de la carte électronique et des fusibles. Ce poste de travail a des temps opératoires variables selon les types de produits, l'orientation du couvercle, la position de la carte électronique dans le stock et la position du fusible utilisé. La Figure 2.22 montre les différentes parties de ce poste de travail ainsi qu'un exemple de trajectoires possibles du robot.

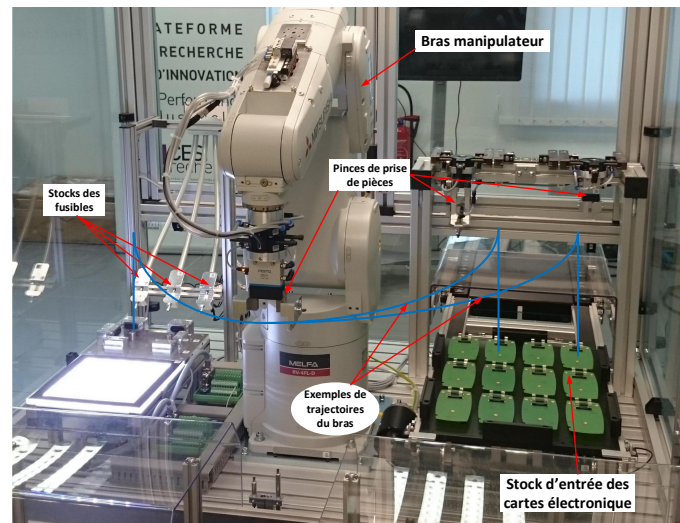


FIGURE 2.22 – Les différents composants du poste de travail *Robot2*.

- Le poste *Depot Couvercle* (Station D) : Le rôle de ce poste de travail consiste au dépôt du couvercle supérieur de la coque du téléphone afin de protéger le circuit imprimé. Les couvercles supérieurs sont stockés dans une colonne rectangulaire les uns au-dessus des autres (cf. Figure 2.23). Lorsque la pièce arrive à ce poste de travail, le couvercle supérieur en bas de la colonne est posé sur la pièce en fabrication. Le niveau du stock est contrôlé par vision en utilisant une application sur un téléphone portable. Le temps d'opération de ce poste de travail est considéré comme constant vu la similitude et la simplicité des opérations effectuées.

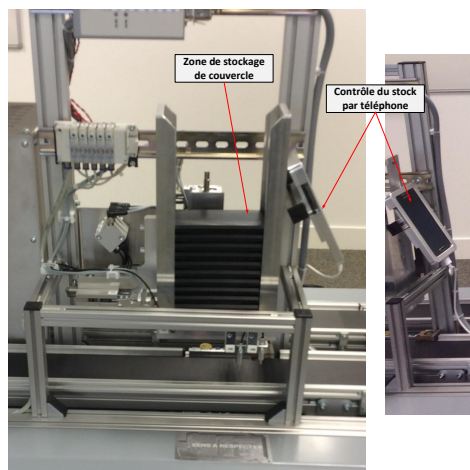


FIGURE 2.23 – Le poste de travail *DéposeCouvercle*

- Le poste *PRESSE* : il a pour rôle d'emboîter le couvercle supérieur sur le reste de la pièce par pression. A l'arrivée de la pièce une plaque métallique est abaissée afin d'appuyer sur le couvercle supérieur pour l'emboîter correctement dans le reste de la pièce (cf. Figure 2.24).
- Le poste *PERCEUSE* : Le rôle de ce poste de travail est de percer le premier



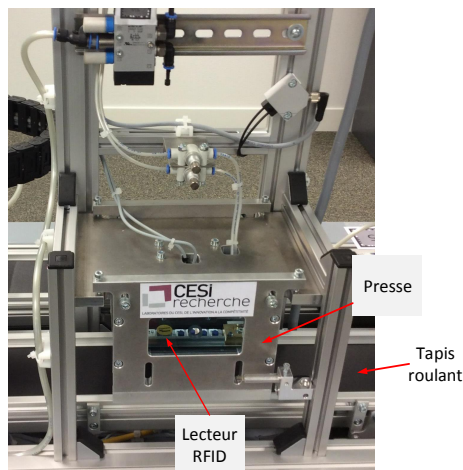


FIGURE 2.24 – Le poste de travail *Presse*

couvercle afin de pouvoir y insérer la carte électronique. Une fois le premier couvercle arrivé au niveau du poste de travail, deux perceuses sont actionnées l’une après l’autre (cf. Figure 2.25). Le temps d’opération de ce poste ne subit pas beaucoup de variation et il est considéré comme constant.

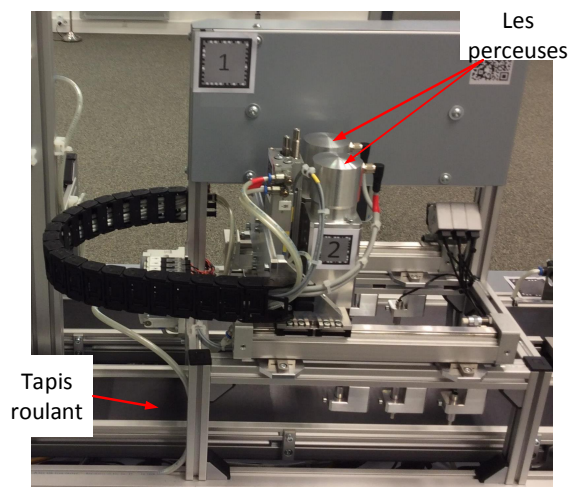
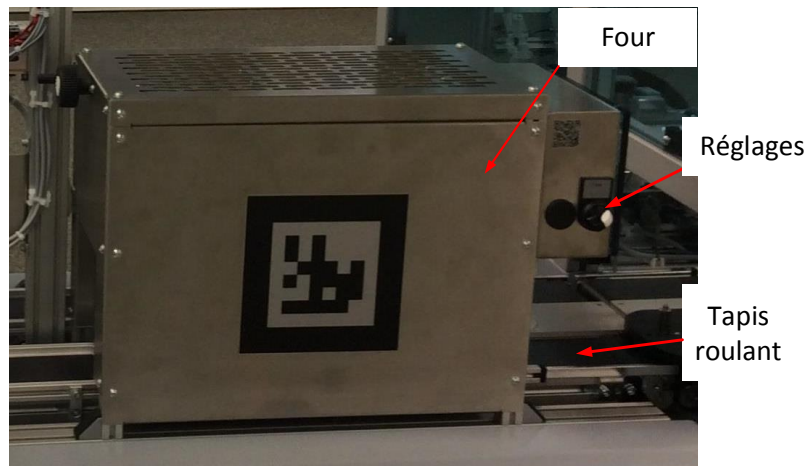


FIGURE 2.25 – Le poste de travail *Perceuse*

- Le poste *FOUR* : Ce poste de travail montré sur le Figure 2.26 est composé d’un four dont on peut régler la température et le temps de chauffe. Grâce à cette élévation de température les deux couvercles seront scellés ensemble. La durée de cette opération ne subit pas beaucoup de variation et est considérée comme constante.

### Les Etapes d’assemblage d’un téléphone portable

L’assemblage d’un téléphone se fait en plusieurs opérations successives qui sont :

FIGURE 2.26 – Le poste de travail *Four*

1. **Opération 1** : Le commencement du processus d'assemblage se fait sur le poste *Robot1*, où il récupère un couvercle inférieur du stock d'entrée et le pose sur une palette au niveau du convoyeur.
2. **Opération 2** : Perçage du couvercle inférieur en utilisant le poste de travail *PERCEUSE*
3. **Opération 3** : Pose des composants électroniques du téléphone, notamment la carte électronique et les fusibles sur le couvercle inférieur en utilisant le poste de travail *Robot2*. Cette opération est composée étapes suivantes :
  - **Étape 1** : Le bras manipulateur récupère le couvercle inférieur de la palette et le place dans la zone de contrôle visuel de l'orientation de la pièce.
  - **Étape 2** : La correction de l'orientation du couvercle s'il en a besoin et son placement dans la position dédiée à l'installation des composants électronique.
  - **Étape 3** : Chargement de l'outil de prise de carte électronique
  - **Étape 4** : Prise et installation de la carte électronique sur le couvercle inférieur.
  - **Étape 5** : Chargement de l'outil de prise de fusible
  - **Étape 6** : Installation des fusibles sur la carte électronique
  - **Étape 7** : Chargement de l'outil de prise du couvercle
  - **Étape 8** : Mise de l'ensemble sur la palette

Le nombre des fusibles dépend du type de produits que l'on souhaite assembler.

4. **Opération 4** : La pose du couvercle supérieur de la coque du téléphone pour refermer le téléphone et protéger la carte électronique. Cette opération est effectuée sur le poste de travail *Depot Couvercle* (Station D)
5. **Opération 5** : Une opération de pressage au-dessus du couvercle supérieur. Cette opération permet d'emboîter les deux parties de la coque du téléphone. Cette opération est effectuée par le poste de travail *PRESSE*

6. **Opération 6** : Une opération de chauffage de l'ensemble pour sceller les deux couvercles. Le poste de travail *FOUR* permet de d'effectuer cette opération avec plusieurs possibilités de durée de chauffe.
7. **Opération 7** : Le bras mécanique (*ROBOT1*)déplace le produit fini à partir de la palette jusqu'à un emplacement vide du stock de sortie.

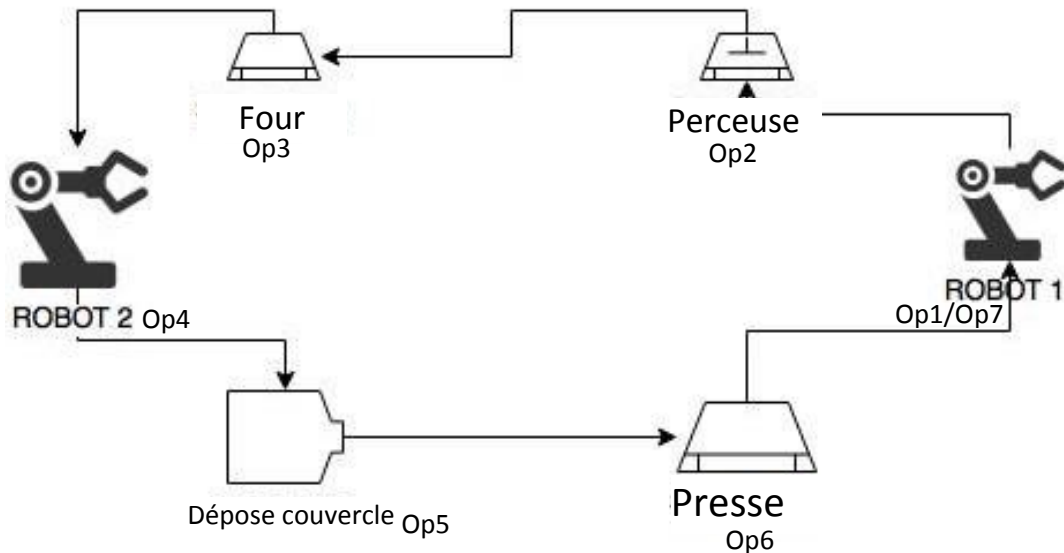


FIGURE 2.27 – Schéma du fonctionnement de l'atelier flexible ,

### Les stocks et le transport

Le fonctionnement de cet atelier décrit précédemment est résumé sur la Figure 2.27. Les flèches représentent le sens de circulation au sein de l'atelier via le tapis roulant, et les numéros associés aux opérations l'ordre dans lequel celles-ci seront effectuées. Ces flèches représentent en réalité les différents tapis roulants ainsi que leur sens, ces six tapis roulants serviront de stocks tampons entre les différentes stations.

La ligne de production du CESI est disposée en forme elliptique , les stations se suivent les unes par rapport aux autres via un tapis roulant sur deux lignes face à face et les deux extrémités forment un U étroit.

#### 2.5.2.4.1 Caractéristiques et contraintes

La ligne d'assemblage est soumise à plusieurs contraintes imposées par les caractéristiques des machines, des moyens de transport et le plan de production. Nous avons identifié les contraintes suivantes : :

- Les contraintes de précédence des opérations d'un produit : les opérations doivent s'exécuter dans un ordre bien défini (c.f. Section 2.5.2.3), aucune inversion de l'ordre des opération n'est possible.



- Les contraintes disjonctives au niveau de chaque machine. Une même ressource (machines de production) ne peut pas travailler sur deux pièces à la fois. Ce type de contraintes ne concerne pas les ressources transport puisque le transporteur ( le tapis roulant) peut déplacer plusieurs pièces en même temps.
- Les temps de processus stochastiques des *Robot1* et *Robot 2*
- La taille des stocks est limitée, elle est proportionnelle à la longueur du tapis roulant reliant deux stations de production
- Le couvercle inférieur ne peut être placé que dans un seul sens, s'il est placé dans le mauvais sens le *Robot1* va le remettre dans le bon sens suite au test réalisé avant la pose du circuit électronique.
- Le couvercle supérieur ne peut être placé que dans un seul sens, s'il est mal placé, la station de travail *DepotCouvercle* ne pourra effectuer correctement son rôle et le produit sera défectueux.
- Les cartes électroniques ne peuvent être placées que dans un seul sens, si elles sont placées dans le mauvais sens elles ne conviendront pas au support.

### La taille des stocks tampons

Afin de modéliser l'intégralité du système, il est indispensable de déterminer la taille des stocks tampons, qui sont présents au nombre de 6. Les stocks d'entrées des postes de travail varient selon la taille des tapis roulants qui les relient. Nous avons identifié les tailles maximales suivantes des stocks selon le poste de travail :

- 13 palettes pour le poste *DepotCouvercleC*
- 5 palettes pour le poste *Presse*
- 6 palettes pour le poste *Perceuse*
- 5 palettes pour le poste *Four*
- 4 palettes pour le poste *Robot1*
- 4 palettes pour le porte *Robot2*

Si un poste de travail n'est pas actif on peut considérer qu'il fait partie du stock d'entrée du poste de travail suivant, la taille de stock d'entrée est donc considérer comme la somme des deux stocks. Pour des raisons de simplification, nous avons choisi de considérer des tailles fixes pour les stocks et considérer qu'un poste de travail inactif possède un temps de fabrication nulle.

## Gammes de production

En fonction des traitements qui sont faits aux produits dans l'atelier, il est possible d'obtenir plusieurs gammes de production. Ce qui peut différencier deux produits sortis de la chaîne de production est :

- La durée du chauffage du four qui peut être réglée entre 0 et 120 secondes.
- Le nombre de fusibles à insérer dans les téléphones portables. On peut en mettre un ou deux.

Dans le cadre de notre expérimentation nous avons défini durée de chauffage du four de 5 secondes et d'insérer deux fusibles dans chaque téléphone portable.

## Détermination de la durée des opérations pour les postes de travail déterministes

La ligne de production est composée de deux types de postes de travail : les postes de travail possédant des durées d'opération constantes (variabilité négligeable) et des postes de travail ayant des temps d'opération variables (variabilité considérable).

Nous avons relevé les durées des opérations au niveau des postes de travail qui ne présentent pas de variabilité importante à savoir les postes : *Perceuse*, *Four*, *DepotCouvercle* et *Presseuse*. La durée des opérations est présentée dans le tableau 2.3. Ces données ont été relevées par le biais capteurs qui sont installés sur la chaîne de production.

Poste de travail	Durée de l'opération
DC	2 secondes
PRESSE	6 secondes
PERCEUSE	7 secondes
FOUR	5 secondes

TABLE 2.1 – Durée des opérations des machines possédant des temps de process constants

Les temps de processus du poste de travail *Robot1* et *Robot2* ne sont pas déterministes. La variabilité de ces temps est due à plusieurs causes que nous allons citer par la suite. Afin de définir des modèles (basés sur les RdP) maniabiles et efficaces, nous avons identifié le modèle stochastique de la variabilité des temps de production et nous les avons utilisés pour déterminer les temps de processus de ces postes de travail

### Détermination de la durée des opérations pour les postes *Robot1* et *Robot2*

Pour commencer, nous avons déterminé les causes de variabilité sur les postes de travail *Robot1* et *Robot2*. Ensuite nous avons fait une série de mesure afin de déterminer la loi de probabilité qui gère la variation des temps de production de ces postes de travail.

#### 2.5.2.8.1 *Robot1* :

Les causes de variabilité de temps de production du *Robot1* que nous avons pu identifier sont les suivantes

- *Position du produit récupéré* : le temps d'opération lorsque le bras robotique récupère un couvercle inférieur dans le stock d'entrée dépendra de la longueur de la trajectoire entre la position du produit et la position de dépôt sur le tapis roulant.
- *Position du produit déposé dans le stock de sortie* : c'est le même principe que précédemment sauf que cette fois on s'intéresse aux trajectoires du tapis roulant vers la position du dépôt du produit dans le stock de sortie.
- *Position initiale du robot* : la distance parcourue par le robot ne dépend pas que de l'emplacement du produit mais aussi de la position initiale du robot.

Nous avons effectué des mesures en faisant varier les 2 facteurs influençant les trajectoires du bras robotisé à savoir, le remplissage du stock d'entrée et le remplissage du stock de sortie.

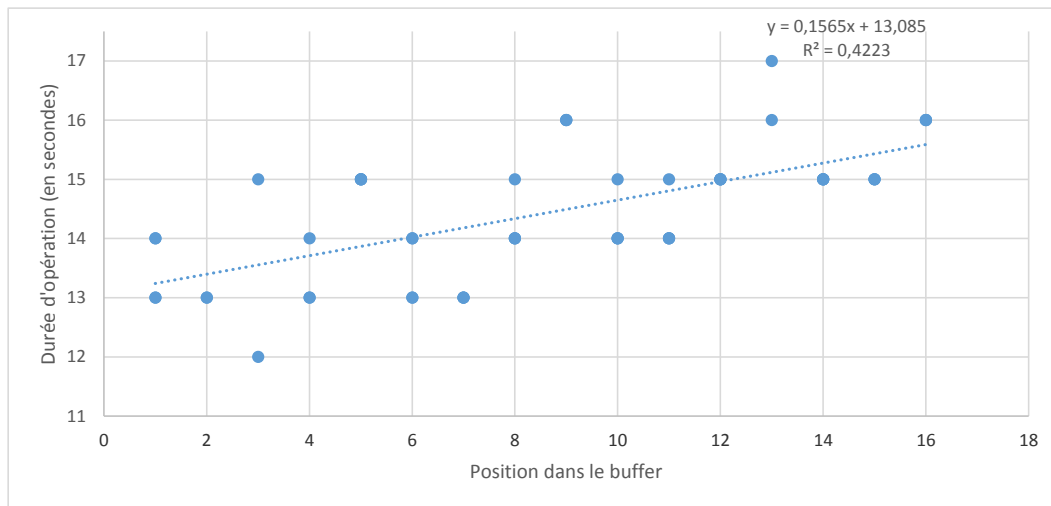


FIGURE 2.28 – Durées des opérations de prise de stock d'entrée réalisées par le Robot 1

Les emplacements dans les stocks d'entrée et de sortie représentent deux matrices  $4 \times 4$  numérotées de 1 à 16. Les Figures 2.28 et 2.29 représentent le nuage de points de la durée des opérations en fonction des emplacements vides dans les stocks d'entrée et de sortie de ligne de production respectivement.

Les graphiques montrent bien que les durées des opérations du poste *Robot1* sont influen-

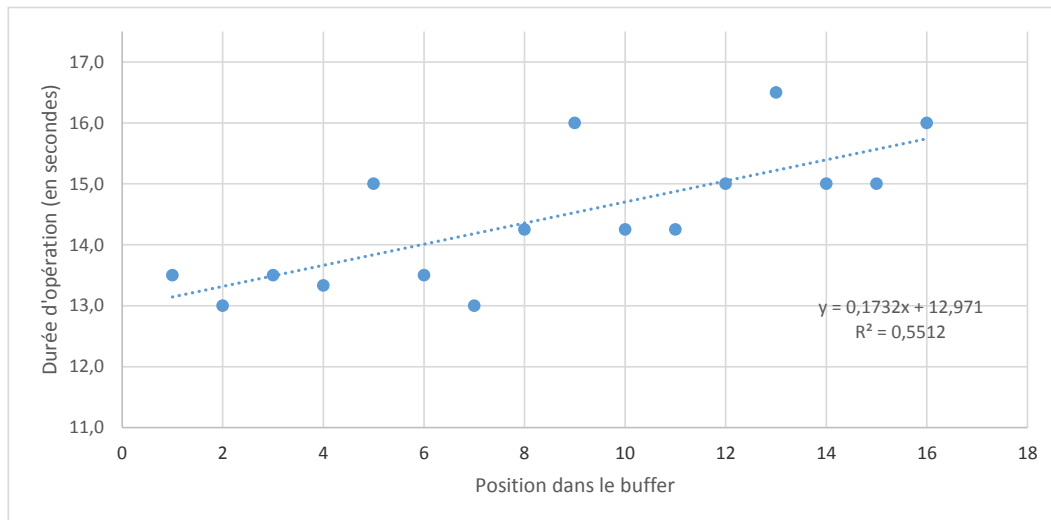


FIGURE 2.29 – Durée de l'opération de sortie des produits

cées par la position des pièces dans le magasin de stockage et par la trajectoire choisie par le robot. Ceci augmente la variabilité de ce poste de travail. Afin de modéliser correctement cette variabilité nous avons choisi d'extraire le modèle stochastique de ce temps en utilisant l'échantillon des temps mesurés et la fonction de *fitdistr()* du logiciel R. Puisque les temps de chargement et de déchargement des stocks d'entrée et de sortie sont dans la même plage, nous allons les modéliser avec la même loi de probabilité. La fonction *fitdistr* donne de bonnes approximations de notre échantillon selon la loi normale de moyenne de  $m = 14.32$  et un écart type de  $\sigma = 1.28$  ou la loi log-normal d'une moyenne de  $mlog = 2.66$  et un écart type de  $olog = 0.089$  comme le montre la Figure 2.30 qui représente la distribution de l'échantillon de mesure ainsi que les distributions des lois normales et log-normales. Nous avons choisi de considérer la loi normale vu la facilité de son implémentation sur CPN TOOLS.

#### 2.5.2.8.2 Robot2 :

Le *Robot2* réalise l'une des opérations les plus complexes de toute la ligne de production. Cette opération est composée de huit étapes ce qui augmente la probabilité d'avoir de la variabilité sur ce poste. Les causes de variabilité de temps de production du *Robot2* que nous avons pu identifier sont les suivant :

- *Sens du couvercle inférieur* : la première action de ce robot est de vérifier le bon positionnement du couvercle inférieur et de le retourner si celui-ci est à l'envers. Cette action prend un temps supplémentaire comparé au cas où la pièce est dans le bon sens.
- *Nombre de fusibles* : le producteur peut insérer de 0 à 2 fusibles sur la carte électronique, ce qui se traduit par des temps d'opération différents.
- *Position des fusibles* : si le producteur décide d'insérer un seul fusible, celui-ci peut être placé soit à gauche de la carte électronique, soit à droite, ce qui correspond à

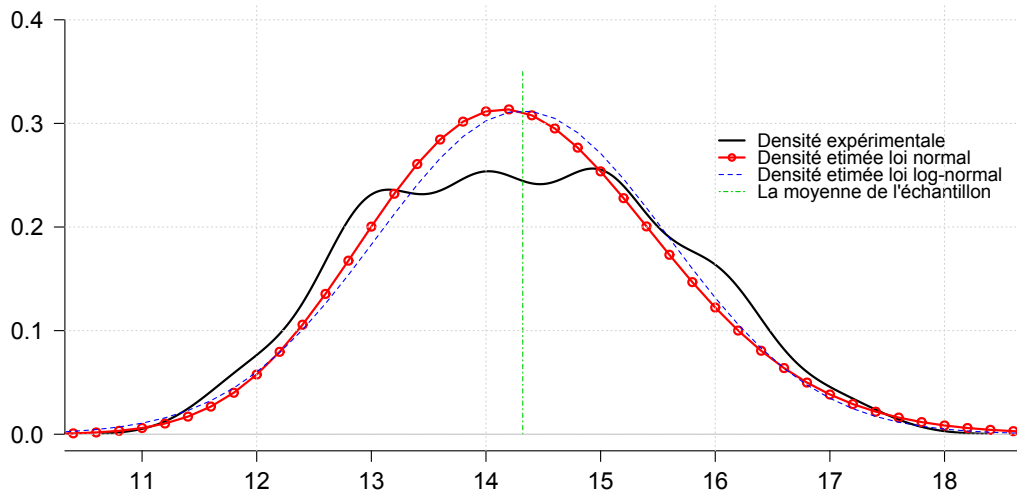


FIGURE 2.30 – Approximation par la loi normal ou log-normal du temps opératoire du *Robot1*

plusieurs trajectoires différents et donc des temps de déplacement différents.

- *Stock de fusibles* : il existe 3 stocks différents de fusibles, un changement de stock entrainerait un temps d'opération différent
- *Position des cartes électroniques* : les cartes sont situées sur une palette contenant 12 emplacements distinctes ce qui génère des trajectoires différentes et donc des temps différents pour l'installation des fusibles.

Afin de déterminer un modèle mathématique d'évolution des durées d'opération pour le poste *Robot2* nous avons relevé les temps de processus relevés par les capteurs. Nous avons extrait la densité de probabilité de l'échantillon de mesure comme le montre la Figure 2.31. Nous avons approximé cette distribution par plusieurs distributions usuelles en utilisant la fonction `fitdistr()` du logiciel R. Nous avons remarqué qu'une approximation par une loi gamma de paramètres  $k = 4710$  et  $\theta = 0.02$ . la lois log-normal permet aussi d'approximer la loi de variation du temps de processus avec une moyenne de  $m = 4.79$  et une écart type de  $\sigma = 0.014$ . Nous avons retenu la loi gamma vu que c'est la loi qui modélise souvent plusieurs files d'attentes.

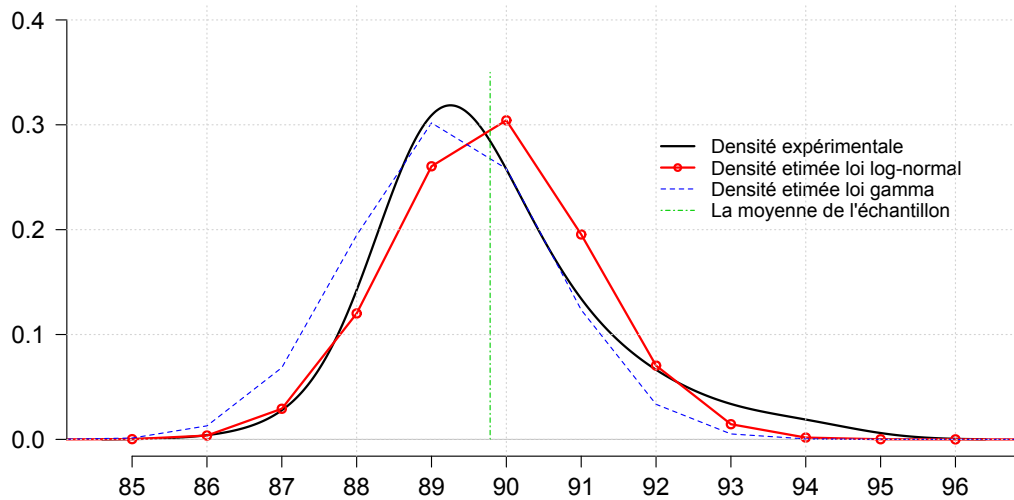


FIGURE 2.31 – Approximation par la loi log-normal ou gamma du temps opératoire du *Robot2*

## Modélisation et simulation de la ligne de production à l'aide des RdPC

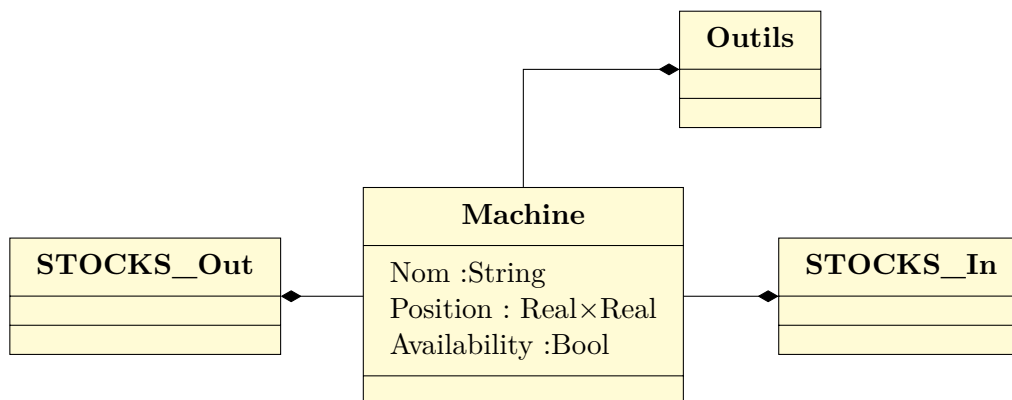


FIGURE 2.32 – Modèle objet d'un poste de travail

La ligne de production de CESI étant constituée de six postes de travail, le modèle basé sur les RdPCH le sera lui aussi. Dans cette partie nous présenterons le modèle RdPC de la ligne du CESI ainsi que ses organes notamment les postes de travail et le système de transport des produits entre les postes.

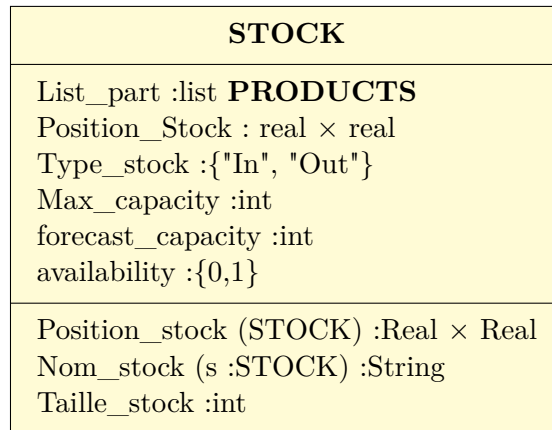


FIGURE 2.33 – Modèle objet d'un stock

### Modélisation des postes de travail

Chaque poste de travail est représenté par un RdP Modulaire de machine (cf. Figure 2.32), tel que défini à la Section 2.3.6. Cependant il faut noter que le modèle de la ligne de production a la spécificité suivante : lorsqu'un produit arrive au niveau d'une machine il doit la traverser même s'il ne subit pas d'opération sur cette machine. D'une certaine manière, le produit occupe l'espace de travail de la machine, même si cela ne dure que le temps de passage. Pour modéliser cette situation nous avons modifié la fonction d'arc sortant  $F_s(DP \rightarrow M1) = EntreeProduitMachine$  (cf. Figure 2.15) de la manière suivante :

```

fun EntreeProduitMachine (s :STRING, item :ITEM, t :TIME, n :INT) =
if (CheckMachine (item , s) = true)
then 1`Name s++
1`Product (Transformation_produit (item , t , n))
else 1`Name s++1`Product (ModifierDueProduit (item , t))

```

Nous avons rajouter une condition selon laquelle si le produit ne subissait pas d'opération au niveau de cette machine alors on le laisserait la traverser.

### Modélisation du transport

Nous avons rajouté un réseau de pétri modulaire pour la modélisation du transport entre deux postes de travail. Ce réseau de pétri modulaire est présenté sur la Figure 2.34.

La place **E** représente le stock de sortie d'une machine. Elle représente aussi une place d'entrée du réseau de Pétri Modulaire du transport entre deux machines. La place **D** représente le stock de la machine qui suit celle possédant le stock **E**. La place **D** est aussi une place de sortie du réseau de Pétri Modulaire du transport entre deux machines . La place **P** est celle du produit elle symbolise le produit qui est en train d'être transporté. La place **T** contient le temps de transport entre le stock **E** et le stock **D** c'est aussi une place d'entrée du RdPM. Le modèle de la ligne de production du CESI est donc formé

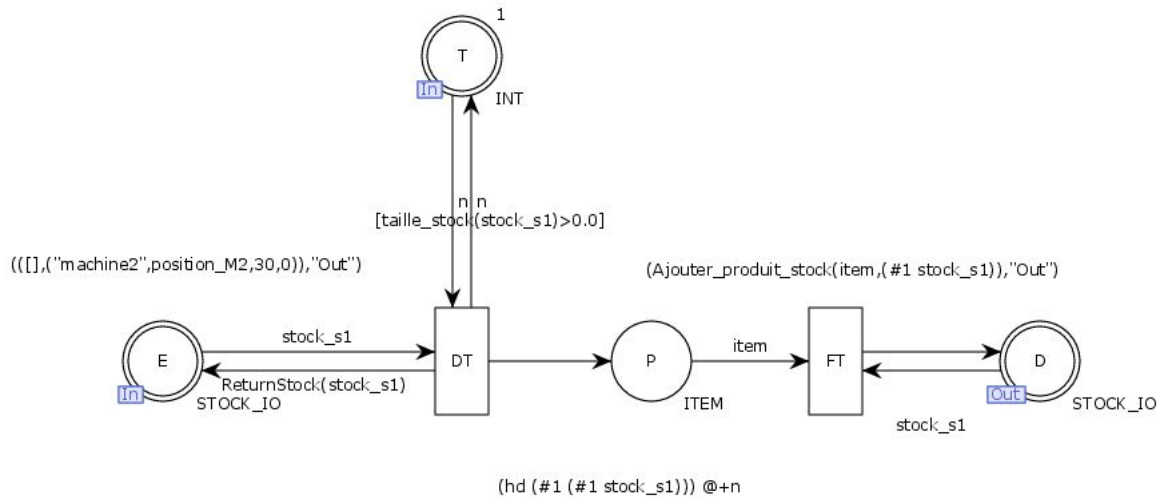


FIGURE 2.34 – RdPM du transport (convoyeur) de la ligne de production du CESI

d'un RdPH constitué des RdPs modulaires des machines et du système de transport qui les relie.

### Modèle global de la ligne de production

Le modèle globale de la ligne de production est représenté sur la Figure 2.35 où on peut observer les modèles hiérarchiques de tout les organes de la lignes production ( machines et transports). Les transitions **ROBOT1**, **PERCEUSE**, **FOUR**, **ROBOT2**, **Dépose Couvercle**, **PRESSE** sont des transitions de substitution qui représentent les RdPM (Réseaux de Pétri Modulaires) des postes de travail représentés sur la Figure 2.15. Les transitions **T1**, **T2**, **T3**, **T4**, **T5**, **T6** sont les transitions de substitution de l'activité de transport représenté par le RdP Modulaire de la Figure 2.34.



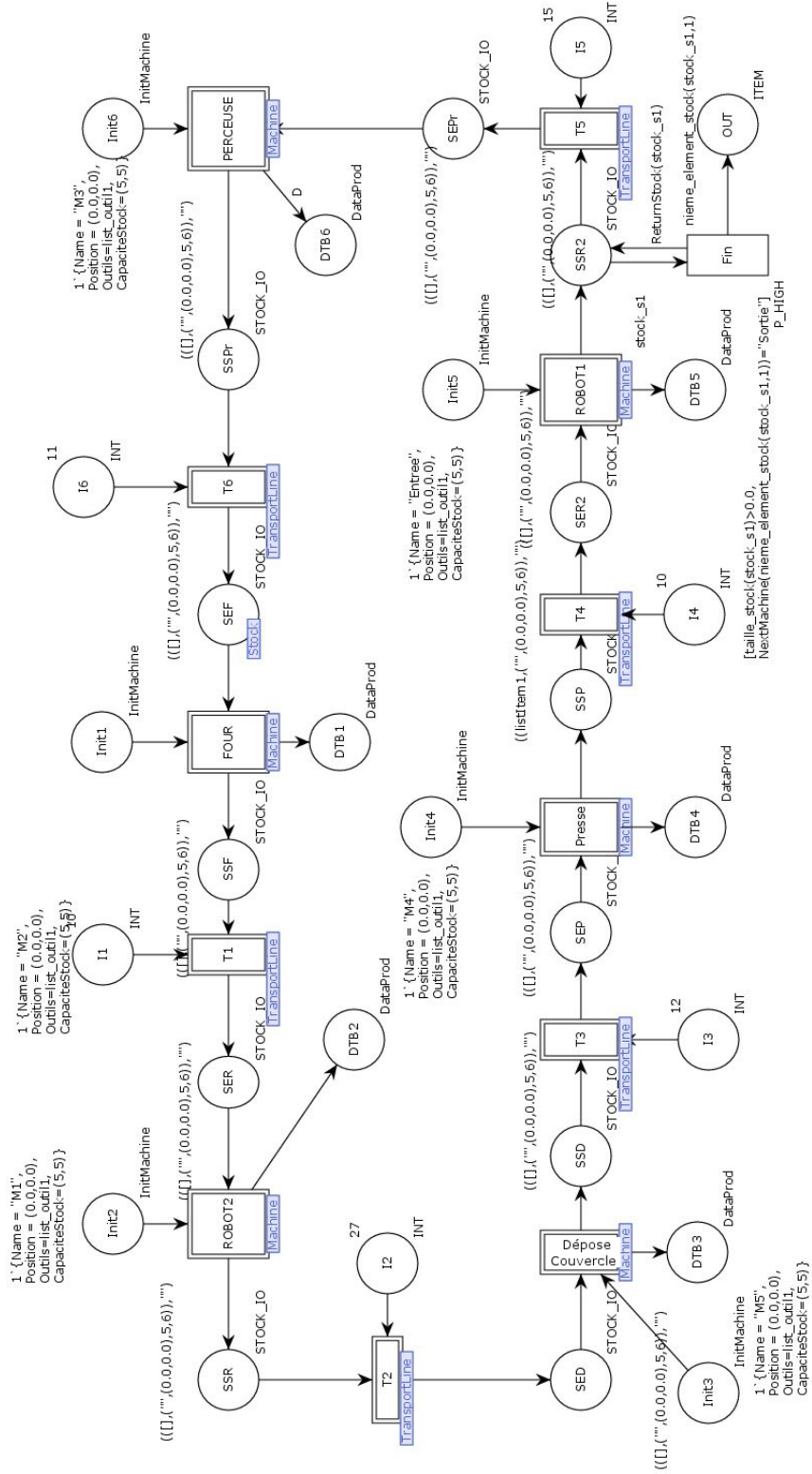


FIGURE 2.35 – Modèle Hiérarchique de la ligne de production du CESI de Rouen

Sur la Table 2.2, nous avons expliqué le rôle de chaque transition de substitution, notamment les différents postes de travail de la ligne de production.

Transition	Description des transitions
ROBOT1	Le Robot 1 a deux opérations : assure l’approvisionnement en début de processus et le transport des produits finis vers la zone de dépôt
PERCEUSE	Cette transition représente l’activité du poste <i>Perceuse</i> qui permet de percer le couvercle inférieur du téléphone et de pouvoir fixer la plaque électronique
FOUR	Cette transition représente l’activité du Four qui aide à sceller les deux parties du téléphone
ROBOT2	Cette transition représente l’activité du Robot 2 qui assure la pose des composants électroniques sur le téléphone
Dépose Couvercle	Cette transition l’activité du poste <i>DéposeCouvercle</i> qui permet d’effectuer le dépôt de couvercle supérieur du téléphone
PRESSE	Cette transition représente l’activité du poste <i>Presse</i> qui permet de bien coller les deux parties du téléphone

TABLE 2.2 – Les transitions du modèle de la ligne de production

Les transitions de substitutions **T1, T2, T3, T4, T5** et **T6** sont des RdP Modulaires de type transport. Nous les avons expliquées et illustrées dans le paragraphe 2.5.3.2 et par les Figures 2.34. Ces transitions de substitutions permettent de modéliser le transport entre chaque deux paires de postes de travail comme le montre la Figure 2.35.

### Validation des modèles par la ligne de production.

Pour valider notre modèle nous allons vérifier que le temps total d’écoulement des produits (makespan) du modèle de la ligne corresponde à celui de la ligne de production réelle. Pour cela nous allons prendre deux scénarios de production. Dans le premier cas, nous allons comparer le makespan d’un lot composé d’un seul produit. Dans le deuxième scénarios nous allons comparer makespan d’un lot composé de huit produits.

#### 2.5.3.4.1 Validation du makespan d’un seul produit

Nous avons lancé la production d’un produit et nous avons relevé la durée des opérations de fabrication de ce dernier dans la table 2.3 :

Poste de travail	Durée de l'opération
<i>DepotCouvercle</i>	<i>2 secondes</i>
PRESSE	<i>6 secondes</i>
PERCEUSE	<i>7 secondes</i>
FOUR	<i>25 secondes</i>
ROBOT1(entrée/sortie)	$\hookrightarrow N(m = 14.32, \sigma = 1.28)$
ROBOT2	$\hookrightarrow \Gamma(k = 4710, \theta = 0.02)$

TABLE 2.3 – Durée des opérations du produit dans le cas 1

Le temps de production total relevé pour ce produit est de

$$t_1 = 360 \text{ secondes}$$

Nous allons donc à présent déterminer le temps de production à l'aide de modèle basé sur les RdPCH.

### Simulation de la ligne de production

Les données d'entrées du système sont données dans les annexes ?? . Nous avons simulé le modèle de la ligne de production afin d'observer le temps d'écoulement d'un seul produit. et nous avons obtenu :

$$t_{m1} = 360640 \text{ ms} = 360,64 \text{ secondes}$$

pour un seul produit

Comparativement au temps d'écoulement réel d'un produit, on a une différence de 0.64 s qui peut s'expliquer par la variabilité des postes de travail *Robot1* et *Robot2* . Ce résultat nous permet d'ores et déjà de vérifier et de valider le modèle de notre machine. Il permet aussi la validation de la modélisation du transport. Cependant cette première étape ne permet pas encore de valider le comportement de la ligne de production lorsqu'il y a plusieurs produits. C'est la raison pour laquelle nous allons faire la simulation de production d'un lot de huit produits de même type que dans le scénario 1.

#### 2.5.3.5.1 Validation du temps d'écoulement de huit produits

Tout comme dans le scénario 1 nous allons lancer la production de huit produits sur la chaîne de production réelle et lancer une simulation de la production à l'aide de nos modèles; ensuite nous allons relever le makespan réel et celui obtenu par simulation. Vu la nature stochastique du comportement de la ligne notamment les postes de travail

*Robot1* et *Robot2* nous avons effectué plusieurs simulations (30 simulations) et plusieurs lancements de production d'un lot de 8 produits (30 lancements). nous avons trouvé une valeur moyenne du makespan réel de  $Makespan_R = 1092 \text{ secondes}$ , alors que la valeur moyenne de makespan simulé est :  $Makespan_S = 1101 \text{ secondes}$  qui représente une différence inférieure à 0.8% ce résultat est largement suffisant pour considérer notre modèle comme valide.

La différence entre les deux valeurs moyennes peut s'expliquer par l'erreur dans la mesure des temps de déplacement entre les postes de travail ainsi que les erreurs dans les mesures des temps de processus de chaque machine de production. En effet, nous avons effectué ces mesures à l'aide d'un chronomètre pour obtenir les temps de déplacement des produits, et nous avons directement prélevé les temps opératoires des portes de travail à partir du système de supervision de la ligne que lui donne des temps avec une précision d'une seconde.

## Conclusion

Dans ce chapitre, nous avons deux objectifs : le premier était la définition d'un cadre de modélisation structurelle d'un système de production flexible et le deuxième le développement de modèles de simulation pour des ateliers de production flexibles.

Pour définir un cadre de modélisation structurelle nous nous sommes appuyés sur le formalisme UML afin de construire l'ensemble des classes d'objets qui à notre sens étaient essentielles pour représenter et simuler un système de production flexible. Pour nous, les objets qui étaient indispensables étaient les produits, les stocks, les machines, et les ressources de transport. De cette manière, nous avons défini les classes **Produit** (section 2.2.1), **STOCK** (section 2.2.3), **Machine** (section 2.2.4) et la classe **Transport** (section 2.2.5) qui représente une ressource de transport. Les quatre classes que nous venons de citer représentent les principaux objets du modèle d'un système de production flexible. Cependant ces dernières ont des dépendances vis-à-vis d'autres classes. Par exemple nous avons montré que la classe **Produit** était **Operation**<sup>2</sup> et **Outil**<sup>3</sup>.

Pour développer un modèle de simulation nous avons fait une description sous le formalisme des Réseaux de Petri Colorés temporisés Hiérarchiques des modèles objets définis dans le cadre de la modélisation structurelle. C'est ainsi que nous avons présenté le modèle de simulation d'un produit (section 2.3.3), d'un stock (section 2.3.4), d'une ressource de transport (section 2.3.5) et d'une machine flexible (section 2.3.6). Tout comme pour le cas de la modélisation structurelle le développement des quatre modèles de simulation précédents dépendent de sous-modèles tels que le modèle de simulation d'un outil défini en section 2.3.1) ou encore celui d'une opération machine défini en section 2.3.2.

Après avoir défini ces différents modèles de simulation, nous avons procédé à une phase de validation indispensable dans le cadre d'un travail de modélisation. Celle-ci avait deux

---

2. Cette classe permet de décrire ce que l'on entend par une opération machine et est définie en section 2.2.2

3. Celle-ci décrit un outil qui possède une machine afin d'exécuter une opération sur un produit

objectifs : le premier était de vérifier le fonctionnement individuel de chaque modèle. Le deuxième objectif était de valider le comportement d'un système de production flexible lorsque les différents éléments (produits, machines, stock et ressources de transport) sont en interaction. Pour la validation nous nous sommes servis d'un cas réel à savoir la chaîne de production du CESI de Rouen (section 2.5). De cette manière nous avons lancé deux scénarios de production. Un premier scénario dans lequel nous avons lancé sur la chaîne de production un produit et mesurer son temps total de production. Nous avons ensuite fait la simulation de ce scénario de production à l'aide d'un modèle de simulation de la ligne que nous avons construit à l'aide des modèles développés en section 2.3 afin de faire une validation fonctionnelle de chaque modèle développé. Les résultats que nous avons obtenus au cours de la simulation nous ont permis de constater qu'il y avait un écart de moins de 1% entre le modèle de la ligne de production du CESI et le système réel. Ce sont ces observations qui nous ont poussé à considérer notre modèle comme valide. Dans le deuxième scénario, nous avons lancé sur la chaîne de production huit produits afin de mesurer le temps total de production de ce lot de huit produits. Ensuite nous avons effectué la simulation de ce scénario sur notre modèle de simulation afin de mesurer l'écart entre le temps total de production réel et celui obtenu par simulation. Une fois de plus nous avons obtenu des écarts inférieurs à 1% ce qui nous a permis à ce moment-là aussi de valider notre modèle (section 2.5.3.5.1).

Cependant il est important de noter que ces deux expériences nous ont effectivement permis de valider le fonctionnement des modèles de machines, des produits, stocks et ressources de transport. Mais cela ne nous garantit pas forcément que pour chaque système réel que le modèle sera valide. Cela n'est pas surprenant d'autant plus qu'il n'existe pas de modèle universel d'un système de production flexible.

Observons aussi que dans le cas du chapitre 2, nous avons étudié le cas d'une chaîne d'une production avec un flux de matière unidirectionnel. Cela a considérablement simplifié l'interaction entre les produits et les ressources de transport. De plus la gestion des files d'attente sur une ligne de production suit en général la règle du premier arrivé premier sorti. Mais qu'en serait-il si nous avions à modéliser un système de production donc les flux sont croisés, et dont les ressources de transport sont autonomes<sup>4</sup>. Comment pourrions nous gérer le transport des produits en coordonnant les ressources de transport entre elles. C'est cet aspect des choses que nous proposons d'aborder dans le chapitre 3 où nous abordons le problème de transport dans un atelier de production flexible.

---

4. le convoyeur d'une ligne de transport est un système mécanisé non autonome il relie toujours deux mêmes postes de travail

# Chapitre 3

## Algorithme de gestion des activités de transport dans un atelier de production flexible

### Introduction

Dans le chapitre 2, nous avons présenté des modèles structurels nous permettant de construire un système de production flexible. Dans ce chapitre, nous allons nous servir de ces modèles pour résoudre le problème d'acheminement des produits dans un atelier de production flexible à flux croisés. En effet, l'acheminement des produits est l'une des tâches les plus complexes dans la gestion des ateliers de production flexibles [81]. De plus c'est une activité centrale qui touche tous les autres services de l'atelier tels que la production, l'inspection, la maintenance. Il est donc indispensable de traiter ce problème si l'on veut proposer un modèle permettant de représenter des ateliers de type JOB SHOP par exemple. En apportant une solution au problème de transport nous allons ainsi résoudre celui du partage des ressources dans un atelier de production flexible. Dans ce chapitre nous proposons une méthodologie de gestion en temps réel de l'acheminement des produits comprenant une méthode d'aide à la décision. Cette méthode d'aide à la décision sera intégrée à nos modèles développés à l'aide des Réseaux de Pétri. Comme toute méthode d'aide à la décision celle-ci s'appuie sur un modèle de données et des critères d'évaluation. Nous avons traduit notre stratégie d'aide à la décision en un algorithme que nous avons intégré dans les modèles de Réseaux de pétri que nous avons définis précédemment. L'objectif ce travail était de rendre notre modèle de simulation utilisable afin de d'évaluer les performances d'un système de production par le biais de la modélisation et de la simulation. .

Ce chapitre s'articule de la manière suivante :

- A la section 3.2 est présenté le problème d'acheminement des produits dans un atelier de production flexible du transport, et les contraintes que nous nous fixons dans le cadre de notre travail. .

- Dans la section 3.3, Nous présentons notre méthodologie de travail et notamment le modèle de données que nous allons utiliser pour mettre en place notre stratégie d'aide à la décision. Ce modèle se présente sous la forme d'une matrice appelée table d'évaluation des activités de transport. Dans cette section nous présentons la nature des critères de cette table.
- En section 3.4, nous présentons un critère particulier de la table d'évaluation des activités de transport appelé flexibilité de routage. Cet indicateur joue pour nous un rôle essentiel dans la mesure de la flexibilité de routage ainsi que que dans les prises de décision pour la résolution des problèmes de transport.
- Dans la section 3.5, nous présentons l'algorithme d'aide à la décision pour la résolution du problème de transport que nous avons développé.
- En section 3.6, nous avons présenté un cas d'étude dans lequel nous utilisons les modèles réseaux de pétri ainsi que l'algorithme d'aide à la décision afin d'étudier les performances d'un atelier de production flexible de type JOB SHOP.

## Le problème du transport dans une cellule de production flexible

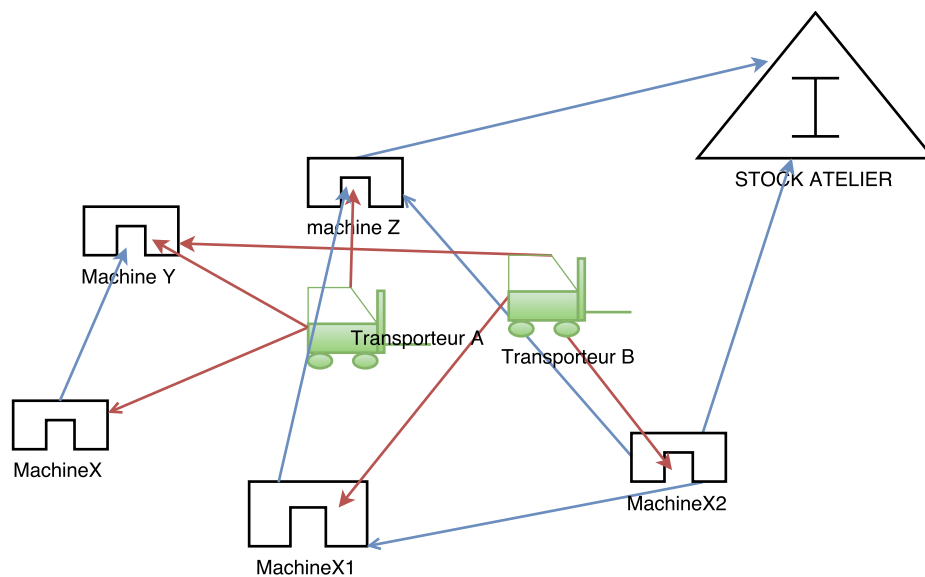


FIGURE 3.1 – Schéma illustrant le problème de transport dans une unité de production

### Définition et contraintes

La gestion du transport consiste à assurer le déplacement des produits en cours de production, ainsi que les produits finis des zones de chargement, vers de zones de déchargement afin que ces derniers subissent des opérations de production, ou qu'ils soient

expédiés (produits finis). Le transport dans un atelier de production doit respecter les contraintes logistiques (ressources limitées, nombres de machines, positions des équipements, temps de production etc.) et les contraintes contractuelles et réglementaires (délai, qualité, coût etc.). Une opération de transport consiste au déplacement d'un produit d'une zone de chargement vers une autre zone de déchargement. Une opération de transport nécessite obligatoirement une ressource de transport autonome. Cette dernière peut être :

- Un ou plusieurs opérateurs transportant des produits, soit manuellement, ou alors à l'aide de transpalette ou au moyen de chariots élévateurs
- Un ou plusieurs robots automatisés qui se coordonnent pour assurer le déplacement des produits.

Cependant, à un instant donné dans un atelier en îlots de production, il peut arriver que le nombre d'opérations de transport soit supérieur à celui des ressources disponibles ; dans ce cas de figure, c'est au gestionnaire du système de production de trouver une stratégie pour résoudre ce problème de partage des ressources. Cette stratégie doit prendre en compte les contraintes de la logistique interne de l'atelier (gestion des ressources de transport, position des machines, temps de processus), mais aussi les contraintes externes à l'atelier (variation de la demande, respect des délais etc.) C'est ainsi que nous définissons le problème de transport. La résolution de ce dernier doit donc prendre en compte la charge des machines, afin d'éviter de les saturer, les temps de production.

Pour traiter le problème de transport nous travaillerons avec les hypothèses suivantes :

- Chaque machine ou poste de travail possède deux types de stocks : un stock d'entrée où arrivent les produits, et un autre de sortie d'où ces derniers repartent lorsqu'ils ont subi une opération. Les stocks en entrée des machines constituent des zones de déchargement, tandis que ceux en sortie des zones de chargement pour les transporteurs.
- Les positions physiques des stocks d'une machine sont confondues avec celles des machines/postes de travail auquel ces derniers sont rattachés.
- Chaque ressource de transport est capable de se déplacer librement d'une machine à une autre. Cela signifie qu'il y a une flexibilité totale des moyens de manutention.
- Le déplacement de la ressource assurant le transport entre deux machines est rectiligne et uniforme.
- La vitesse de déplacement des ressources de transport, ainsi que les temps de chargement et déchargement sont constants.
- La gestion des files d'attente des machines se fera suivant des règles de dispatching qui seront définies de manière locale au niveau de chaque machine.



## Description du problème du transport dans un atelier de production

Dans un atelier de production flexible, plusieurs facteurs influencent le transport des produits dans l'atelier et par conséquent les performances globales du système de production. Prenons l'exemple d'un atelier où il y a deux machines A et B pouvant effectuer une même opération  $O_i$  d'un produit  $i$ . Cela signifie que la ressource de transport à un moment donné pourra choisir de transporter un produit nécessitant une opération O vers la machine A ou vers la machine B. Dans ce cas de figure comment devra réagir le système de transport face à cette flexibilité. Il faut aussi noter que en choisissant la machine A au lieu de la B, le transporteur impose d'une certaine manière une charge à cette dernière et modifie son ordonnancement. Enfin lorsque dans une zone de chargement plusieurs produits sont susceptibles d'être transportés, à qui devrait revenir le choix du produits à transporter en premier ? Cet exemple nous a poussés à identifier les problèmes de transport en fonction du niveau d'implication du système de transport dans la gestion des files d'attente, mais aussi de la flexibilité des produits ou des machines. Nous allons classifier le problème de transport dans notre atelier de production en fonction de l'impact que les décisions de transport peuvent avoir sur l'affectation des ressources, et l'ordonnancement des files d'attente des produits au niveau des machines. Ainsi nous allons définir quatre cas de figure qui pourront se combiner dans un même atelier de production. Nous définirons ces cas de figure à l'aide de quatre propositions :  $P$ ,  $E$ ,  $F$  et  $T$

- $P$  : L'atelier de production possède une flexibilité d'un ou de plusieurs produits. Un produit est flexible si une ou plusieurs de ses opérations de fabrication peuvent être effectuée par plusieurs machines de l'atelier.. Lorsqu'aucun produit de l'atelier de production ne possède de flexibilité, nous noterons  $\bar{P}$ , la proposition contraire de  $P$ .
- $E$  : La Stratégie de Gestion du Transport (SGT) modifie l'ordonnancement de la file d'attente des produits en sortie des machines ou en entrée de l'atelier de production. Cela signifie que dans une opération de transport la ressource peut choisir aisément le produit qu'elle va transporter dans la zone de chargement ; celui-ci ne lui ai pas imposé par une stratégie extérieure de gestion de la file d'attente des stocks. En résumé dans ce cas de figure la stratégie de gestion des files d'attente des stocks en sortie ou en entrée des machines n'est pas gérée de manière locale au niveau de chaque machine. Dans le cas contraire nous noterons  $\bar{E}$
- $F$  La SGT modifie l'ordonnancement de la file d'attente en entrée des machines. Dans le cas contraire nous noterons  $\bar{F}$
- $T$  la SGT inclut l'affectation des ressources de transport. Cela signifie que, dans le cas où il y a plusieurs ressources disponibles pour effectuer une opération de transport, la stratégie d'affectation d'une ressource à cette opération est définie de manière dynamique en fonction de la configuration actuelle du système. Dans le cas contraire nous noterons  $\bar{T}$

Par mesure de simplification nous allons définir le problème de transport par les variables  $\alpha, \beta, \gamma, \delta$  tel que  $SGT = (\alpha|\beta|\gamma|\delta)$  avec :

- $\alpha \in \{P, \bar{P}\}$  et renseigne sur la flexibilité des produits dans l'atelier de production,
- $\beta \in \{E, \bar{E}\}$  renseigne sur la manière dont est choisi le produit qui doit être transporté,
- $\gamma \in \{F, \bar{F}\}$ , renseigne sur la manière dont les zones de déchargement sont réorganisées lorsqu'une opération de transport est effectuée,
- $\delta \in \{T, \bar{T}\}$  renseigne sur la stratégie d'affectation des ressources de transport à une opération de transport.

Nous venons de définir les paramètres qui influencent la gestion du transport dans un atelier de production. Dans la prochaine section nous expliquerons la méthodologie que nous avons mise en place pour la résolution des problèmes de transport notamment dans le partage des ressources de transport et dans l'affectation des opérations aux machines.

## Présentation de la méthodologie d'aide à la décision pour la gestion du transport

### Objectifs et étapes méthodologiques

La méthode d'aide à la décision que nous proposons dans le cadre de ce chapitre doit permettre d'assurer la simulation des flux en gérant les conflits liés au transport dans l'atelier de production et ainsi d'empêcher le blocage du modèle de simulation au cours de l'exécution d'une séquence de production. De plus la méthode d'aide à la décision que nous proposons permet de proposer une stratégie d'ordonnancement dont les buts sont :

- L'optimisation des charges des machines
- La minimisation de la distance parcourue par les ressources de transport
- La minimisation des temps moyens de production par produit
- L'augmentation de la flexibilité dynamique du routage du système de production

Pour la mise en place de la méthode d'aide à la décision, nous allons suivre les étapes méthodologiques présentées dans les travaux de [82]. Dans ceux-ci, l'auteur présente les quatre grandes étapes d'un processus d'aide à la décision à savoir :

1. Établir la liste des actions potentielles

2. Dresser la liste des critères à prendre en considération
3. Établir le tableau des performances
4. Agréger les performances

Dans la prochaine section nous présenterons la liste des actions potentielles.

## Liste des actions potentielles pour l'aide à la décision dans la gestion du transport

Nous avons défini une activité de transport comme l'ensemble des opérations de transport susceptibles d'être réalisées à un instant  $t$  de la production. Ainsi, l'ensemble des actions potentielles sera aussi celui des opérations de transport susceptibles d'être réalisés à un instant donné. Concrètement cela signifie que la notion d'activité de transport dépend du temps (ou de l'instant) et de la disponibilité des ressources de transport à ces dits-instants. Ainsi, à chaque instant de décision, il y aura des actions potentielles (opérations de transport) qui seront réalisables. [83] [84] [85] Nous appelons instant de décision tout instant où une ressource de transport est disponible, et où un ou plusieurs produits sont disponibles et prêts à être transportés.

Une opération de transport est caractérisée par les éléments suivants : [85]

- Le produit à transporter
- La zone de chargement ou le stock d'entrée dans lequel se trouve le produit à transporter
- La zone de déchargement vers laquelle le sera transporté le produit.
- La ressource qui devra permettre le transport du produit de la zone de chargement vers la zone de déchargement.

La définition des caractéristiques d'une activité de transport permet de déterminer pour une opération un ensemble de données au cours de la production. Par exemple la connaissance de la zone de déchargement permet d'avoir des données sur le nombre de produits et le temps d'attente au niveau de cette machine. La connaissance des couples zone de chargement et de déchargement permet de connaître la distance à parcourir par la ressource de transport afin d'effectuer cette opération de transport. Les données que l'on rattache ainsi à une opération de transport permettent de définir des critères de décision.

L'ensemble des actions potentielles (opération de transport réalisables) sera stocké à chaque instant de décision dans une table appelée table d'évaluation instantanée du transport que nous présentons en détail dans le paragraphe 3.3.4 . La Table d'Évaluation Instantanée du Transport permet de répertorier toutes les opérations de transport susceptibles d'être effectuées dans l'atelier de production. Cette table permet d'avoir toutes les opérations de transport et les critères d'évaluation de ces dernières. Le nombre de

lignes de la Table d'Évaluation Instantanée du Transport correspondra à celui des opérations qui constituent l'activité de transport à l'instant de décision. Dans le prochain paragraphe nous allons définir quelques critères d'évaluation des activités de transport. L'utilisation des tables nous donnera un cadre formel qui nous permettra d'identifier et de comparer les opérations de transport entre elles. La comparaison d'une opération de transport n'a de sens que si elle est faite sur la base de critères. L'utilisation des critères joue un rôle dans la gestion des conflits liés aux ressources de transport. Nous verrons dans le paragraphe 3.3.3, comment sont définis les critères d'évaluations et quelle est leur nature.

## Les critères d'évaluation des activités de transport

Un critère est une fonction  $g$ , définie sur l'ensemble des action  $A$ , qui prend ses valeurs dans un ensemble totalement ordonné et qui présente les préférences de l'utilisateur selon un point de vue[86]. Les critères permettent de faire des comparaisons(surclassement) entre deux actions potentielles. Ainsi, considérons deux actions potentielles  $a$  et  $b$  et un critère  $g_i$  si

$$g_i(a) > g_i(b) , \quad (3.1)$$

on dit que l'action  $a$  surclasse la  $b$  suivant le critère  $i$ . Dans ce cas de figure l'action  $a$  sera préférée à l'action  $b$ . Dans le cas où il y a une égalité entre  $g_i(a)$  et  $g_i(b)$ , on dit que qu'il y a une relation d'indifférence entre les actions  $a$  et  $b$ .

- La distance entre un stock de départ et le stock d'arrivée ;
- La distance entre le stock de départ et une ressource de transport ;
- Le temps de parcours par une ressource de transport pour effectuer le transport entre un stock de départ et un stock d'arrivée ;
- Le taux de remplissage des zones de chargement et de déchargement ;
- La charge du stock d'arrivée (C'est la durée totale des opérations dans la file d'attente du stock d'entrée de la machine de destination.) ;

Dans prochain paragraphe nous verrons comment l'on peut se servir de la table d'évaluation de l'activité de transport pour contrôler l'activité de transport dans un système de production.

## Établissement de la Table d'Evaluation Instantanée de l'Activité de transport

La table d'évaluation instantanée de l'activité de transport est une matrice donc le nombre de lignes correspond au nombre d'opérations de transport réalisables et le nombre de colonnes correspond au nombre de critères de décision relatifs aux opérations de transport. Rappelons qu'une opération de transport n'est possible que s'il existe un produit

$p$  à transporter, d'une Zone de Chargement  $Zc$  vers une Zone de Déchargement  $ZD$  par une ressource de transport  $Tr$ .

Dans un premier temps définissons les ensembles suivants à chaque instant de décision. Soient :

- $Tr$ , l'ensemble des ressources de transport disponibles. C'est un ensemble fini et dénombrable. On peut donc définir un sous ensemble de  $\mathbb{N}$  noté  $\mathbb{L} = \{1, 2, \dots, l\}$  qui est l'ensemble des indices des éléments de  $Tr$ , de manière à ce que  $(Tr_n)_{n \in \mathbb{L}}$  soit une suite des éléments de  $Tr$ . Alors :  $Tr = \{Tr_1, Tr_2, \dots, Tr_l\}$
- $Pr$  : L'ensemble des produits prêts pour le transport. De la même manière que pour l'ensemble  $Tr$  on définit l'ensemble  $\mathbb{M} = \{1, 2, \dots, m\}$  des indices des éléments de  $Pr$  de manière à définir une suite  $(Pr_n)_{n \in \mathbb{M}}$  des éléments de  $Pr$ . De cette manière,  $Pr = \{Pr_1, Pr_2, \dots, Pr_m\}$
- $ZC = \{ZC_1, ZC_2, \dots, ZC_s\}$  : L'ensemble des zones de chargement disponibles avec  $(ZC_n)_{n \in \mathbb{S}}$  la suite des éléments de  $ZC$  et  $S = \{1, 2, \dots, s\}$  avec  $s$  le nombre de zones de chargement.
- $ZD = \{ZD_1, ZD_2, \dots, ZD_q\}$  L'ensemble des zones de déchargement disponibles, avec  $(ZD_n)_{n \in \mathbb{Q}}$  la suite des éléments de  $ZD$  et  $Q = \{1, 2, \dots, q\}$  avec  $q$  le nombre de zones de déchargement.
- $O = \{O_1, O_2, \dots, O_I\}$  l'ensemble des opérations de transports réalisables, avec avec  $(O_n)_{n \in \mathbb{Q}}$  la suite des éléments de  $O$  et  $\mathcal{I} = \{1, 2, \dots, I\}$  avec  $I$  le nombre d'opérations de transport susceptibles d'être réalisées à l'instant de décision.

### La définition des fonctions d'affectation des ressources

Nous aurons besoin d'attribuer à chaque opération de transport un transporteur, un produit, une zone de chargement ainsi qu'une zone de déchargement c'est pour cela que nous allons définir les fonctions  $\sigma$ ,  $\lambda$ ,  $\mu$ , et  $\phi$ .

Soit :

$$\begin{aligned} \sigma : \mathcal{I} &\rightarrow \mathbb{L} \\ i &\longmapsto \sigma(i) \end{aligned}$$

Rappelons que les ensembles  $\mathcal{I}$  et  $\mathbb{L}$  sont les ensembles sur lesquels sont indexés  $O$  et  $Tr$ . La fonction  $\sigma$  permet d'affecter à une opération de transport  $i$ , une ressource de transport disponible  $\sigma(i)$ . Alors,  $Tr_{\sigma(i)}$  : est la ressource de transport susceptible d'assurer l'opération  $i$ .

De même

$$\begin{aligned} \lambda : \mathcal{I} &\rightarrow \mathbb{M} \\ i &\longmapsto \lambda(i) \end{aligned}$$

La fonction  $\sigma$  permet d'affecter à l'indice d'une opération de transport  $i$ , l'indice du produit qui va être transporté. Alors,  $Pr_{\lambda(i)}$  est le produit transporté au cours de l'opération  $i$ .

$$\begin{aligned} \mu : \mathcal{I} &\rightarrow \mathbb{S} \\ i &\longmapsto \mu(i) \end{aligned}$$

La fonction  $\mu$  permet d'affecter à l'indice d'une opération de transport  $i$ , la zone de chargement  $ZC_j$  où le produit va être pris pour être transporté. De cette manière,  $ZC_{\mu(i)}$  est la zone de chargement de l'opération de transport  $i$

$$\begin{aligned} \phi : \mathcal{I} &\rightarrow \mathbb{Q} \\ i &\longmapsto \phi(i) \end{aligned}$$

La fonction  $\phi$  permet d'affecter à l'indice d'une opération de transport  $i$ , la zone de chargement  $ZD_j$  où le produit va être pris pour être transporté. De cette manière,  $ZD_{\phi(i)}$  est la zone de déchargement de l'opération de transport  $i$ .

## La Présentation de la table d'évaluation des activités de transport

Pour définir la table d'évaluation des activités de transport, considérons qu'à un instant de décision  $T$ , il ait un nombre  $I$  d'opérations de transport à réaliser. Considérons aussi que pour toute la phase de production il y a un nombre  $J$  de critères d'évaluation. La table d'évaluation instantanée de l'activité de transport sera alors celle de la table 3.1 la table

Opération	Tr	Pr	ZC	ZD	$critère_1$	...	$critère_j$	...	$critère_J$
1	$Tr_{\sigma(1)}$	$Pr_{\lambda(1)}$	$ZC_{\mu(1)}$	$ZD_{\phi(1)}$	$A_{11}$	...	$A_{1j}$	...	$A_{1J}$
2	$Tr_{\sigma(2)}$	$Pr_{\lambda(2)}$	$ZC_{\mu(2)}$	$ZD_{\phi(2)}$	$A_{21}$	...	$A_{2j}$	...	$A_{2J}$
3	$Tr_{\sigma(3)}$	$Pr_{\lambda(3)}$	$ZC_{\mu(3)}$	$ZD_{\phi(3)}$	$A_{31}$	...	$A_{3j}$	...	$A_{3J}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$
$i$	$Tr_{\sigma(i)}$	$Pr_{\lambda(i)}$	$ZC_{\mu(i)}$	$ZD_{\phi(i)}$	$A_{i1}$	...	$A_{ij}$	...	$A_{iJ}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$
$I$	$Tr_{\sigma(I)}$	$Pr_{\lambda(I)}$	$ZC_{\mu(I)}$	$ZD_{\phi(I)}$	$A_{I1}$	...	$A_{Ij}$	...	$A_{IJ}$

TABLE 3.1 – Table d'évaluation de l'activité de transport

d'évaluation possède  $I$  lignes et  $J + 5$  colonnes. chaque ligne représente une opération de transport. La première colonne donne le numéro de l'opération de transport, les quatre colonnes suivantes représentent les attributs liés à chaque opération à savoir la ressource de transport affectée (colonne  $Tr$ ), le produit à transporter (colonne  $Pr$ ), la zone de chargement (colonne  $ZC$ ) et la zone de déchargement ( $ZD$ ). Les  $J$  dernières colonnes représentent les critères des opérations de transport qui ont été au préalable définis par le gestionnaire du système de production.

La table d'évaluation que nous venons de définir permet de caractériser à chaque moment de prise de décision, l'activité de transport de l'atelier de production. Cependant même si dans notre travail nous avons considéré une flexibilité totale des moyens de manutentions cette table peut évidemment être exploitée dans un cas où des ressources de transport ont des chemins interdits.

Ce qui différencie une table d'évaluation du transport d'un atelier de production à l'autre, c'est le choix des critères d'évaluation des activités de transport. Ces derniers reflètent les éléments qui sont les plus importants pour les gestionnaires de l'appareil productif. Il faut souligner que pour représenter un problème d'aide à la décision on peut aussi faire le choix d'une représentation sous forme de graphe où les sommets seront des actions potentielles et les arcs orientés définiront les relations de surclassement. Cette représentation est proposée dans [87]. Si celle-ci permet une meilleure vision des relations de surclassement, on voit tout de même qu'il faut un graphe pour représenter les relations de surclassement suivant chaque critère. Cette représentation est intéressante s'il y a un faible nombre de critères et d'actions potentielles. La table que nous proposons permet d'utiliser les méthodes classiques d'aide à la décision à savoir les méthodes ELECTRE[88] et PROMETHEE[89]. Nous n'avons pas fait le choix de ces méthodes parce que ces dernières nécessitent une définition de poids pour les critères d'évaluation et nous voulions éviter de privilégier de manière statique un critère par rapport à un autre mais faire évoluer la l'ordre de priorité des critères au cours de la production en fonction des aléas.

La table d'évaluation des activités de transport nous a servi comme support pour nos méthodes d'aide à la décision. Dans la suite de notre travail, nous l'utiliserons pour définir métrique de la flexibilité d'acheminement des produits.

## La Flexibilité de Routage (FR)

Dans la littérature il existe plusieurs définitions de la flexibilité de routage. [13] définit la flexibilité de routage (FxR) comme l'aptitude d'un système de production à faire face aux pannes tout en continuant la production d'un ensemble de pièces donné. Cette aptitude existe dans un système de production si un produit peut être fabriqué par différents processus ou alors si une opération peut être réalisée par plusieurs machines

[90] définit la flexibilité de routage comme l'aptitude d'un système de production à utiliser des chemins alternatifs afin de produire un ensemble de pièces. Dans les travaux de [13, 91], Les auteurs prennent en considération trois aspects de la flexibilité de routage : l'efficacité, la versatilité, et l'homogénéité de la distribution des chemins alternatifs. L'efficacité de la flexibilité de routage considère le coût opérationnel et temporel des routes alternatives pour produire un ensemble de pièces. De cette manière, dans [92] présente l'efficacité d'acheminement en l'assimilant à l'efficacité d'un chemin pour la production d'un bien. C'est ainsi que l'on a la relation suivante :

$$e_{ij} = \frac{\text{Min}\{F_{ij}\}}{F_{ij}} \quad i = 1, 2, \dots, r \quad (3.2)$$

$F_{ij}$  est le temps nécessaire pour la production d'un produit  $j$  par un chemin  $i$ .  $r$  est

le nombre total de chemins que l'on pourrait utiliser pour fabriquer un produit  $j$ . La versatilité de la flexibilité de routage considère que plus il y a de chemins pour fabriquer un produit, plus le système est flexible. Elle prend en compte le nombre de chemins alternatifs possibles. L'auteur propose ainsi de mesurer l'efficacité du routage d'un produit par la relation suivante.

$$E_j = \frac{1}{r} \sum_{i=1}^r e_{ij} \quad (3.3)$$

Cela signifie que l'efficacité de routage d'un produit est égale à la moyenne des efficacités de routage des chemins possibles pour réaliser ce produit.

En ce qui concerne la versatilité du routage, [93] propose un indicateur en utilisant une approche basée sur l'entropie. Cette approche permet de mesurer la flexibilité d'une machine mais aussi la versatilité du routage. C'est ainsi que l'auteur propose l'indicateur suivant :

$$V_{ij} = -\alpha_{ij} \log_{10} \alpha_{ij} \quad (3.4)$$

$V_{ij}$  représente la versatilité du routage du chemin  $i$  pour la production d'un produit  $j$  et  $\alpha_{ij}$  la valeur normalisée de l'efficacité du routage définit par

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{i=1}^r e_{ij}} \quad (3.5)$$

Le calcul de la versatilité du routage  $R_j$  pour une pièce  $j$  sera alors donnée par la relation suivante :

$$R_j = \sum_{i=1}^r V_{ij} \quad (3.6)$$

La variété du routage est définit par [92] comme la différence qui existe entre plusieurs chemins pour la fabrication d'un même produits. le même auteur exprime cette variété par la relation suivante :

$$S_{il} = \frac{R_i \cap R_l}{m} \quad (3.7)$$

$R_i$  (resp.  $R_l$ ) est l'ensemble des machines qui sont utilisées lorsque le chemin  $i$  (resp.  $l$ ) est sollicité pour la fabrication d'un produit, et  $m$  est le nombre total de machines sollicités par les chemins  $i$  et  $l$ . L'auteur propose aussi une mesure de la variété de deux chemins  $d_{il}$  par la relation :

$$d_{il} = 1 - s_{il} \quad (3.8)$$

Plus le critère  $d_{il}$  est important moins la différence est importante entre deux chemins. De cette manière la variété de routage d'un produit est définit de la manière suivante :

$$D_j = \frac{1}{r(r-1)} \sum_{i=l}^r \sum_{i=1}^r d_{il} \quad (3.9)$$

Ensuite dans d'autres travaux, le même auteur définit un autre indicateur de la flexibilité appelé *routing variety* permettant de quantifier la différence entre les chemins alternatifs pour la fabrication d'une famille de pièces [94]. Le calcul de la flexibilité se fait à l'aide d'un modèle *CCR (Concurrent and Coordination Runtime Model)* [95] Ce modèle prend



en compte les coûts de production, et prend en entrée les données opératoires et renvoie en sortie la qualité et la quantité des chemins alternatifs possibles. [14] Propose une mesure entropique de la versatilité de la flexibilité de routage

En définitive en se servant des trois aspects de la flexibilité de routage à savoir l'efficacité, la versatilité et la variété cette dernière est définie par :

$$RF = \frac{1}{k} \sum_{j=1}^k E_j \times R_j \times D_j \quad (3.10)$$

Dans des travaux de [91], les trois aspects de la flexibilité sont développés en profondeur et l'auteur définit plusieurs autres indicateurs tels que : the Alternative Route Efficiency (ARE), Job Routing Average Efficiency (JRAE), Job Routing Range (JRR), Job Routing Versatility (JRV), Global Routing Efficiency (GRE), Global Routing Range( GRR), Global Routing Versatility (GRV) and Global Routing Flexibility (GRF).

Le JRAE (Job Routing Average Efficiency) noté  $ARE_{rp}$  est l'efficacité du chemin alternatif  $r$  pour la production d'un produit  $p$  par rapport au chemin standard en terme de coût. Il est donné par la relation suivante :

$$ARE_{rp} = \frac{RC_{0p}}{RC_{rp}} \quad (3.11)$$

avec  $RC_{0p}$  et  $RC_{rp}$  respectivement le coût du chemin standard et le coût du chemin alternatif  $r$

Le JRAE (Job Routing Average Efficiency) est défini lorsque un produit a plusieurs chemins alternatifs  $\tilde{R}_p$  et est donné par la relation :

$$JRAE_p = \frac{\sum_{r=1}^{\tilde{R}_p} ARE_{rp}}{\tilde{R}_p} \quad (3.12)$$

L'auteur rappelle que même si l'utilisation de cet indicateur permet un calcul rapide, il occulte la notion de versatilité des chemins, En effet celui-ci ne permet pas de différencier un produit qui a un seul chemin alternatif de celui qui en a plus d'un. Il présente entre autre dans son travail des exemples où il en fait la démonstration. C'est pour cette raison qu'il propose un autre indicateur qui est le JRR( job Routing Range) défini de la manière suivante :

$$JRR_p(\tilde{R}_p) = \begin{cases} 1 - (1 - \frac{\tilde{R}_p}{TR-1})^{\alpha_p} & \text{si } TR \geq 2 \\ 0 & \text{sinon} \end{cases} \quad (3.13)$$

$TR$  est le nombre total de route du système de production. et  $\alpha_p$  un paramètre de forme. La détermination du paramètre de forme fait l'objet d'une étude approfondie que nous ne présenterons pas dans notre travail.

Dans ces mêmes travaux de [91], on retrouve aussi une définition de la versatilité de routage d'un produit telle que présentée par [94] mais cette fois ci l'auteur l'introduit

par la notion de Job Routing versatility à travers un couple de vecteurs de telle sorte la versatilité de routage d'un produit  $JRV_p$  soit défini de la manière suivante :

$$JRV_p = \langle E_p, R_p \rangle \quad (3.14)$$

où  $E_p$  est le vecteur efficacité de routage d'un produit  $p$  et  $R_p$  un vecteur caractérisant la gamme de production de ce même produit. Ces deux vecteurs sont définis par :

$$E_p = [(ARE_{rp})_1, (ARE_{rp})_2, \dots, (ARE_{rp})_i, \dots, (ARE_{rp})_{\tilde{R}_p}, 0_{R_p}, \dots, 0_{TR-1}] \quad (3.15)$$

Dans ce travail l'auteur présente plusieurs autres indicateurs de la flexibilité de routage ainsi que la notion de qualité de routage. L'une des remarques que nous pouvons faire sur la notion de Flexibilité de routage c'est que celle-ci dépend uniquement des produits et des machines et de la nature physique ou encore de coûts qui sont connus à l'avance. En définitive, même si les indicateurs de mesures de flexibilité du système de production peuvent varier en fonction des auteurs ; on relève quand même la constance des trois caractéristiques de la flexibilité de routage à savoir la versatilité, la variété et l'efficacité. Nous abordons la notion de flexibilité de routage parce que cette dernière est fortement liée à la stratégie de gestion du transport. Nous allons proposer dans le prochain paragraphe un outil qui sera intégré à nos modèles de réseaux de Pétri afin d'observer la variation de la flexibilité de routage de notre système au cours d'une phase de production. Dans les prochaines lignes nous en expliquerons la méthodologie.

## Méthodologie d'évaluation de la Flexibilité de Routage

Pour mesurer la flexibilité d'acheminement des produits, nous allons calculer l'entropie de la matrice issue de la table d'évaluation des opérations de transports possibles. En effet nous avons observé que le nombre de lignes de la table d'évaluation des activités de transport permettait de déterminer le nombre de chemins possibles pour effectuer une opération de transport d'un produit, mais aussi le nombre de choix du(des) transporteur(s). En plus de cela le calcul de l'entropie suivant un critère nous permet de comparer deux critères en terme de facilité de prise de décision. C'est pour cela que nous avons décidé de calculer l'entropie de la matrice d'évaluation des opérations de transport. Dans ce paragraphe, nous utiliserons des modèles basés sur les réseaux de Pétri et la simulation pour calculer à chaque prise de décision la valeur de l'entropie de la table d'évaluation. Cette méthodologie comprend trois étapes :

- Le développement d'un modèle de simulation d'un atelier flexible en utilisant les RdPCH
- La définition des critères d'évaluation pour la table de décision du transport, en construisant la matrice de décision développée dans les paragraphes précédents
- Évaluation de la flexibilité en fonction du temps à l'aide de la simulation.

En ce qui concerne le modèle basé sur les réseaux de Pétri nous allons utiliser ceux que nous avons développés dans le chapitre 2. Nous allons donc uniquement présenter le calcul

de l'entropie de la matrice issue de la table d'évaluation des opérations de transport. Le calcul de cet indicateur passe par une série d'étapes à savoir :

1. La normalisation de la matrice d'évaluation : Soit  $A$  une matrice d'évaluation des activités de transport,  $A$  contient  $I$  lignes et  $J$  colonnes. Alors si  $AN$  est sa matrice normalisée, on a :

$$AN_{ij} = \frac{A_{ij}}{\sum_i^I A_{ij}} \quad (3.16)$$

tel que :  $I$  est le nombre d'opérations de transport possible à un instant donné (le nombre de lignes de la matrice  $A$ ). La matrice normalisée des activités de transport est une matrice dont les éléments  $AN_{ij} \in [0, 1]$  et :

$$\sum_i^I AN_{ij} = 1$$

En normalisant les éléments de la matrice suivant chaque critère on peut ainsi voir le poids d'une activité de transport par rapport à une autre suivant un critère donné.

2. Calcul de l'entropie de l'activité de transport  $i$  suivant un critère d'évaluation  $j$  : afin d'observer la dispersion des valeurs d'un critère donné, nous avons proposé de calculer l'entropie de chaque valeurs comme suit :

$$h_{ij} = -AN_{ij} \times \ln(AN_{ij}) \quad (3.17)$$

3. Calcul de l'entropie suivant chaque critère d'évaluation :

$$H_j = \sum_{i=1}^I h_{ij} \quad (3.18)$$

La valeur de l'entropie suivant un critère  $H_j$  montre le niveau de difficulté dans le choix d'une activité de transport suivant un critère  $j$ . Si  $H_j < H_k$  alors il sera plus facile de choisir l'activité de transport suivant le critère  $i$  que suivant le critère  $j$  parce que les valeurs du critère d'évaluation selon le critère  $j$  sont plus rapprochées que celles suivant le critère  $i$ .

4. Calcul de l'entropie de la matrice d'évaluation des activités de transport :

$$H = \sum_{j=1}^J H_j \quad (3.19)$$

A présent nous allons chercher un encadrement de l'entropie de la matrice que nous venons de définir. Pour cela nous avons besoin de rappeler la convexité de la fonction  $\ln$ . D'autres part prenons la fonction d'entropie

$$\begin{aligned} h : ]0, 1] &\rightarrow \mathbb{R} \\ x &\mapsto -x \ln x \end{aligned}$$

le tableau de variation de la fonction entropie  $h$  est :

$x$	0	$\frac{1}{e}$	1
$h'(x)$		+	-
$h'(x)$		$\frac{1}{e}$	
	0		0

La fonction  $h$  a un maximum de  $\frac{1}{e}$  ; alors cela signifie que :

$$\forall x \in ]0, 1] \text{ alors : } 0 \leq -x \ln(x) \leq \frac{1}{e}$$

La fonction  $\ln$  est une fonction convexe donc on peut appliquer l'inégalité de Jensen alors :

$$\ln\left(\sum_{i=1}^n x_i^2\right) \leq \sum_{i=1}^n x_i \ln x_i \text{ avec } \sum_{i=1}^n x_i = 1$$

En multipliant par -1 et en se servant de l'encadrement de la fonction  $h$  on a :

$$0 \leq -\sum_{i=1}^n x_i \ln x_i \leq -\ln\left(\sum_{i=1}^n x_i^2\right) \text{ avec } \sum_{i=1}^n x_i = 1$$

avec cette relation on peut aisément encadrer l'entropie suivant un critère  $H_j$  de la manière suivante :

$$0 \leq H_j \leq -\ln\left(\sum_{i=1}^I AN_{ij}^2\right)$$

Alors l'encadrement de l'entropie de la matrice sera :

$$0 \leq H \leq -\sum_{j=1}^J \left(\ln\left(\sum_{i=1}^I AN_{ij}^2\right)\right)$$

En se servant du maximum de la fonction d'entropie on peut trouver un maximum à l'entropie de la matrice  $H$ . Comme

$$\forall x \in ]0, 1] \text{ alors : } 0 \leq -x \ln(x) \leq \frac{1}{e}$$

alors,

$$h_{ij} = -AN_{ij} \times \ln(AN_{ij}) \leq \frac{1}{e} \text{ alors ,}$$

$$H_j = \sum_{i=1}^I h_{ij} \leq \frac{I}{e} \text{ et ,}$$

$$H = \sum_{j=1}^J H_j \leq \frac{IJ}{e}$$

en conclusion,

$$0 \leq H \leq \frac{IJ}{e}$$

Cependant il faut aussi noter que la valeur de l'entropie est maximale si tous les  $AN_{ij}$  sont égaux c'est à dire que :

$$AN_{ij} = \frac{1}{I}$$

La valeur maximale correspond dans ce cas à :

$$H_{max} = J \ln(I)$$

Par simple étude de la fonction  $x \mapsto x - e \ln(x)$  on peut très facilement vérifier que

$$J \ln(I) < \frac{IJ}{e}$$

Une des remarques que l'on puisse faire sur l'indicateur de flexibilité que nous proposons est que ce dernier ne dépend pas du classement de la position des colonnes des critères les unes par rapport aux autres. Cependant ce que nous voulons mesurer ici c'est l'influence que peut avoir les décisions prises par le transport sur la flexibilité moyenne du système au cours d'une phase de production. En effet même si la valeur de l'entropie d'une table ne change pas, la décision qui est prise en ce qui concerne le transport aura très certainement une influence sur l'entropie de la table d'évaluation à aux instants de décisions suivants.

Nous allons utiliser cet indicateur de l'entropie pour mesurer la flexibilité globale d'un système de production au cours d'une phase de production. Cela va nous permettre de mesurer l'impact des décisions sur la mesure de la flexibilité de l'atelier de production.

## Présentation de la stratégie de décision

Dans la section précédente nous présentions une table d'évaluation instantanée de l'activité de transport d'un atelier de production. Dans ce paragraphe, nous allons nous intéresser à la sélection des meilleures actions potentielles. Pour cela nous proposons deux algorithmes qui vont nous permettre de décider quelle opération de transport sera la meilleure par rapport aux autres. Nous allons proposer dans un premier temps un algorithme de gestion statique et dans un deuxième temps un algorithme dynamique

Les deux algorithmes que nous présentons reposent sur un certain nombre de considérations.

- Dans la table d'évaluation les critères sont rangés suivant un ordre décroissant d'importance et de la gauche vers la droite.

- Pour trouver la meilleure solution on fera une filtration des opérations en partant du critère le plus important au moins important et on ne s'arrêtera que si après une filtration, il ne reste plus qu'une opération de transport réalisables. L'opération de filtrage consistera à déterminer, pour un critère  $i$  concerne l'ensemble des opérations qui surclassent toutes les autres. Ainsi à l'issue de chaque filtration, on éliminera de la table les opérations de transport qui sont surclassées par d'autres.

La stratégie que nous proposons accorde une importance à la position des critères dans la table d'évaluation. Comment choisir la position de ces deniers au cours de la production. Pour répondre à cela nous allons présenter deux types de stratégies :

- La première consiste à définir de manière figée le rangement des critères dans la table d'évaluation. c'est que nous allons présenter avec l'algorithme statique.
- La deuxième méthode consiste à changer l'ordre de rangement des critères lors de la production en fonction des caractéristiques du système. C'est cette stratégie que nous allons présenter à travers l'algorithme dynamique.

Nous avons décidé de proposer les algorithmes parce que tout dépend dans un système de production de la stratégie de management et parfois imposer des règles d'ordonnancement peut être voulu par les décideurs. Déterminer toutes les opérations de transport susceptibles d'être réalisées.

Pour la présentation des deux algorithmes , nous adopterons les notations suivantes :

- M : La table d'évaluation des activités de transport ;
- L : le nombre de lignes de M ;
- J le nombre de critères retenus.

## L'algorithme de gestion statique du transport

Nous allons présenter dans cette section un algorithme statique pour la gestion du transport dans une cellule de production d'un atelier de production flexible. Nous nous servirons de la table d'évaluation de l'activité de transport pour le pilotage des opérations de transport. L'objectif de cet algorithme est de déterminer une opération de transport à chaque instant de décision. Nous allons nous servir des critères d'évaluation pour sélectionner une opération de transport. La figure 3.2 présente un algorithme simplifié de cet algorithme.

Nous présentons dans les prochains paragraphes les données et entrée et en sortie de l'algorithme de gestion statique. Les données d'entrée de l'algorithme sont :

- M : table d'évaluation de l'activité de transport
- L Nombre de ligne de la table d'évaluation de M

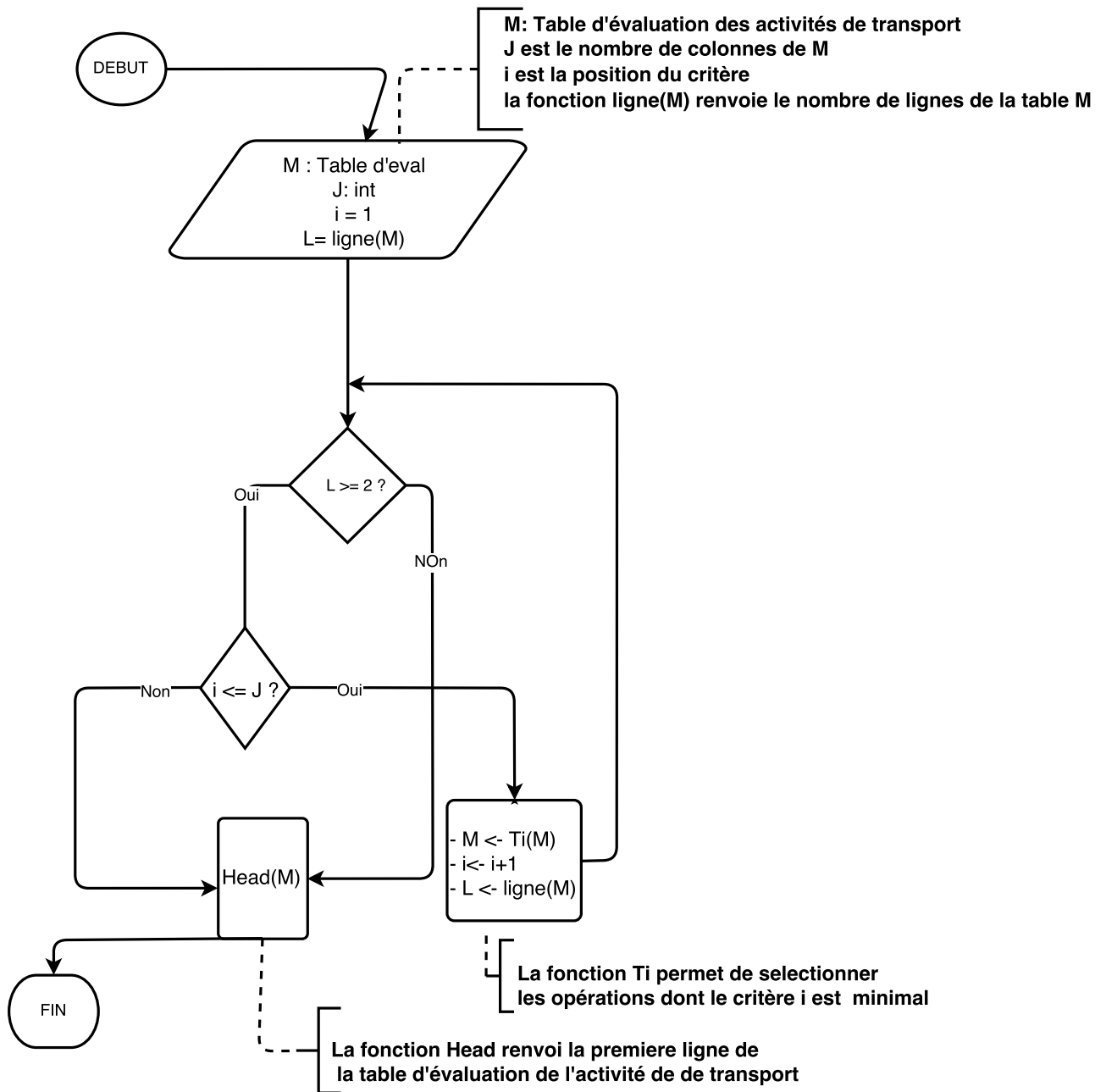


FIGURE 3.2 – Algorithme de gestion statique du transport

- J est le nombre de colonnes de M
- i est un compteur de colonne. De cette manière  $i$  représente la colonne  $i$  donc d'une certaine façon le critère de la colonne  $i$ .

En sortie de l'algorithme on ne doit avoir qu'une seule opération de transport.

Les fonctions de l'algorithme sont :

- $Ligne(M)$  : est une fonction qui renvoie le nombre de lignes d'une table d'évaluation des activités de transport.
- $Ti(M)$  est une fonction qui permet de faire le tri et la sélection des opérations de transport suivant le critère de colonnes  $i$  de manière à ne retenir que les opérations pour laquelle le critère de la colonne  $i$  est minimal.
- $Head(M)$  est la fonction qui sélectionne la première ligne de la table d'évaluation  $M$ . Cette fonction sera surtout utilisé si à la fin des phases de tri et de sélection on n'a pas pu retenir une seule opération de transports.

### l'exécution de l'algorithme

L'algorithme est exécuté lorsque la table d'évaluation est générée à un instant de décision  $t$ . A ce moment là si le nombre de lignes de la tables d'évaluation est égale à un, cela signifie qu'il n'y a qu'une seule opération de transport, il n' y a donc pas de choix à faire l'algorithme s'arrête.

Si par contre le nombre de lignes est supérieur ou égale à deux, on sélectionne les opérations de transport pour lesquelles le critères d'évaluation de la première colonne est minimale. Si à l'issue de cette sélection le nombre d'opérations de transport est supérieur à un, on refait la même sélection mais en utilisant le critère de la colonne deux. Tant que l'on obtient pas une seule opération on refera le processus de sélection continu en se basant sur les critères restants. Si on a effectué la sélection suivant tous les critères et qu'il reste toujours plus d'une opération on prendra l'opération qui sera sur la première ligne de la table restante. Une traduction de l'algorithme que nous venons de décrire est présentée par l'algorithme ci-dessous.

```

 $M \leftarrow [O_1, O_2, O_3, \dots, O_M]$            ▷ Liste des opérations de transport possibles
 $i \leftarrow J$ 
 $J \leftarrow 1$ 
 $M \leftarrow Ti(M)$        ▷ Liste des opérations de transport dont le critère de la colonne  $j$  est
minimal;
while  $L > 1$  do
   $i \leftarrow i + 1$ 
   $M \leftarrow Ti(M)$ 
   $L \leftarrow ligne(M)$ 
end
 $Op \leftarrow Head(M)$    ▷  $hd$  est une fonction qui renvoie le premier élément d'une liste  $L$ 
    
```

#### Algorithm 1: Algorithme de gestion du transport

Pour illustrer notre algorithme, prenons l'exemple d'une table d'évaluation qui contient 5 opérations et 3 critères d'évaluation (confère table 3.2).

A partir de cette table courante nous allons effectuer un tri suivant le critère de la première colonne. La règle du tri sera de conserver dans la table (les)opération(s) dont le



Opération	Tr	Pr	ZC	ZD	$critère_1$	$critère_2$	$critère_3$
1	$Tr_{\sigma(1)}$	$Pr_{\lambda(1)}$	$ZC_{\mu(1)}$	$ZD_{\phi(1)}$	3	5	7
2	$Tr_{\sigma(2)}$	$Pr_{\lambda(2)}$	$ZC_{\mu(2)}$	$ZD_{\phi(2)}$	2	4	5
3	$Tr_{\sigma(3)}$	$Pr_{\lambda(3)}$	$ZC_{\mu(3)}$	$ZD_{\phi(3)}$	4	10	5
4	$Tr_{\sigma(4)}$	$Pr_{\lambda(4)}$	$ZC_{\mu(4)}$	$ZD_{\phi(4)}$	2	4	4
5	$Tr_{\sigma(5)}$	$Pr_{\lambda(5)}$	$ZC_{\mu(5)}$	$ZD_{\phi(5)}$	9	9	2

TABLE 3.2 – Table d'évaluation initiale

critère de la première colonne est(ont) le (les) plus faible(s). A la fin de ce tri on évalue le nombre d'opérations qui restent dans la table, si ce dernier est supérieur à 1 alors on effectue sur la nouvelle table le même tri suivant le critère de la colonne suivante. Tant que le nombre d'opérations dans la table est supérieur à 1 nous continuons le tri suivant les colonnes suivantes, en cherchant à chaque fois à ne garder dans la table que les opérations de transport qui ont le critère minimal suivant la dite colonne. Si après avoir effectué le tri suivant toutes les colonnes de la table d'évaluation, il reste toujours plus d'une opérations dans la table alors à ce moment là nous prendrons par défaut la première opération de la table finale obtenue après le tri suivant la dernière colonne.

Pour illustrer cela prenons un premier exemple d'une petite instance représentée par la table d'évaluation 3.2. Si l'on applique un tri suivant le critère de la colonne 1 à la table 3.2 on obtient la table 3.3

Opération	Tr	Pr	ZC	ZD	$critère_1$	$critère_2$	$critère_3$
2	$Tr_{\sigma(2)}$	$Pr_{\lambda(2)}$	$ZC_{\mu(2)}$	$ZD_{\phi(2)}$	2	4	5
4	$Tr_{\sigma(4)}$	$Pr_{\lambda(4)}$	$ZC_{\mu(4)}$	$ZD_{\phi(4)}$	2	4	4

TABLE 3.3 – Etape 1 : Table obtenue après avoir effectué un tri sélectif sur la suivant le critère de la première colonne de la table 3.2

La table 3.3 contient deux lignes ,alors dans ce cas on effectue un tri suivant le critère de la deuxième colonne afin de déterminer l'opération dont le critère 2 est minimal. Dans le cas de figure on observe que le tri suivant le critère de la colonne 2 ne permettra pas de réduire le nombre d'opérations alors nous allons passer au critère de la colonne 3. En appliquant le tri suivant la colonne 3 la table devient alors :

Opération	Tr	Pr	ZC	ZD	$critère_1$	$critère_2$	$critère_3$
4	$Tr_{\sigma(4)}$	$Pr_{\lambda(4)}$	$ZC_{\mu(4)}$	$ZD_{\phi(4)}$	2	4	4

TABLE 3.4 – Etape 3 de l'algorithme : tri de la table d'évaluation courante suivant le critère de la troisième colonne

On observe que la table 3.4 obtenue après le tri sélectif sur la troisième colonne ne contient qu'une seule ligne dans ce cas l'algorithme s'arrête et l'opération de transport qui sera effectuée sera l'opération<sub>4,+</sub>

## Analyse de l'algorithme statique de gestion du transport

L'algorithme de gestion statique du transport permet de résoudre les conflits qui surviennent lorsque le nombre de ressources de transport est inférieur au nombre d'opérations de transport à effectuer. L'un des objectifs de sa mise en place est de permettre :

- D'assurer le contrôle de la simulation afin d'éviter le blocage du modèle au cours de celle-ci ;
- De définir des règles de dispatching

Cependant il est légitime de s'interroger sur l'impact d'un tel algorithme sur la gestion de l'activité de transport dans un atelier de production et bien évidemment sur les performances globales du système au cours d'une phase de production. On observe que le résultat de cet algorithme est lié à l'ordre qu'occupe chaque critère sur les colonnes de la table d'évaluation, car l'inversion des colonnes peut provoquer un résultat différent. En effet si pour la table 3.2 on inversait les colonnes 1 et 3 alors c'est l'*Operation*<sub>5</sub> qui aurait été choisie par l'algorithme et non l'*Operation*<sub>4</sub>. Cela signifie que l'utilisation d'un tel algorithme impose au gestionnaire de production de déterminer avec précaution l'ordre des critères au niveau de la table d'évaluation de l'activité de transport.

Le choix de la position d'un critère sur une colonne révèle donc le caractère prioritaire de ce dernier par rapport aux autres. Cependant d'une situation de décision à une autre au cours de la production, un critère pourtant situé à une position lui conférant une importance inférieure à un autre peut se révéler en réalité plus important. Supposons une table d'évaluation définie sur deux critères d'évaluation, la durée du transport pour le premier, et la taille des zones de déchargement pour le deuxième. Si l'on considère que la durée de transport est un critère prioritaire par rapport à celui de la charge des stocks en sortie des machines cela pourra créer à un moment donné le blocage d'une machine qui sera saturée de produits parce que dans l'arbitrage par rapport au premier critère (durée de transport) elle ne sera jamais choisie par la ressource de transport. Cela signifie qu'il faut définir l'importance des critères de manière à ce que celle-ci suive l'évolution du système au cours de la production. Alors, la capacité des zones de déchargement devient prioritaire par rapport au temps de transport. Ce type de situation pose donc un problème quand à la priorisation des critères d'évaluation en fonction de l'état du système de production. Pour tenter de répondre à ce problème, nous avons proposé un algorithme nous permettrait de manière dynamique de classer nos critères en fonction de l'instant de décision.

## Algorithme de gestion dynamique de l'activité de transport

L'algorithme de gestion dynamique de l'activité de transport a pour objectif de déterminer l'ordre d'importance des critères de décision.. Pour le faire nous avons choisi de calculer l'entropie de chaque critère de la table d'évaluation de transport. L'étude de l'entropie de chaque critère permet de définir le niveau de difficulté que l'on a se décider rapport à ce dernier si l'on souhaite choisir une opération de transport parmi d'autres en se basant sur ce critère. Nous allons donc reclasser les colonnes de critère de la table

d'évaluation en fonction de la valeur de l'entropie. Pour calculer et comparer les entropies de chaque critère de décision, nous allons dans un premier temps définir une table normalisée de la table d'évaluation des activités de transport. La table 3.5 est la table normalisée des activités de transport.

Opération	Tr	Pr	ZC	ZD	<i>critère</i> <sub>1</sub>	...	<i>critère</i> <sub><i>j</i></sub>	...	<i>critère</i> <sub><i>J</i></sub>
1	$Tr_{\sigma(1)}$	$Pr_{\lambda(1)}$	$ZC_{\mu(1)}$	$ZD_{\phi(1)}$	$N_{11}$	...	$N_{1j}$	...	$N_{1J}$
2	$Tr_{\sigma(2)}$	$Pr_{\lambda(2)}$	$ZC_{\mu(2)}$	$ZD_{\phi(2)}$	$N_{21}$	...	$N_{2j}$	...	$N_{2J}$
3	$Tr_{\sigma(3)}$	$Pr_{\lambda(3)}$	$ZC_{\mu(3)}$	$ZD_{\phi(3)}$	$N_{31}$	...	$N_{3j}$	...	$N_{3J}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
<i>i</i>	$Tr_{\sigma(i)}$	$Pr_{\lambda(i)}$	$ZC_{\mu(i)}$	$ZD_{\phi(i)}$	$N_{i1}$	...	$N_{ij}$	...	$N_{iJ}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
<i>I</i>	$Tr_{\sigma(I)}$	$Pr_{\lambda(I)}$	$ZC_{\mu(I)}$	$ZD_{\phi(I)}$	$N_{I1}$	...	$N_{Ij}$	...	$N_{IJ}$

TABLE 3.5 – Table d'évaluation normalisée de l'activité de transport

La table normalisée a exactement la même structure que la table d'évaluation du transport à la seule exception que les valeurs des critères  $A_{ij}$  sont remplacés par les valeurs normalisés  $N_{ij}$  avec :

$$N_{ij} = \frac{A_{ij}}{\sum_i^I A_{ij}} \quad (3.20)$$

La normalisation de la table d'évaluation des activités de transport va nous permettre de pouvoir comparer deux critères qui n'ont pas la même unité. La valeur  $N_{ij}$  représente le poids que l'opération de transport  $i$  par rapport aux autres opérations de transport, et ce suivant le critère de la  $j^{\text{ième}}$  colonne. La définition de la table normalisée va nous permettre de déterminer la table entropique de l'évaluation des activités de transport

Pour cela, il suffit de remplacer dans la table normalisée les coefficients  $N_{ij}$  par :

$$H_{ij} = -N_{ij} \times \ln N_{ij}$$

De cette manière on obtient la table 3.5. Nous allons nous servir à présent de la table entropique pour calculer l'entropie de chaque critère. L'entropie  $H_j$  d'un critère  $j$  sera donnée par la relation :

$$H_j = \sum_{i=1}^I H_{ij} \quad (3.21)$$

$H_j$  permet de mesurer la dispersion des valeurs du critère  $j$  des opérations de transport. Plus l'entropie d'une colonne est importante, plus les valeurs des opérations de transport suivants ce critère sont dispersées et plus il est sera difficile de faire un choix par rapport à ce critère. Inversement plus cette entropie suivant un critère est faible plus il sera facile de se décider par rapport à ce critère là. Il faut remarquer que la valeur de l'entropie de chaque critère dépendra de l'instant de décision, par conséquent de l'évolution du système.

Opération	Tr	Pr	ZC	ZD	$critère_1$	...	$critère_j$	...	$critère_J$
1	$Tr_{\sigma(1)}$	$Pr_{\lambda(1)}$	$ZC_{\mu(1)}$	$ZD_{\phi(1)}$	$H_{11}$	...	$H_{1j}$	...	$H_{1J}$
2	$Tr_{\sigma(2)}$	$Pr_{\lambda(2)}$	$ZC_{\mu(2)}$	$ZD_{\phi(2)}$	$H_{21}$	...	$H_{2j}$	...	$H_{2J}$
3	$Tr_{\sigma(3)}$	$Pr_{\lambda(3)}$	$ZC_{\mu(3)}$	$ZD_{\phi(3)}$	$H_{31}$	...	$H_{3j}$	...	$H_{3J}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
$i$	$Tr_{\sigma(i)}$	$Pr_{\lambda(i)}$	$ZC_{\mu(i)}$	$ZD_{\phi(i)}$	$H_{i1}$	...	$H_{ij}$	...	$H_{iJ}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
$I$	$Tr_{\sigma(I)}$	$Pr_{\lambda(I)}$	$ZC_{\mu(I)}$	$ZD_{\phi(I)}$	$H_{I1}$	...	$H_{Ij}$	...	$H_{IJ}$

TABLE 3.6 – Table d'évaluation normalisée de l'activité de transport

Pour l'algorithme dynamique que nous proposons, nous allons exploiter la table des entropies afin de reclasser les colonnes de la table d'évaluation des activités de transport. Pour classer les colonnes de la table d'évaluation des activités de transport nous allons nous servir de la table des entropies 3.6. Nous allons calculer l'entropie sur chaque colonne par la formule :

Nous allons classer les critères suivant l'ordre des entropies maximales. De cette manière le critère ayant l'entropie la plus élevée sera en première colonne. Et celui qui possède l'entropie minimale sera positionné sur la dernière colonne. Une fois que la table sera reclassée la détermination de la meilleure opération de transport se fera en utilisant l'algorithme 1. Nous allons nous servir de la table d'évaluation des entropies pour écrire le deuxième algorithme de gestion du transport.

La figure 3.3 est un algorithme de l'algorithme de gestion dynamique du transport.

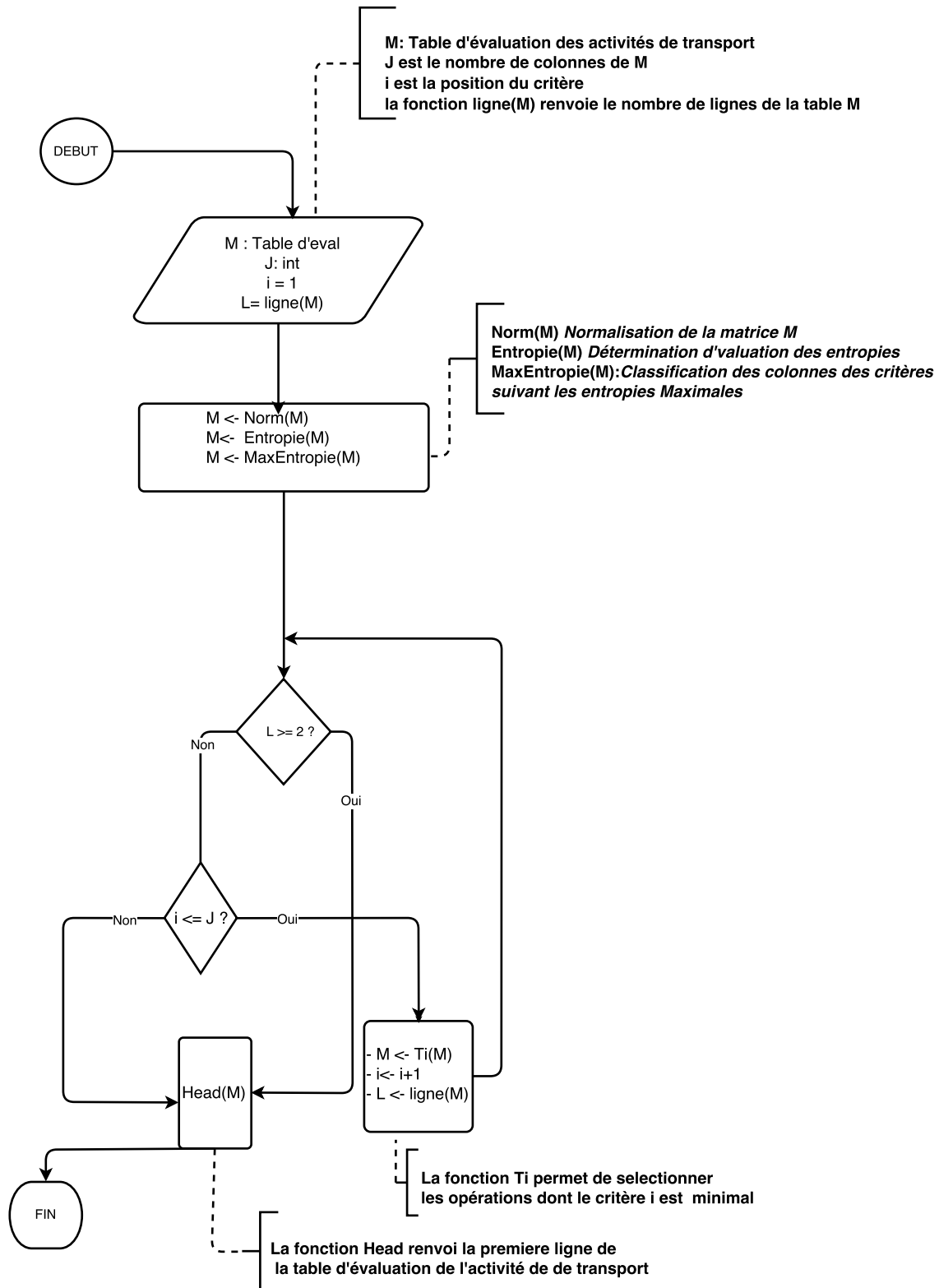


FIGURE 3.3 – Algorithme de gestion dynamique des opérations de transport

Nous allons définir les données en entrée et en sortie de l’algorithme de gestion même s’il sont identiques à celui de l’atelier statique.

Les données d’entrée sont

M : table d'évaluation de l'activité de transport

L Nombre de ligne de la table d'évaluation de M

J est le nombre de colonnes de M

$i$  est un compteur de colonne. De cette manière  $i$  représente la colonne  $i$  donc d'une certaine façon le critère de la colonne  $i$ . La donnée de sortie est : l'opération de transport qui sera effectuée. Les Fonctions définies dans l'algorithme sont :  $Norm(M)$  est une fonction qui normalise les coefficients de la table d'évaluation tel que nous l'avons défini dans la relation 3.20.  $MaxEntropie(M)$  est une fonction qui classe les colonnes de la table d'évaluation suivant les entropies maximales. Ainsi la colonne possédant la plus grande entropie sera en première position et celle qui a la plus faible entropie en dernière position.  $Entropie(M)$  est une fonction qui permet de déterminer la table des entropies proposée dans la table 3.6

### Excécution de l'algorithme

L'algorithme de gestion dynamique débute lorsque l'on a une table d'évaluation des activités de transport. Si le nombre de ligne est égale à un il n'y a qu'un choix possible. Dans le cas contraire, on normalise la table d'évaluation, ensuite on la transforme en table des entropies et enfin on classe les colonnes suivant les entropies maximales. Après ces trois opérations on applique l'algorithme statique à la table obtenue.

```

 $O_H \leftarrow [O_{1H}, O_{2H}, O_{3H}, \dots, O_{IH}]$     ▷ Liste des opérations de transport possibles de la
table d'évaluation entropique
 $j \leftarrow 1$ 
 $L \leftarrow \min_{C_j} \{O_{iH}\}$     ▷ Liste des activités de transport dont le critère de type  $j$  est
minimal;
while  $Length(L) > 1$  do
|    $j \leftarrow j + 1$ 
|    $L \leftarrow \min_{C_j} \{L\}$ 
|   end
 $path \leftarrow hd(L)$     ▷  $hd$  renvoie le premier élément de la liste  $L$ 

```

**Algorithm 2:** Algorithme de gestion du transport

### Discussion sur l'algorithme de gestion dynamique

Nous avons proposé un algorithme de gestion dynamique pour tenter de répondre au cours d'une phase de production à l'évolution des contraintes internes à l'atelier. Le calcul de l'entropie permet de déterminer la dispersion des opérations de transport du point de vue d'un critère donné. Nous avons utilisé l'entropie pour reclasser les critères en vue d'utiliser l'algorithme statique que nous avons présenté à la section 3.5.1. Dans l'algorithme

dynamique que nous avons proposé nous avons classé les critères de la table d'évaluation suivant les entropies maximales. Nous avons expliqué au paragraphe précédent que plus l'entropie d'un critère était maximale plus il était facile de décider par rapport à ce critère. De cette manière après le reclassement des colonnes de la table d'évaluation, l'application de l'algorithme statique devrait à priori permettre de déterminer assez rapidement une opération de transport. Cependant, il faut tout de même noter que pour un bon fonctionnement de cet algorithme, il faudrait que d'un instant de décision à l'autre, que les critères des opérations évoluent sinon on peut se retrouver d'un instant de décision à l'autre avec les mêmes entropies ce qui signifierait que le classement des colonnes ne changerait pas.

Nous avons présenter dans cette section nos stratégies de gestion du transport, ces dernières seront intégrées dans les modèles que nous avons présentés dans les sections 2.3 et 2.4. Dans la prochaine section, nous allons faire une étude comparative des algorithmes statique et dynamique à l'aide d'une étude de cas.

## Etude de cas : pilotage d'un JOB SHOP flexible à quatre machines

Dans cette section, nous allons étudier les performances des algorithmes statiques et dynamiques de gestion du transport. Pour cela nous avons défini un cas d'étude d'un atelier de type JOB SHOP flexible constitué de quatre machines et d'une ressource de transport des produits entre les zones de chargement et de déchargement. La complexité de cet exemple permet de tester les limites des algorithmes proposés. Nous proposons aussi un algorithme pour mesurer la flexibilité de routage dans le dit atelier de production. Nous allons évaluer les performances de l'atelier de production à travers les indicateurs suivants :

- Le temps total d'écoulement des produits, le plus souvent appelé Makespan ;
- La distance parcourue par la ressource de transport ;
- La charge des stocks des machines : c'est la valeur moyenne du nombre de produit dans l'ensemble des stocks au cours de la production.
- La flexibilité de routage moyenne.

### Présentation du cas d'étude

Soit un système de production constitué de quatre machines  $M_1 M_2 M_3 M_4$ . Chaque machine possède au moins un outil . Soit  $T = \{A, B, C, D, E, F, G\}$  l'ensemble des outils disponibles au niveau des machines. Dans cette section nous présentons les caractéristiques de l'atelier de production à savoir :

- Sa configuration ou organisation spatiale ;

- Les caractéristiques de machines
- La description de la gamme de production ;
- La description de la production
- La description des critères d'évaluation des activités de transport.

### Organisation spatiale de l'atelier de production

L'atelier a une surface totale délimitée par un rectangle d'aire est de  $25m \times 35m$ . La figure 3.4 en illustre la configuration.

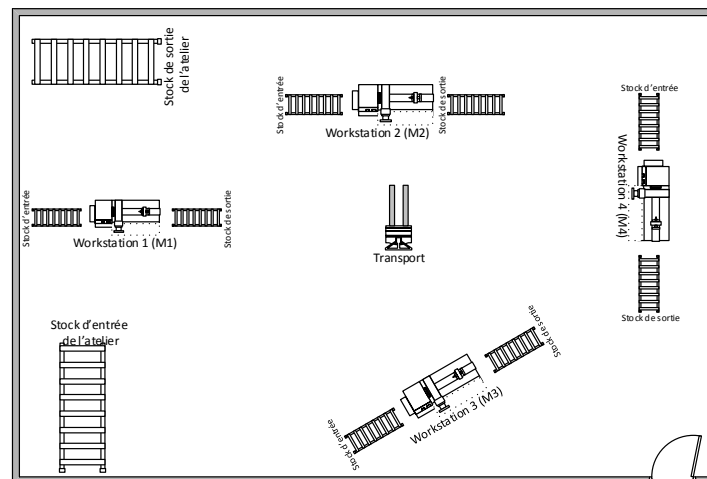


FIGURE 3.4 – Illustration de la configuration de l'atelier de production

Workstation 1, workstation 2, workstation 3 et workstation 4 représentent les quatre machines 1, 2, 3 et 4 respectivement. L'atelier possède des zones de stockage d'entrée et de sorties distinctes comme on peut le voir sur la Figure 3.4. Les positions des machines ainsi que les outils dont dispose chaque machine sont récapitulés dans la Table 3.7. Nous considérons que les stocks liés à une machine ont les mêmes coordonnées que celle-ci.

### Les caractéristiques des machines

Nous considérons que chaque opération de transformation réalisable par une machine à l'aide d'un outil  $X$  possédant un temps de changement fixe  $\sigma$ . L'atelier de production peut réaliser 7 opérations différentes appartenant à l'ensemble des outils de l'ateliers  $T = \{A, B, C, D, E, F, G\}$ . La table 3.7 donne pour chaque machine le nombre d'outils que celles-ci possède.



TABLE 3.7 – Position des machines dans l’atelier de production

Machines	Types d’outillage	Positions des Machines
$M_1$	A,B,E	(1.8 ;9.5)
$M_2$	C,D	(9.84 ;17.42)
$M_3$	D,G,B	(18.99 ;9.05)
$M_4$	E,F	(9.88 ;1.2)

Les temps de montage de chaque outil est donné par la table 3.8

TABLE 3.8 – temps de montage des outils

Outils	Temps de montage
$A$	2
$B$	3
$C$	2
$D$	1
$E$	3
$F$	3
$G$	2

### Gamme de production

L’atelier de production sert à la fabrication de trois types de produits :  $item_1$   $item_2$   $item_3$ . Le tableau 3.9 donne les différentes étapes de fabrication de chaque produit :

TABLE 3.9 – plan de production des produits

$item_1$	IN $\rightarrow$ F $\rightarrow$ E $\rightarrow$ D $\rightarrow$ C $\rightarrow$ OUT
$item_2$	IN $\rightarrow$ F $\rightarrow$ B $\rightarrow$ E $\rightarrow$ C $\rightarrow$ D $\rightarrow$ OUT
$item_3$	IN $\rightarrow$ B $\rightarrow$ F $\rightarrow$ E $\rightarrow$ C $\rightarrow$ D $\rightarrow$ C $\rightarrow$ OUT

$A \rightarrow B$  signifie que l’opération A précède l’opération B. Les opérations IN et OUT sont respectivement les opérations d’entrée et de sortie d’un produit de l’atelier de production. Les opérations  $A, B, C, D, E, F$  sont les opérations effectuées par les outils  $A, B, C, D, E, F$ .

### Description de la production

L’atelier produit des lots 150 produits de type  $item_1$   $item_2$   $item_3$ . Les types de produits sont dans les mêmes proportions à savoir : 50 produits de chaque type. Les produits arrivent dans la file d’attente de l’atelier dans l’ordre suivant :

$$p1, p2, p3, p1, p2, p3, \dots p1, p2, p3$$

, où  $p_1, p_2, p_3$  sont les produits de type 1, 2 et 3. La livraison des produits se fait à la fin de la production ce qui oblige d'avoir une taille suffisante du stock de sortie (150 minimum).

### Critères d'évaluation des opérations de transport

Pour évaluer l'activité de transport nous avons choisi cinq critères :

- Le taux de remplissage des zones de déchargement, représenté par la variable  $TRD$  ;
- Le taux de remplissage des zones de chargement, représenté par la variable  $TRC$  ;
- La distance à parcourir par la ressource pour effectuer l'opération de transport représentée par la variable  $dist$  ;
- Le temps d'attente du produit dans la zone de chargement si il était pris en dernier représenté par la variable  $TempsAttenteStockSortie$  ;
- Le temps d'attente du produit dans la zone de déchargement avant d'être traité par la machine, représenté par la variable  $TempsAttenteStockEntree$ .

Il faut noter que le transporteur considéré dans le cas d'étude ne peut transporter qu'un produit à la fois.

### Modélisation de l'atelier de production

Le modèle de cet atelier de production est présenté en 3.5. Il a été développé en utilisant les modèles de machines, stocks et ressources de transport développés dans le chapitre 2, ainsi que le modèle de gestion de transport que nous avons présenté à la section 2.4.

Les transitions de substitutions  $M1, M2, M3$  et  $M4$  sont les réseaux de Pétri modulaires des machines  $M1, M2, M3$  et  $M4$ . Les places  $IN1, IN2, IN3$  et  $IN4$  représentent les stocks en entrée de ces machines et les places  $OUT1, OUT2, OUT3$  et  $OUT4$  celles des stocks en sortie des mêmes machines. Les places  $InitM1, InitM2, InitM3, InitM4$  permettent d'initialiser les paramètres des machines avant le début de la simulation. Les places  $Entree$  et  $END$  représentent respectivement le stock en entrée et en sortie de l'atelier de production.

La transition de substitution  $MDT$  permet de modéliser la prise de décision lors des activités de transport et  $SCT$  permet de modéliser le déplacement effectif des produits. La place  $Mvt$  permet d'observer les ressources de transport qui sont en train de déplacer un produit. Elle permet aussi en outre de contenir le résultat de la prise de décision du problème du transport à savoir l'opération qui va être réalisée. La place  $TRANSPORT$  contient les ressources de transport disponibles. Ce modèle est implémenté sous CPN TOOLS et peut être simulé directement. Il est présenté à la figure 3.5.

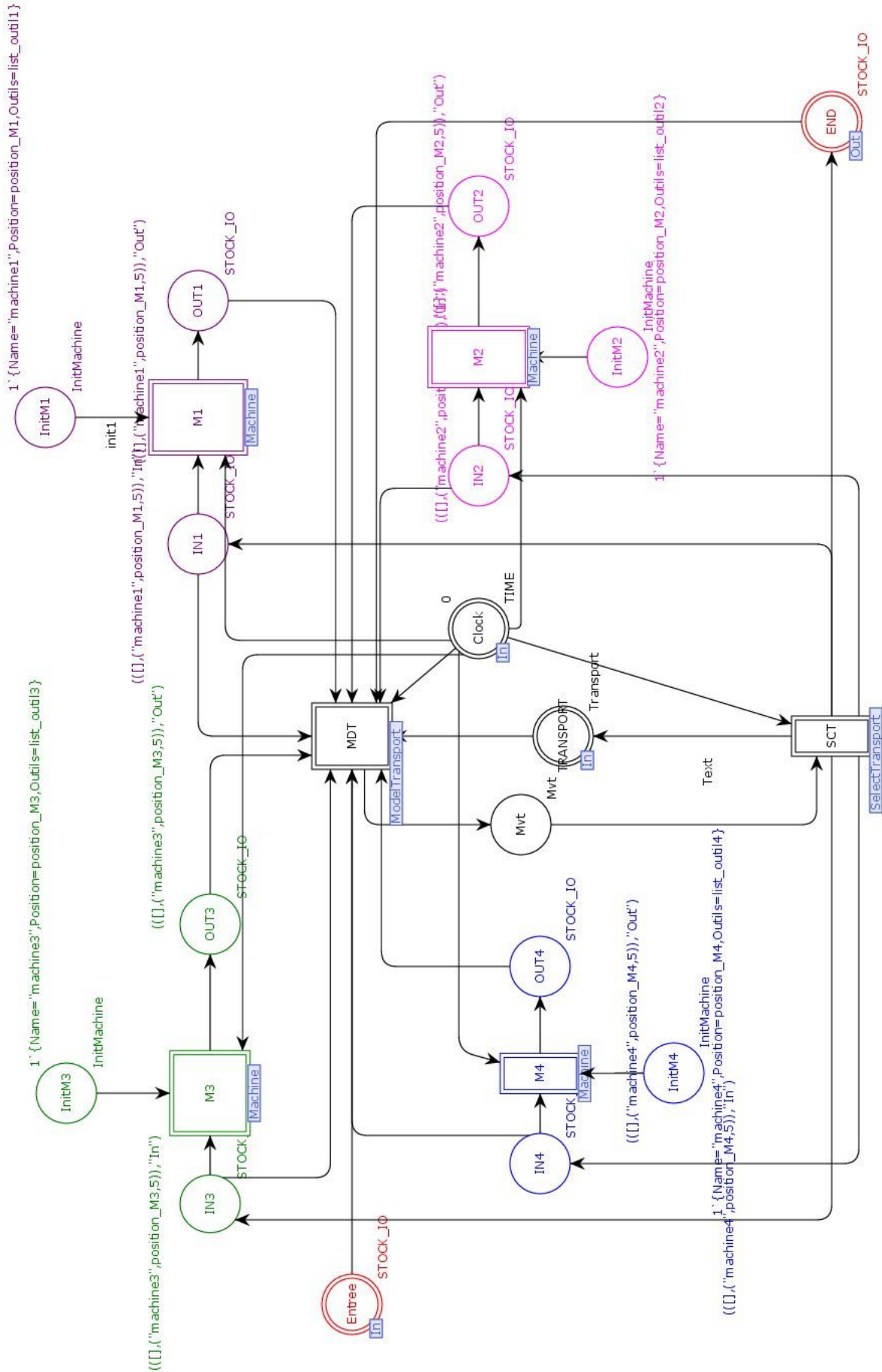


FIGURE 3.5 – Exemple d'un atelier à cheminement multiples à quatre machines

Nous allons présenter dans le prochain paragraphe les résultats que nous avons obtenus simulation.

## Simulations et résultats

Afin de : (1) tester l'algorithme d'ordonnement dynamique des tâches transport, (2) prouver son efficacité et (3) démontrer la pertinence de l'approche entropique pour la mesure de la flexibilité de routage, nous allons utiliser le simulateur du job shop flexible (présenté dans la Section 3.6) pour comparer notre approche à un type d'ordonnement statique déjà utilisé dans l'industrie. Afin de tester notre algorithme dans divers situation, nous avons proposé plusieurs types de système de production avec des niveau de flexibilité différent. Afin de changer le niveau de flexibilité dans le simulateur, nous avons fait varier les paramètres de simulation relatifs à la gestion des stocks de sortie des machines et à l'assignation des machines de production.

### Les paramètres du modèle de simulation

Dans la section 3.2.2 nous avons défini quatre paramètres influençant la gestion du transport dans un atelier par le quadruplet  $SGT = (\alpha|\beta|\gamma|\delta)$ . Pour la simulation nous allons jouer uniquement sur les paramètres  $(\alpha, \beta)$ . Ces paramètres permettent de renseigner d'une part les niveaux de flexibilité des produits et de l'autre part la stratégie d'ordonnement des stocks en sortie des machines. Le premier paramètre  $\alpha$  prendra deux valeurs : A pour signifier que toutes les opérations des produits ne s'effectue que sur une machine, et NA si elle peuvent s'effectuer sur une ou plusieurs machines, sous réserve que les machines possèdent les outils nécessaires. Le deuxième paramètre  $\beta$  lui aussi prendra deux valeurs : F signifiera que l'ordonnement de la file d'attente en sortie des machines suivra une règle de type FIFO (First In, First Out). Autrement dit le transporteur ne pourra pas choisir des produits dans le stock de sortie des machines et prendra toujours le produit en tête de la file d'attente. On notera NF, le cas où le transporteur pourra décider lui même du produit qui va transporter c'est à dire qu'il n'y aura pas de règle d'ordonnement pour la file d'attente des stocks en sortie des machines. En ce qui concerne les stocks en entrée des machines, la règles d'ordonnement est toujours FIFO. Dans la mesure où il n'y a qu'une ressource de transport, il n'y a pas de nécessité de définir une règle de gestion de la file d'attente des ressources de transport.

En considérant les variations des paramètres  $\alpha$  et  $\beta$  uniquement on aura quatre types de système de production possibles :

1. **La situation A-F** : le gestionnaire de l'atelier de production a choisi de manière délibérée d'affecter les opérations des produits aux machines avant le début de production. De cette manière on obtient la gamme de production de la Table 3.10 ci dessous.

TABLE 3.10 – Gamme de production

Étape de production	10	20	30	40	50	60
$item_1$	$M_4(2)$	$M_1(4)$	$M_2(2)$	$M_3(3)$		
$item_2$	$M_4(3)$	$M_1(2)$	$M_4(4)$	$M_2(3)$	$M_3(2)$	
$item_3$	$M_1(4)$	$M_4(3)$	$M_1(4)$	$M_2(2)$	$M_3(2)$	$M_2(5)$

On peut voir au travers cette table que le produit  $item_1$  doit nécessairement passer et de manière successive par les machines  $M_4$ ,  $M_1$ ,  $M_2$  et  $M_3$ . Cela signifie que le gestionnaire de l'atelier décide de ne pas utiliser le fait qu'une opération puisse se faire sur plusieurs machines. Il restreint donc la flexibilité des produits et même celle des machines. De plus la gestion de la file d'attente en sortie des machines suit une règle FIFO (First In First Out)

2. **La situation A-NF** : le gestionnaire suit toujours la gamme de production de la Table 3.10 à savoir que les opérations des produits sont toujours assignées au préalable aux machines. Cependant, la gestion de la file d'attente en sortie des machines ne suit aucune règle d'ordonnancement, celle-ci est laissée à la charge de l'algorithme de gestion du transport. Autrement dit la ressource de transport pourra choisir librement le produit qu'elle souhaite transporter.
3. **La situation NA-F** : Dans ce cas, le gestionnaire n'assigne pas les opérations des produits aux machines, celle-ci se feront à travers l'algorithme de gestion statique ou dynamique du transport. Cependant, la gestion de la file d'attente en sortie des machines suit une règle de type FIFO (First In First Out).
4. **La situation NA-NF** : Dans cette situation, le gestionnaire de l'atelier n'impose aucune assignation des opérations des produits aux machines ; en plus la gestion de la file d'attente des stocks en sortie ne suit aucune règle d'ordonnancement. L'assignation des opérations aux machines ainsi que la gestion des files d'attente des stocks de sortie se fait lors de la production par l'algorithme de gestion des activités de transport.

### Paramétrage de la table d'évaluation

La construction de la table d'évaluation se fait en déterminant l'ordre des critères de chaque action potentielle (opération de transport). Pour ça nous allons adopter la notation suivante :

- $Cr = \{cr_1, cr_2, cr_3, \dots, cr_n\}$ ,  $n \in \mathbb{N}$   
Où  $Cr$  l'ensemble des  $n$  critères  $cr$  retenus pour évaluer les chemins que le transporteur peut prendre et ainsi définir la table d'évaluation des opérations de transports réalisables.
- $O_d = (cr_{\sigma(1)}, cr_{\sigma(2)}, cr_{\sigma(3)}, \dots, cr_{\sigma(n)})$   
Un classement des critères de  $Cr$  avec  $\sigma$  une permutation sur  $[1, n]$ . Dans ce cas, le

critère  $cr_{\sigma(1)}$  est prioritaire au critère  $cr_{\sigma(2)}$ . Nous allons utiliser cette notation pour définir à chaque fois le classement les critères d'une table d'évaluation.

Comme présenté dans le Paragraphe 3.3.3, nous avons choisi cinq critères d'évaluation pour une opération de transport, ce qui nous donne  $5! = 120$  possibilités de classement de critères. Vu le nombre important de possibilités nous avons choisi 5 possibilités parmi les 120 en s'assurant que chaque critère est au moins classé une fois en premier. Pour faire une comparaison efficace a travers la simulation, nous avons défini les cinq classements des critères pour pouvoir implémenter l'algorithme 1. les cinq permutations choisies sont :

- $Od_1 = (dist, TempsAttenteStockSortie, TempsAttenteStockEntree, TRD, TRC)$
- $Od_2 = (TRC, dist, TempsAttenteStockSortie, TempsAttenteStockEntree, TRD)$
- $Od_3 = (TRD, TRC, dist, TempsAttenteStockSortie, TempsAttenteStockEntree)$
- $Od_4 = (TempsAttenteStockEntree, TRD, TRC, dist, TempsAttenteStockSortie)$
- $Od_5 = (TempsAttenteStockSortie, TempsAttenteStockEntree, TRD, TRC, dist)$

Pour le classement  $Od_1$  le critère prioritaire est la distance parcourue par la ressource de transport, pour le classement  $Od_2$  c'est le taux de Remplissage de la zone de chargement qui est le critère le plus important. Pour  $Od_3$  c'est le taux de remplissage de la zone de Déchargement, et pour les classements  $Od_4$  et  $Od_5$ , ce sont respectivement les temps d'attente dans le stocks d'entrée et les temps d'attente dans les stocks de sortie qui sont considérés comme prioritaires.

### Déroulement de la simulation

Nous allons lancer plusieurs simulations ; chacune d'elle aura pour but d'observer les indicateurs de performances du système sous les contraintes de flexibilité et d'ordonnement de file d'attente définies à travers les situations **A-F**, **A-NF**, **NA-F** et **NA-NF**. Pour la simulation statique nous allons rajouter pour chaque situation le classement des critères d'évaluation  $Od_1, Od_2, Od_3, Od_4$  et  $Od_5$ . Nous simulons la production pour les quatre situations **A-F**, **A-NF**, **NA-F** et **NA-NF**. Pour chacune de celle-ci, nous allons utiliser dans un premier temps l'algorithme de gestion statique du transport et dans un second temps l'algorithme dynamique de gestion du transport. Pour la simulation avec l'algorithme de gestion statique, nous allons observer les cas où les classement des critères  $Od_1, Od_2, Od_3, Od_4$  et  $Od_5$  seront utilisés. Pour la simulation avec l'algorithme dynamique, nous n'avons pas besoin de définir un classement des critères d'évaluation, dans la mesure le classement est dynamique.

Nous rappelons les indicateurs de performance que nous allons observer à travers ces simulations :

- Dist : distance parcourue par la ressource de transport tout au long de production.

- Makespan : Durée totale d'écoulement des produits (Makespan)
- Stock entrée : la moyenne des tailles des stocks d'entrée de toutes les machines
- Stock Sortie : la moyenne des tailles des stock de sortie de toutes les machines
- Flexibilité : la moyenne de la flexibilité dynamique du routage.

## Analyse de l'impact de l'ordonnancement dynamique

Dans cette partie, nous allons observer pour chaque situation (type d'atelier) l'effet de la stratégie d'ordonnancement des tâches de transport déployée sur les critères de performances de l'atelier de production flexible.

- La distance parcourue par la ressource de transport ;
- Le temps total d'écoulement des produits(Makespan)
- Les stocks moyen en sortie des machines (stock sortie)
- Les stocks moyens en entrée des machines (stock entrée)

Pour chaque règle d'ordonnancement nous allons comparer les indicateur de performance de l'atelier de production. Chaque indicateur de performance est représenté sur une échelle de  $1/\max_i(Cr_i)$ . De cette manière la règle d'ordonnancement dont le critère de performance sera égal à 1 sera la plus défavorable par rapport aux autres règles. On observant chaque diagramme en toile on pourra conclure que la règle d'ordonnancement dont le polygone a la plus faible surface est la meilleure.

### La situation A-F (Assigné - FIFO)

Le type d'atelier A-F présente le moins de flexibilité dans la mesure où les opérations des produits sur les machines sont affectées avant le début de la production, et que la ressource de transport n'a pas le choix en ce qui concerne la sélection dans une zone de chargement, du produit qu'elle pourrait transporter. La Table 3.11 présente les valeurs des indicateurs de performance de l'atelier en appliquant les différents ordonnancement statiques ( $Od_1, Od_2, \dots$ ) et l'ordonnancement dynamique  $H_{max}$ . Ces mêmes résultats sont présentés sur la courbe en radar de la Figure 3.6 où nous avons représenté les valeurs normalisées des indicateurs de performances. L'indicateur de performance flexibilité de routage est représenté par son inverse c-à-d  $1/Flexibilité$ . Nous avons choisi de représenter l'inverse de la flexibilité par souci de cohérence par rapport aux autres indicateurs de performances du système de production. En effet, comme la Flexibilité est un critère à maximiser alors que les autres sont des critères à minimiser, nous avons décidé d'observer son inverse de manière à ce que tous nos critères soient à minimiser afin d'avoir une meilleure lisibilité du graphique.



TABLE 3.11 – Valeurs des indicateurs de performances pour un atelier A-F

	Ditance [m]	makespan [s]	Stock E	Stock S	Flexibilité
$Od_1$	16466,37	5065	0,21	12,01	2.93
$Od_2$	19068,87	5647	0,22	0,37	0.93
$Od_3$	18808,99	5478	1,22	16,77	2.65
$Od_4$	18879,3	5499	0,3	12,41	2.47
$Od_5$	21714,41	6145	0,21	0,29	0.82
$H_{max}$	16332,9	5021	0,22	12,12	3.02

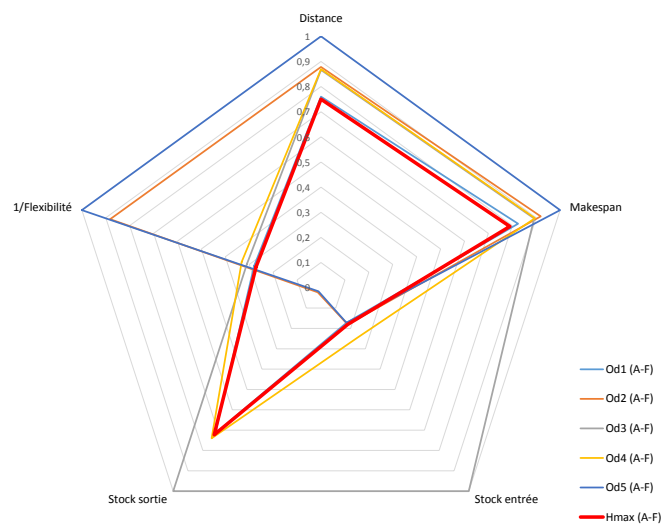


FIGURE 3.6 – Comparaison graphique des différentes stratégies d'ordonnancement dans le cas d'un atelier type A-F

En effet, un bon algorithme d'ordonnancement doit minimiser les valeurs des indicateurs de performance Distance, Makespan, et charge des stock en entrée et sortie des machines de production et maximiser les flexibilités de routage afin de donner au transporteur toujours la possibilité de prendre divers choix d'acheminements des produits. Ceci permettra d'éviter facilement des situations de blocage en cas de panne ou d'urgence.

Les valeurs de la Table 3.11 ainsi que les radars de la Figure 3.6 montrent bien que la stratégie d'ordonnancement dynamique n'est pas toujours la meilleure pour tous les indicateurs, mais elle permet d'avoir un bon compromis sur l'ensemble des indicateurs de performances. Par exemple l'ordonnancement dynamique présente une flexibilité moyenne de 18.45 contrairement à la stratégie  $Od_3$  qui présente une flexibilité moyenne de 2.65, mais cette même stratégie présente des valeurs moins intéressantes sur les autres indicateurs de performances comme on peut l'observer sur la Figure 3.6.

Les courbes de la stratégie dynamique  $H_{max}$  et la stratégie  $Od_1$  sont très rapprochées (pratiquement superposées) malgré un léger avantage pour la stratégie dynamique qu'on peut observer sur la Table 3.11. Ceci peut être expliqué par le fait que dans le cas d'étude, il a été considéré une taille maximale de 50 qui représente une grande taille pour un lot de 150 produits. Ce qui réduit la probabilité d'avoir des situations de saturation des stocks



et favorise souvent le critère distance dans l'ordonnanceur dynamique qui est le premier critère d'évaluation de la stratégie  $Od_1$ .

### La situation A-NF (Assigné - Non FIFO)

Dans la situation A-NF les opérations des produits sur les machines sont toujours déterminées avant le lancement de la production alors que la règle de sélection des produits dans les stocks de sortie n'est pas imposée à la ressource de transport. Le nombre maximal de chemins possibles pour le transporteur est donc supérieur au type d'atelier A-F et il est au plus égale au nombre de de produits présent dans les stocks de sortie multiplié par le nombre de stock de sortie non vide.

Comme nous l'avons fait pour le type d'atelier précédent (A-F), nous avons représenté les valeurs des indicateurs de performance sur la Table 3.12 et nous avons représenté leurs valeurs normalisées en graphique radar sur le Figure 3.7. On peut déjà remarquer que les valeurs de la flexibilité sont nettement plus élevé comparé au premier type d'atelier (A-F).

Pour cette situation, nous pouvons aussi remarquer que la stratégie d'ordonnement dynamique n'est jamais la meilleure stratégie pour un indicateur spécifique mais elle représente le meilleur compromis pour l'ensemble des indicateurs . De plus la valeur donnée par l'ordonnement dynamique pour le critère n'est pas loin de la meilleure valeur obtenues par les autres stratégies.

Sur le Graphique on peut remarquer une grande différence entre l'ordonnement dynamique ( $H_{max}$ ) et les ordonnancements  $Od_2$  et  $Od_5$  pour les tailles les stocks notamment pour les stocks de sortie des machines. En effet, malgré la différence qui est assez importante (10.16 pour  $H_{max}$  et 0.51 et 0.39 respectivement) cette taille reste acceptable puisqu'elle représente uniquement 20% de la saturation du stock qui provoque un arrêt de la machine.

TABLE 3.12 – Valeurs des indicateurs de performances pour un atelier A-NF

	Ditance [m]	makespan [s]	Stock E	Stock S	Flexibilité
$Od_1$	16362,4	5028	2,31	10,5	16,394
$Od_2$	19615,31	5735	0,56	0,51	14,95
$Od_3$	18769,85	5483	1,05	16,44	19,95
$Od_4$	18324,05	5410	0,28	14,7	19,48
$Od_5$	20316	5866	0,22	0,39	14,65
$H_{max}$	17288,07	5237	0,86	10,16	17,53

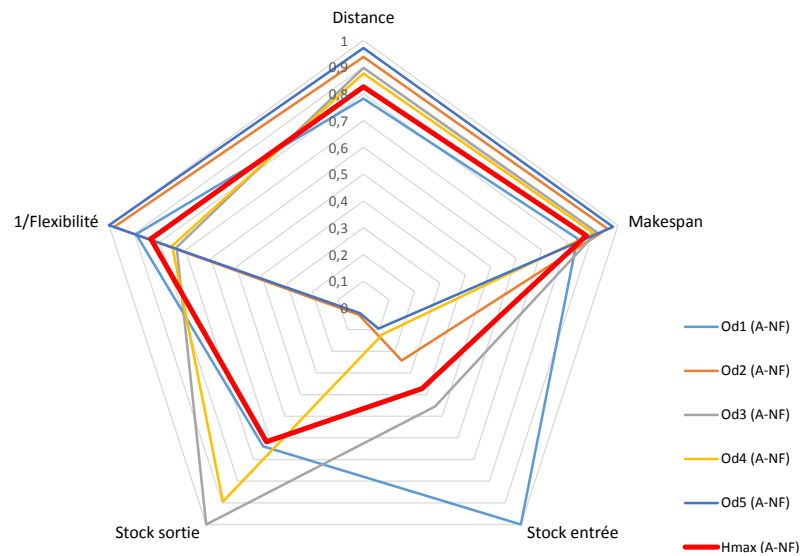


FIGURE 3.7 – Comparaison graphique des différentes stratégie d’ordonnancement dans le cas d’un atelier type A-NF

### La situation NA-F (Non Assigné - FIFO)

Dans la situation NA-F l’affectation des opérations des produits aux machines se fait de manière dynamique au cours de la production alors que la règle de sélection des produits dans les stocks en sortie des machines est imposée et suit la règle FIFO. On ne peut pas affirmer que le nombre de choix possibles pour le transporteur augmente, comparé à la précédente configuration de l’atelier puisque le nombre maximale de chemins que le transporteur peut prendre est égale à la somme de nombre de machines capable de faire la prochaine opération du produit en tête de la file d’attente de chaque stock de sortie des machines en considérant le stock d’entrée de l’atelier comme étant un stock de sortie d’une machine.

La Table 3.13 ainsi que la Figure 3.8 présente les valeurs des différents indicateurs de performances pour chaque stratégie d’ordonnancement testée. On peut observer à travers les résultats que la stratégie dynamique propose toujours un compromis entre tous les indicateurs de performance. Elle propose le plus mauvais choix pour les stocks d’entrée des machines, mais cela reste acceptable lorsqu’on observe la valeur de ce stock qui est très faible ( 0.8 produit en moyenne).

On peut remarquer que lorsque l’on applique l’algorithme de gestion dynamique, le système a tendance à avoir le même comportement que si l’on utilisait l’algorithme statique avec le classement des critères  $Od_1$ .

TABLE 3.13 – Valeurs des indicateurs de performances pour un atelier NA-F

	Ditance [m]	makespan [s]	Stock E	Stock S	Flexibilité
$Od_1$	13333,04	4486	0,52	11,54	5.24
$Od_2$	17650,37	5343	0,2	0,38	2.03
$Od_3$	17141,64	5127	0,76	16,53	4.4
$Od_4$	17343,81	5201	0,69	13	4.19
$Od_5$	18435,07	5445	0,21	0,38	2.2
$H_{max}$	14358,56	4693	0,80	11,21	4.87

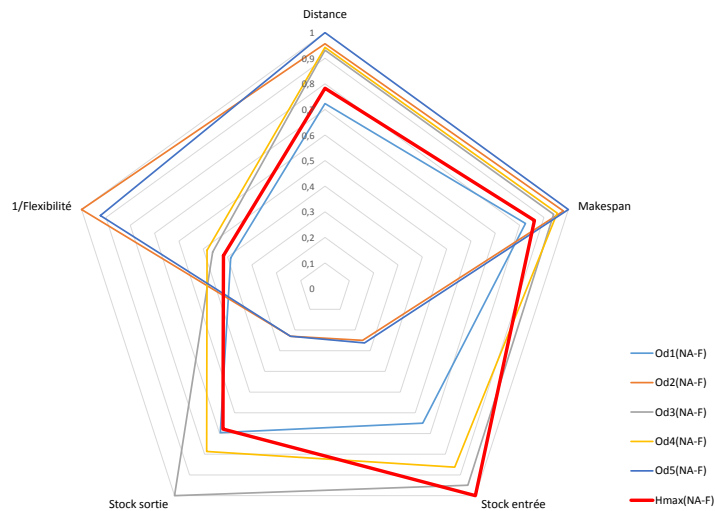


FIGURE 3.8 – Comparaison graphique des différentes stratégie d'ordonnancement dans le cas d'un atelier type NA-F

### La situation NA-NF (Non Assigné - Non FIFO)

Dans la situation NA-NF, l'affectation des opérations des produits aux machines se fait de manière dynamique au cours de la production ainsi que la sélection des produits dans les stocks en sortie des machines. Le nombre maximal de choix possibles pour le transporteur augmente étant donné qu'il est égal à la somme des chemins possibles pour tous les produits présents dans les stocks de sortie des machines et le stock d'entrée de l'atelier.

Les résultats de simulation présentés sur la Table 3.14 et la Figure 3.9 ne montrent pas une augmentation considérable de la flexibilité du routage parce que le cas d'étude ne donne pas une flexibilité totale des machines en plus de cela le nombre de produits dans les stocks de sortie est relativement maîtrisé, ce qui influence directement le nombre de chemins possibles.

La stratégie d'ordonnancement dynamique a aussi permis de proposer une solution avec un bon compromis pour les différents indicateurs de performance considérés. La stratégie de classement des critères  $H_{max}$ , propose une solution proche la stratégie  $Od_1$ , sauf pour la charge sur la machine qui est nettement améliorée avec la stratégie dynamique  $H_{max}$ . En effet l'observation des classements instantanés des critères d'évaluation

TABLE 3.14 – Valeurs des indicateurs de performances pour un atelier NA-NF

	Ditance [m]	makespan [s]	Stock E	Stock S	Flexibilité
$Od_1$	13074,45	4449	4,184	15,44	20,44
$Od_2$	15377,96	4872	0,4	0,61	15,86
$Od_3$	17700,86	5241	1,24	16,65	20,43
$Od_4$	17542,81	5237	0,29	15,72	20,65
$Od_5$	18183,21	5417	0,208	0,35	14,97
$H_{max}$	13787,48	4576	4,01	10,63	19,65

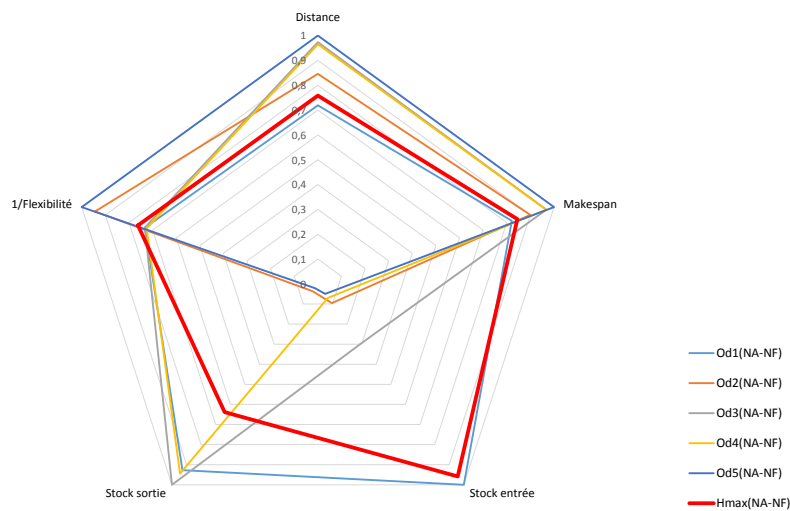


FIGURE 3.9 – Comparaison graphique des différentes stratégie d'ordonnancement dans le cas d'un atelier type NA-NF

d'une opération de transport lors de l'exécution de l'algorithme dynamique, montre que le critère Distance parcourue est souvent prioritaire. Ceci est dû à la forme symétrique de la disposition des machines dans l'atelier du cas d'étude. On imagine que le changement de disposition des machines ne vas pas permettre à l'ordonnancement  $Od_1$ , qui donne une priorité 1 à la distance parcourue) d'aboutir à d'aussi bons résultats que ceux présentés par notre cas d'étude.

Pour les quatre situations que nous avons étudiées on voit clairement que la stratégie d'ordonnancement dynamique propose toujours un très bon compromis entre plusieurs indicateurs de performance. Il faut rappeler que ces indicateurs de performances ne sont pas tous identiques aux critères d'évaluation des chemins du transporteur. Les résultats ont montré clairement que la classification des critères pour permettre instantanément de considérer le plus de critères possibles, donne de bons résultats par rapport à des indicateurs globaux tels que le makespan ou la distance totale parcourue par le transporteur.

## Analyse des indicateurs de performances du système par type d'atelier de production

Dans cette section nous allons présenter les indicateurs de performance du système de production en observant leurs variations en fonction de la stratégie d'ordonnancement des files d'attente dans les zones de chargement, de la flexibilité des produits et aussi de la flexibilité des machines. Cette comparaison va nous permettre de comprendre l'effet de l'ordonnancement sur les indicateurs de performance et d'identifier s'il existe une stratégie d'ordonnancement des files d'attente dans les zones de chargement et un niveau de flexibilité pour lequel notre stratégie de gestion dynamique serait la plus efficace.

### Analyse de la distance totale parcourue par la ressource de transport

La Table 3.15 donne la distance parcourue par la ressource de transport dans les situations **AF**, **ANF**, **NA-F** et **NA-NF**. Pour chacune d'elles nous avons appliqué d'une part l'algorithme statique avec le rangement des critères d'évaluation  $Od_1$ ,  $Od_2$ ,  $Od_3$ ,  $Od_4$  et  $Od_5$ , et d'une autre part l'algorithme de gestion dynamique des critères d'évaluation  $H_{max}$ .

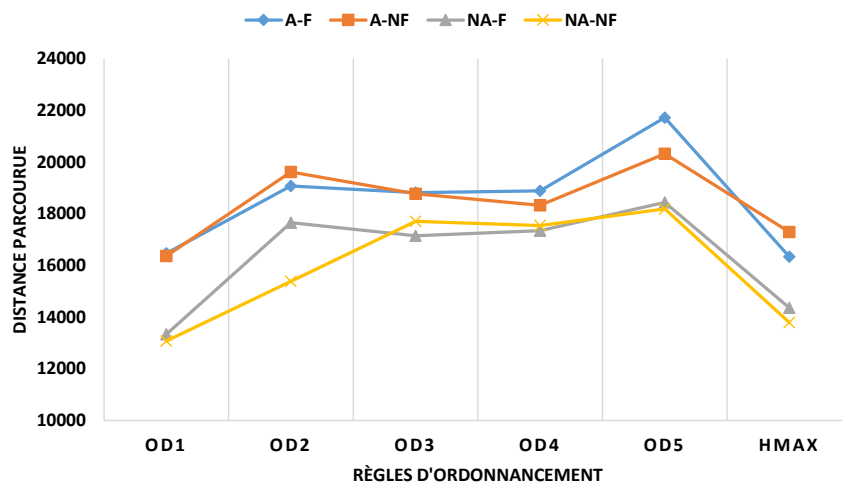


FIGURE 3.10 – Distance parcourue par la ressource de transport [m]

Au vu des résultats présentés sur la Table 3.15 et la Figure 3.10 on peut faire les

TABLE 3.15 – Distance parcourue par la ressource de transport

	<b>A-F</b>	<b>A-NF</b>	<b>NA-F</b>	<b>NA-NF</b>
$Od_1$	16466.37	16362.4	13333.04	13074.45
$Od_2$	19068.87	19615.31	17650.37	15377.96
$Od_3$	18808.99	18769.85	17141.64	17700.86
$Od_4$	18879.30	18324.05	17343.81	17542.81
$Od_5$	21714.41	20316.00	18435.07	18183.21
$H_{max}$	16332.90	17288.07	14358.56	13787.48

observations suivantes :

1. La non-assignation au préalable des opérations aux machines diminue la distance parcourue par la ressource de transport. Pour cela, il suffit d'observer que pour chaque simulation les résultats pour le cas **AF** et le cas **NAF**. On voit que lorsque l'on n'impose pas le choix des machines en début de production (cas **NAF**), la distance diminue en moyenne de 8 à 20 %, comparativement au cas où les produits sont au préalable assignés. Si l'on compare aussi deux à deux les cas **ANF** et **NANF** nous faisons le même constat qui est celui de la diminution de la distance parcourue par la ressource de transport. Ce constat est mis en évidence sur la Figure 3.10 où nous pouvons voir les deux courbes des situations non assignées **NANF** et **NAF** en dessous des deux courbes des situations assignés **ANF** et **AF**
2. Dans une situation où les opérations ne sont pas assignées le fait de permettre à l'algorithme de décider librement du produit qui sera transporté (les cas Non FIFO), améliore aussi la distance parcourue par le transporteur comparativement à une situation où la gestion de la file d'attente suit une règle FIFO. En effet, si l'on compare les cas **NAF** et **NANF**, on observe une diminution de la distance parcourue par la ressource de transport.
3. Dans une situation où le gestionnaire décide d'affecter les opérations des produits aux machines avant le début de la production, on observe que le fait de laisser l'algorithme décider du produit dans chaque stock qu'il peut transporter, n'améliore pas toujours la distance parcourue par la ressource de transport. Il suffit de prendre les cas **Od2-AF** et **Od2-ANF** pour remarquer que la distance dans le cas **Od2-ANF** ne diminue pas par rapport au cas **Od2-AF** ;
4. L'utilisation de l'algorithme de gestion dynamique du transport  $H_{max}$  donne de meilleurs résultats en termes de distance que dans tous les autres cas de figure exceptés les cas où le classement des critères est  $Od_1$ . Cela s'explique par le fait que l'ordre des critères  $Od_1$  donne au critère distance élémentaire parcourue la première priorité. Malgré cela, l'écart en terme de performance est inférieur à 8% entre les distances de l'ordonnancement  $Od_1$  et l'ordonnancement dynamique  $H_{max}$ .
5. L'utilisation de l'ordonnancement dynamique  $H_{max}$  paraît plus efficace dans la situation **A-F**, où il donne de meilleur résultat que l'ordonnancement  $Od_1$  qui met la distance parcourue élémentaire en priorité. Cette configuration est possible parce que la minimisation d'un critère locale (la distance parcourue pour effectuer une tâche de transport élémentaire) ne signifie pas l'optimisation de la distance globale parcourue par le transporteur pour la fabrication de tout le lot.

### Analyse du makespan

La Table 3.16 ainsi que la Figure 3.11 donne le temps total d'écoulement du lot des produits ( 150 produits) dans les situations **AF**, **ANF**, **NA-F** et **NA-NF**. Pour chacune d'elles nous avons appliqué d'une part l'algorithme statique avec le rangement des critères

d'évaluation  $Od_1$ ,  $Od_2$ ,  $Od_3$ ,  $Od_4$  et  $Od_5$ , et d'une autre part l'algorithme de gestion dynamique  $H_{max}$ .

À priori, les stratégies d'ordonnancement  $Od_4$  et  $Od_5$  doivent donner de meilleurs résultats puisque ceux-ci mettent en avant un critère d'évaluation en relation avec le temps.

TABLE 3.16 – Temps total d'écoulement des produits (Makespan)

	A-F	A-NF	NA-F	NA-NF
$Od_1$	5065	5028	4486	4449
$Od_2$	5647	5735	5343	4872
$Od_3$	5478	5483	5127	5241
$Od_4$	5499	5410	5201	5237
$Od_5$	6145	5866	5445	5417
$H_{max}$	5021	5237	4693	4576

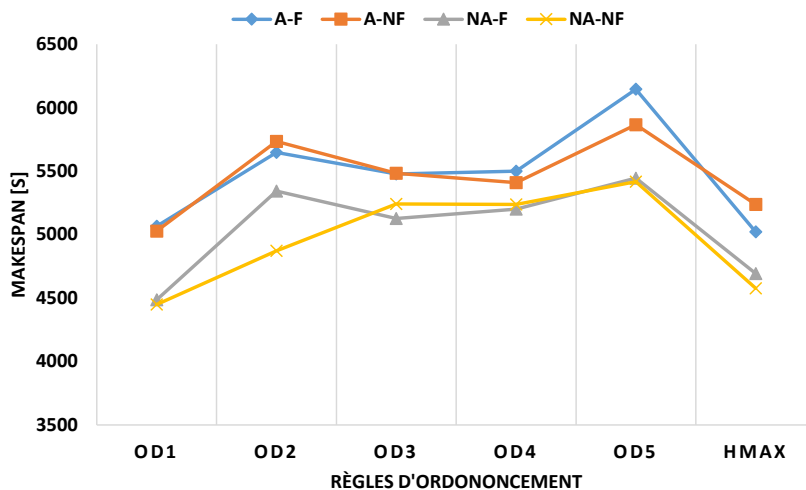


FIGURE 3.11 – Makespan des différents stratégie d'ordonnancement pour toutes les situation considérés

À travers l'analyse des résultats on peut déduire les conclusions suivantes :

1. La non-assignation au préalable des opérations aux machines diminue le temps total d'écoulement des produits, on peut observer sur la Figure 3.11 que les courbes des situations **NA-F** (jaune) et **NA-NF** (grises) sont tout le temps en dessous des autres situation où les opérations des produits sont assignées à des machines (jobshop). En effet, lorsque l'on n'impose pas le choix des machines en début de production (cas **NA-F** et **NA-NF**), le Makespan diminue de 5.3 à 11.5 % pour le cas FIFO, et de 3.1 à 15 % Non FIFO comparativement aux cas où les produits sont au préalable assignés.
2. Dans les situations où les opérations des produits sont non assignées(**NAF** et **NANF**), le fait d'utiliser ou non une FIFO pour la gestion des stocks d'entrée ne permet pas de conclure sur la valeur du makespan. En effet, les courbes **NAF**

(grise) et **NANF** (jaune), se croisent en plusieurs points. Cela signifie que le makespan dépend plus de l'ordonnancement utilisé que de la règle de gestion du stock de sortie de la machine.

3. L'utilisation de l'algorithme de gestion dynamique du transport  $H_{max}$  donne de meilleurs résultats en termes de Makespan que dans tous les autres cas de figure exceptés les cas où l'ordre des critères est  $Od_1$ . On observe un écart de performance inférieur à 5% entre les deux ordonnancements  $H_{max}$  et  $Od_1$ .
4. Malgré le fait que les stratégies d'ordonnancement  $Od_4$  et  $Od_5$  favorise la réduction des temps d'attente des produits dans les stocks d'entrée et de sortie des machines respectivement, ils ne présentent pas de bon résultats vis à vis du makespan comme on peut l'observer sur la Figure 3.11.
5. La stratégie d'ordonnancement dynamique donne toujours des makespans réduits malgré que l'ordonnancement  $Od_1$  présente des résultats légèrement meilleurs exceptés pour les ateliers en situation **A-F** où l'ordonnancement dynamique est meilleur.

### Stocks en entrée des machines

La Table 3.17 et la Figure 3.12 donne la taille moyenne des stocks en entrée des quatre machines dans les situations **AF**, **ANF**, **NA-F** et **NA-NF**. Pour chacune d'elle nous avons appliqué d'une part l'algorithme statique avec le rangement des critères d'évaluation  $Od_1$ ,  $Od_2$ ,  $Od_3$ ,  $Od_4$  et  $Od_5$ , et d'une autre part l'algorithme de gestion dynamique avec la règle de classement des entropies maximales  $H_{max}$ .

TABLE 3.17 – Taille moyenne des stocks en entrée

	<b>A-F</b>	<b>A-NF</b>	<b>NA-F</b>	<b>NA-NF</b>
Od1	0.21	2.31	0.52	4.184
Od2	0.22	0.56	0.2	0.4
Od3	1.22	1.05	0.76	1.24
Od4	0.30	0.28	0.69	0.29
Od5	0.21	0.22	0.21	0.208
Hmax	0.22	0.86	1.87	4.01

À travers la Figure 3.12 on peut observer que la stratégie d'ordonnancement  $Od_5$  minimise les stocks en entrée des machines ce qui corrobore l'efficacité du critère d'évaluation somme des temps de processus dans le stock d'entrée. En général on obtient un stock d'entrée élevé dans le cas d'une panne ou d'une saturation du stock de sortie de la machine. Comme les capacités maximales des stocks sont limitées à 50, il est difficile de tomber dans des cas de blocage du système. On peut aussi observer que la stratégie d'ordonnancement dynamique donne de meilleurs résultats pour les ateliers type **AF** (jobshop avec FIFO). On peut remarquer que l'augmentation de la complexité des ateliers (de **A-F** à **NA-NF**) est proportionnelle à l'augmentation de la taille moyenne des stocks d'entrée des machines.



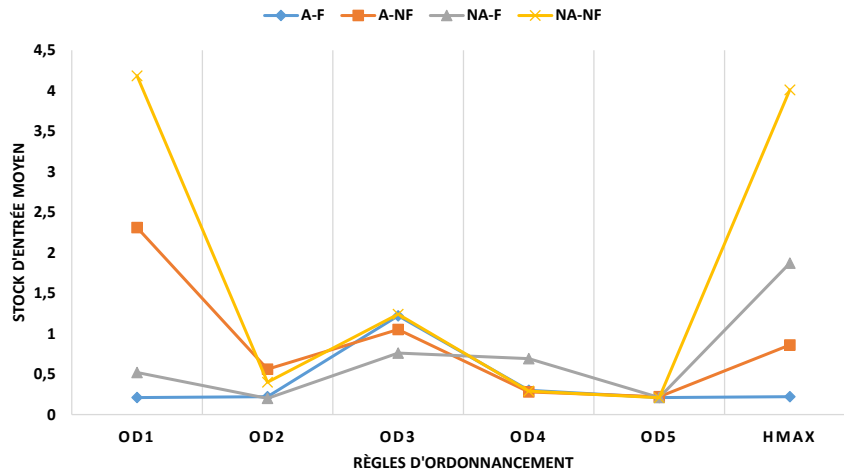


FIGURE 3.12 – Charge moyenne des stock en entrée

### Stocks en sortie des machines

La Table 3.18 et la Figure 3.13 donnent la taille moyenne des stocks en sortie des machines de production pour tous l'atelier flexible dans les situations **A-F**, **A-NF**, **NA-F** et **NA-NF**. Pour chacune d'elle nous avons appliqué d'une part l'algorithme statique avec le rangement des critères d'évaluation  $Od_1$ ,  $Od_2$ ,  $Od_3$ ,  $Od_4$  et  $Od_5$ , et d'une autre part l'algorithme de gestion dynamique  $H_{max}$ .

TABLE 3.18 – Charge moyenne totale des stocks en sortie des machines

	<b>A-F</b>	<b>A-NF</b>	<b>NA-F</b>	<b>NA-NF</b>
$Od_1$	12.01	10.50	11.54	15.44
$Od_2$	0.37	0.51	0.38	0.61
$Od_3$	16.77	16.44	16.53	16.65
$Od_4$	12.41	14.70	13.00	15.72
$Od_5$	0.29	0.39	0.38	0.35
$H_{max}$	12.12	10.16	11.21	10.63

Lorsque l'on utilise le rangement des critères  $Od_2$  et  $Od_5$  on observe que la charge moyenne des stocks en sortie est plus faible que les autres cas. En effet on a en moyenne 1 produit dans les stocks en entrée des machines. Cela s'explique par le fait que le critère le plus important pour le l'ordonnement  $Od_2$  est le taux de remplissage des stocks en sortie ( $TRC$ ) des machines. Cela signifie qu'on aura tendance à réduire la taille du stock en sortie des machines qui ont un stock élevé. Pour l'ordonnement  $Od_5$  le critère prioritaire est le temps d'attente minimal dans le stock de sortie des machines ( $TempsAttenteStockSortie$ ), cela veut dire que l'on va favoriser les opérations de transport pour lesquelles le temps d'attente dans la zone de Chargement est le plus faible. Il faut noter que les critères d'évaluation :  $TRC$  et  $TempsAttenteStockSortie$  sont très corrélés.

La Figure 3.13 montre, pour la stratégie d'ordonnement  $Od_1$ , que la charge des stocks en entrée est comprise entre 10 et 16 produits. L'observation des résultats détaillés

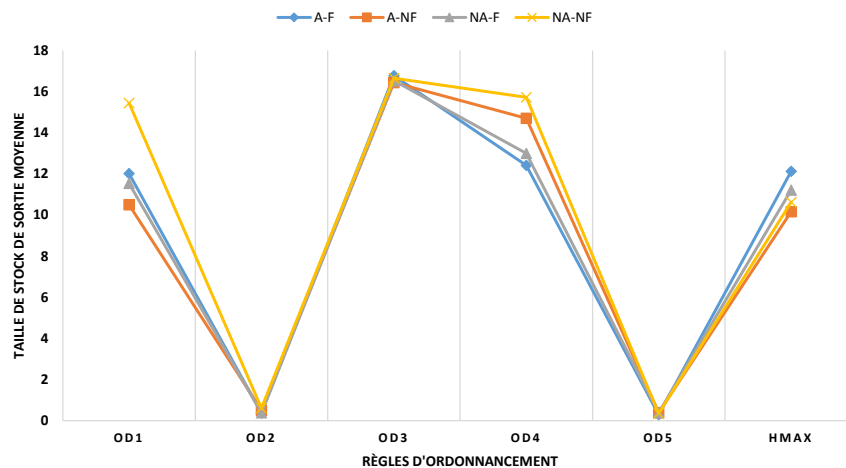


FIGURE 3.13 – Charge moyenne des stocks en sortie des machines

de la simulation (cf. Annex ??) montre que le stock de sortie subit plusieurs situations de saturation, ce qui explique la valeurs élevé de la taille moyenne des stocks. Ce qui est très supérieure à la valeur moyenne des stocks en entrée lorsque l'on applique les classements des critères Od2 et Od5 par exemple.

Il faut noter que la moyenne des charges des stocks dans un atelier de production ne permet pas toujours d'observer l'atteinte de seuil critique ou bien de la capacité maximale des stocks. Pour illustrer observons la Table 3.19 qui présente la charge des stocks en entrée et en sortie de chaque machine. On peut aussi voir la valeur maximale de la charge de ces stocks ainsi que leur écart type. Si la charge moyenne des stocks en de sortie est de 12.01 comme nous l'avons relevé dans la Table 3.17 en rouge, et la charge des stocks en entrée est de 0.21 (cf. Table 3.18 en rouge), en observant les charges des stocks sur chaque machine sur la Table 3.19 on peut faire les remarques suivantes :

- La charge maximale de chaque stock en entrée est comprise entre 1 et 2. Cela signifie que le nombre de produit reste très limité en entrée des machines dans le cas Od1-A-F.
- Quand on compare la charge moyenne des stocks en sortie des machines on observe que les Machine 1 (0.454) et Machine 2 (2.253) sont le plus souvent moins chargées que les Machine 3 (24.896) et Machine 4 (20.442). Ceci est dû au fait que la stratégie d'ordonnement  $Od_1$  favorise la distance, et le transporteur se trouve souvent pas loin des Machine 1 et Machine 2 et ne considère donc pas les Machine 3 et Machine 4 que lorsque les stocks de sortie des Machine 1 et Machine 2 sont vides.
- En ce qui concerne la stratégie d'ordonnement dynamique, on peut observer que les charges des stocks de sorties dans le cas **A-F** sont relativement équilibrés (cf. Annexe ?? Table ??)

TABLE 3.19 – Stock des machines dans le cas Od1-A-F

	Machine 1	Machine 2	Machine 3	Machine 4
Nbr. Moy Produit SE	0.229	0.222	0.167	0.222
Ecart type charge stocks en entrée	0.426	0.416	0.373	0.416
Nbr Max Produit SE	2	1	1	1
Charge Moyenne des SS	0.454	2.253	24.896	20.442
Ecart type charge des SS	0.658	1.374	14.461	11.864
Charge maximale des SSe	3	6	47	40

### Analyse de La flexibilité de routage

Nous allons présenter les valeurs moyennes de la flexibilité de routage de l’atelier de production durant la phase de production. Dans la Table 3.20 on présente pour chaque cas de figure les statistiques suivantes :

- La moyenne arithmétique de la flexibilité de routage durant la phase de production(Moyenne FR).
- Les valeur extrêmes (maximum et minimum)de la mesure de la flexibilité de routage
- La valeur moyennes des opération de transport réalisables à un instant de décision (Moy.Op).

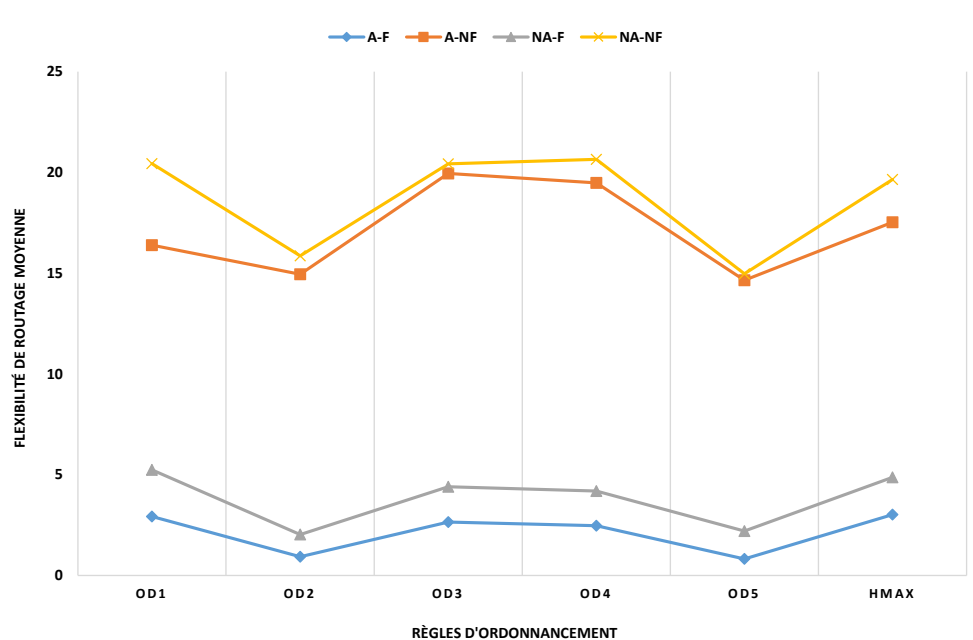


FIGURE 3.14 – Variation de la flexibilité de routage selon l’ordonnancement et le type d’atelier

L’observation de la Figure 3.14 En observant l’entropie de l’atelier de production on observe que

TABLE 3.20 – Entropie du système de production

	A-F			A-NF			NA-F			NA-NF		
	Moy	Max	N.op	Moy	Max	N.op	Moy	Max	N.op	Moy	Max	N.op
$Od_1$	2.93	4.99	3.84	16.94	23.82	97.40	5.24	8.97	6.31	20.44	24.65	140.39
$Od_2$	0.93	3.80	2.23	14.95	24.01	77.81	2.03	6.94	3.36	15.86	25.67	90.30
$Od_3$	2.65	7.34	3.36	19.95	24.78	123.83	4.40	10.24	4.89	20.43	25.67	147.52
$Od_4$	2.47	5.84	3.19	19.48	24.79	120.72	4.19	10.53	4.65	20.65	25.67	146.95
$Od_5$	0.82	3.89	2.01	14.65	24.82	77.46	2.20	7.07	3.64	14.97	24.93	89.11
$H_{max}$	3.02	5.5	3.93	17.53	23.94	102.01	4.87	9.41	6.05	19.65	25.22	121.94

- L'entropie augmente avec le nombre moyen d'opérations de transport possibles par instant de décision.
- Pour chaque ordonnancement on remarque que le nombre d'opérations augmente lorsque l'on rajoute de la flexibilité au produit et lorsque l'on permet à la ressource de gérer la file d'attente des stocks de sortie.
- L'amélioration de la flexibilité des opérations et des produits permet l'augmentation de la flexibilité de routage. Cela s'explique par le fait que l'augmentation de la flexibilité des opérations et des produits augmente le nombre de lignes de la table d'évaluation du transport.
- Lorsque l'ordonnancement de la file d'attente en sortie des machines n'est pas FIFO (situations **A-NF** et **NA-NF**), on observe que l'entropie de la table de transport augmente. Cela signifie que la flexibilité du routage augmente elle aussi comme on peut le voir sur la Figure 3.14 ou les courbes **A-NF** et **NA-NF** (courbes jaune et orange) présentent des valeurs bien plus supérieures que celles des situations **A-NF** et **NA-NF** (courbes bleu et grise).
- Même si la différence n'est pas très importante, les situations non assignées présentent des flexibilités meilleurs que les situations assignées. Sur la Figure 3.14; la courbe **NA-NF** (courbe jaune) est supérieure à la courbe **A-NF** (courbe orange), et la courbe **NA-F** est supérieure à la courbe **A-F**. On peut déduire que la relaxation des contraintes augmente la flexibilité de routage dans les atelier flexibles.
- On peut remarquer que les ordonnancements  $Od_3$  et  $Od_4$  donne un niveau élevé de flexibilité de routage surtout pour les situations Non FIFO ( **A-NF** et **AN-NF**). Le point commun entre ces deux ordonnancements est le fait qu'ils favorisent la minimisation des stocks d'entrée mais donnent des stocks de sorties élevés (cf. Figure 3.9). Ceci donne plus de choix au transporteur dans les cas Non FIFO et donc une flexibilité élevé.

## Conclusion

Dans ce chapitre nous avons tenté de trouver une méthode gestion des activités de transport dans un atelier de production flexible en proposant une stratégie d'ordonnancement dynamique du transport. Pour cela nous avons procédé de la manière suivante :

- L'élaboration d'une table d'évaluation des activités de transport. Cette dernière permet d'effectuer une évaluation multicritères de chaque tâche élémentaire de transport possible.
- La proposition d'une méthode d'aide à la décision pour choisir parmi les opérations réalisables, celle qui serait meilleure suivant des critères au préalablement définis par les gestionnaires du système de production.

- Le développement d'un algorithme d'ordonnancement dynamique basé sur une approche entropique pour classer les critères d'évaluation des opérations de transports possibles.

Afin de prouver l'efficacité de notre ordonnancement, nous avons utilisé les modèles de simulation des systèmes de production flexibles que nous avons développés dans le chapitre précédent. Nous avons défini un système de production pouvant se retrouver dans quatre situations (A-F, NA-F, A-NF et NA-NF). Nous avons effectué des comparaisons de notre ordonnancement dynamique avec 5 ordonnancements statiques.

Pour cela, nous avons défini des indicateurs de performances liés à différents aspects de l'atelier de production tels que : le temps de production, la charge des machines, le confort du transporteur ( la distance) et la robustesse du système (flexibilité de routage). Ce dernier indicateur a été calculé par une méthode basée sur l'entropie et que nous avons détaillée dans ce même chapitre.

Les résultats ont montré que l'utilisation de l'ordonnancement dynamique ne permet pas de minimiser un indicateur de performance bien défini mais offre un bon compromis entre plusieurs indicateurs de performances hétérogènes et contradictoires. Nous avons constaté aussi que notre proposition est efficace dans plusieurs situations de l'atelier surtout le cas où l'atelier est contraint (la situation AF).



# Conclusion générale et perspectives

Notre objectif tout au long de ce travail de recherche était d'aider des PME dans l'adoption d'un système de production flexible. Pour cela, nous avons proposé un outil de modélisation et de simulation des systèmes de production flexibles. L'intérêt de cette démarche étant de permettre aux PME d'évaluer le coût, et d'étudier les performances d'un système de production avant son implantation. Un autre avantage de cet outil était d'apporter des solutions aux problématiques de gestion de production telles que l'ordonnancement, l'implantation, ou encore la gestion des ressources de transport. Pour atteindre cet objectif nous avons procédé par :

- Le choix d'un type de modèle de simulation
- Le développement et a validation des modèles.
- La proposition d'un algorithme de gestions des flux et des ressources de transport.

## Choix d'un type de modèle de simulation

Nous avons choisi d'utiliser les réseaux de Pétri colorés hiérarchiques pour développer nos modèles de simulation. Ce choix se justifie par le fait que :

- Le réseau de Pétri étant un modèle graphique, il permet une meilleure analyse et facilite la compréhension comparativement à un modèle analytique ;
- Les réseaux de Pétri sont de puissants outils de vérification des systèmes à événements discrets. En effet les propriétés des réseaux de pétri telles que la bornitude, exclusion mutuelle ou encore la vivacité permettent de représenter des situations réelles (partage des ressources, tâches disjonctives etc.)
- L'existence de nombreuses plateformes de simulation des RdP.
- La possibilité de développer une approche objet avec réseaux de pétri hiérarchiques.

Il faut aussi reconnaître des inconvénients au Réseau de Pétri Colorés hiérarchiques à savoir : la difficulté à en extraire un modèle analytique du fait de la complexité. Cela rend par exemple très difficile la mise en oeuvre de méthode d'optimisation exactes et même certaines métaheuristiques. Enfin il faut reconnaître que même si les réseaux de



pétri permettent de représenter des systèmes dynamiques, la structure du modèle elle reste figée et la gestion de problème d'ordres spatiaux (notamment les problèmes de recouvrement de machines) reste très difficile comparativement à des modèles basés sur des approches agents.

Cependant, les réseaux de pétri permettent de gérer plus facilement des problèmes où il y a une distribution des tâches et des prises de décisions locales (prise de décision au niveau d'une machine ou encore gestion locale de l'activité de transport).

### **Le développement et validation des modèles.**

Nous avons utilisé les réseaux de pétri afin de développer des modèles objets d'entités constituant les ateliers de production flexibles à savoir les machines flexibles, les stocks, les produits ou encore les ressources de transport. Nous avons modélisé les contraintes inhérentes à la flexibilité tels que : le multi-outillage, la flexibilité des produits, la flexibilité des moyens de transport. . . Afin de faciliter la modélisation et réduire les paramètres de simulation, nous avons émis quelques hypothèses simplificatrices telles que :

- La non prise en compte de l'encombrement des machines ; en effet nous avons assimilé l'ensemble machines-stocks à un seul point sur un repère.
- Le déplacement des transporteurs se fait en ligne droite ce qui n'est pas forcément le cas parce que la trajectoire d'une ressource de transport peut être modifiée au cours de la production.
- Le déplacement de la ressource de transport à vitesse constante peut fausser le calcul des temps de déplacements par rapport à un système réel.

Même si nous avons présenté des modèles de produits, de stocks, de ressources de transport, nous avons mis un accent particulier sur le modèle de la machine flexible qui est un élément essentiel d'un système de production flexible. Nous avons pu intégrer à ce modèle le multi-outillage, les temps de préparation, la différenciation des produits.

Nous devons reconnaître que les hypothèses simplificatrices que nous avons posées ne permettent pas de représenter certaines situations. Par exemple l'assimilation d'une machine à un point ne permet pas d'intégrer les problèmes d'ergonomie. De plus le déplacement en ligne droite des ressources de transport n'est pas toujours réaliste. Nous pensons que la prise en compte de ces hypothèses demandent l'utilisation de modèles basés sur des approches agents.

Nous avons évalué la validité de nos modèles à l'aide d'un cas réel que nous avons proposé en section 2.5. Nous avons éprouvé nos modèles en confondant nos résultats de simulation à ceux relevés sur la chaîne de production du Laboratoire du CESI de Rouen. Les hypothèses simplificatrices que nous avons faites ont très peu remis en cause la validité de notre modèle dans le cadre de la confrontation entre les résultats réels et ceux de la simulation.

Toutefois le cas réel de la chaîne de production ne permet pas de généraliser la validation de nos modèles à des systèmes de production flexibles à flux croisés et à ressources de transports limitées. En effet la configuration d'une ligne de production impose un flux unidirectionnel et celle d'un convoyeur exclu le problème du partage des ressources de transport. Nous avons tout de même abordé le problème de la gestion des ressources de transport dans un atelier de production flexible à flux croisés dans le chapitre 3.

### **Proposition d'un algorithme de gestions des flux et des ressources de transport**

Dans le chapitre 3, nous avons proposé un algorithme de gestion du transport dans un atelier de production flexible avec des flux de production. Nous avons intégré cet algorithme aux modèles précédemment développés dans le chapitre 2. Grâce à cet algorithme nous avons proposé une méthode d'aide à la décision visant à :

- Arbitrer des conflits de partage des ressources lors des opérations de transport dans l'atelier en proposant un algorithme d'aide à la décision.
- Améliorer des indicateurs de performances donnés,
- Adapter les décisions à l'évolution de la production.

Pour la mise en place de la méthode d'aide à la décision, nous avons dans un premier temps défini un modèle de données sous forme matricielle appelé table d'évaluation des activités de transport. Cette table avait pour objectif de donner en temps réel l'état du système de transport de l'atelier de production (état des stocks, ressources de transports disponibles etc.) et de prendre des décisions en fonctions de critères précédemment définis.

Les critères que nous avons retenus étaient relatifs à des grandeurs spatiales ou encore temporelles (dates d'arrivée, durées de production, distance parcourue). Mais nous avons aussi défini un critère appelé entropie de la table d'évaluation des activités de transport dont le but était de mesurer le niveau de difficulté du problème de transport à chaque instant de décision.

Après avoir défini le modèle de données et les critères de décision, nous avons proposé un algorithme de gestion du transport basée principalement sur l'entropie de la table des évaluation des activités de transport. Cet algorithme avait pour originalité de modifier les critères de décision en fonction de la situation de production.

Nous avons comparé notre algorithme d'aide à la décision à un algorithme dans lequel la priorité des critères de décision sont figés tout au long de la production. Il ressort de cette comparaison que le fait de figer l'ordre des critères de priorité peut apporter des résultats satisfaisants, mais uniquement à condition de bien choisir les critères prioritaires. En revanche nous sommes rendus compte en utilisant notre méthode basée sur l'entropie on avait des résultats qui moyennement étaient plus intéressants. Nous pouvons d'ores et déjà présenter un avantage à de la méthode que nous proposons. En effet, l'utilisation

de la méthode basée sur l'entropie permet d'éviter une phase d'étude ou d'arbitrage sur le caractère prioritaire d'un critère par rapport à un autre ; En effet dans le cas d'étude que nous avons proposé, nous n'avons que 5 critères de décision ce qui nous donnait déjà 120 manières de classer les critères par ordre de priorité qu'en serait-il si on avait 10 ? le temps d'étude serait encore plus long.

En conclusion, nous pensons que même si notre nos modèles restent acceptables et qu'il permettent de simuler certains types de systèmes de production flexibles, il reste tout de même de nombreux points que nous aurions souhaité abordé notamment :

- L'utilisation de modèles agents pour gérer le problème d'encombrement et d'ergonomie des machines
- Une étude approfondie de la stabilité de notre méthode d'aide à la décision et notamment lorsque l'on a un très grand nombre de produits ou encore lorsque l'on a des variations brusque de volume ou même des pannes.

Pour terminer nous pensons que nos modèles conviennent très bien si l'on souhaite faite du pré-dimensionnement d'un atelier de production, observer par exemple des problèmes de gestion de ressources, des saturations de machines, les effets d'une panne ou bien tester des stratégies d'ordonnancement.

# ANNEXES

## Fonctions du Réseau de Pétri Modulaire

### Fonction garde1

```

fun garde1 (c1 :REAL, c2 :REAL, stock_IO1 :STOCK_IO, stock_IO2 :STOCK_IO
)=
)
if (taille_stock (stock_IO1) > 0.0) andalso (taille_stock (
    stock_IO1) < c1) andalso (taille_stock (stock_IO2) < c2)
then true
else false

```

### Fonction *Stock\_prev\_change*

```

fun Stock_prev_change (s1 :STOCK_IO, n :INT) :STOCK_IO=
let
val stock = (#1 s1)
val typestock = (#2 s1)
val liste_produit = (#1 stock)
val nom = (#1 (#2 stock))
val position_stock = (#2 (#2 stock))
val capacite = (#3 (#2 stock))
val stock_prev = n + length(liste_produit)
in
(
( (liste_produit), ((nom, position_stock, capacite, stock_prev))) ,
typestock
)
end

```

### ReponseStock

```

fun ReponseStock (s1 :STOCK_IO, s2 :STOCK_IO) =
if IdentifierStock (s1, s2)
then ReturnStock (s2)
else s1

```

**fonction *PreparationMachine***

```
fun PreparationMachine(stock_IO1:STOCK_IO, list_tool:ListTool)=  
1`produit (hd (#1 (#1 stock_IO1))) ++  
1`outils (Outil_courant(hd (#1 (#1 stock_IO1)), list_tool)) ++  
1`change_time (Temps_changement_outil(hd (#1 (#1 stock_IO1)),  
list_tool))
```

**fonction *ModificationInfoMachine***

```
fun ModificationInfoMachine(stock_IO1:STOCK_IO)=  
if (taille_stock (stock_IO1)=0.0)  
then stock_IO1  
else Stock_prev_change(ReturnStock (stock_IO1), 0)
```

**fonction *CheckMachine***

```
fun CheckMachine(item:ITEM, s:STRING):BOOL=  
let  
val a=(#Routing item)  
val b=hd a  
val c=(#1 b)  
val d=s  
in  
if (d=c)  
then true  
else false  
end;
```

**fonction *EntreeProduitMachine***

```
fun EntreeProduitMachine(s:STRING, item:ITEM, t:TIME, n:INT)=  
1`Name s ++  
1`Product (Transformation_produit(item, t, n))
```

**fonction *garde2***

```

fun garde2(item:ITEM, a1:A, n:INT, t:TIME)=
  [t=(#due item), (#id item)=n, (#state item)=a1]

```

**fonction *liberationMachine***

```

fun liberationMachine(pos:RxR)=
  1` Available true++
  1` position pos

```

**fonction *ModificationInfoMachine***

```

fun ModificationInfoMachine(stock_IO1:STOCK_IO)=
  if (taille_stock (stock_IO1)=0.0)
  then stock_IO1
  else Stock_prev_change(ReturnStock (stock_IO1)
    ,0)

```

**fonction *Ajouter\_produit\_stock***

```

fun Ajouter_produit_stock(i:ITEM, s:STOCK):STOCK=
  let
  val Nom_stock= #1 (#2 s)
  val position_stock= #2 (#2 s)
  val capacite_stock=(#3 (#2 s))
  val produit_en_stock= #1 s
  in
  ([i]^produit_en_stock, (Nom_stock, position_stock,
    capacite_stock))
  end

```

**Fonction r1**

```

fun r1(a:STOCK_IO, b:STOCK_IO):STOCK_IO=Retour_info_ss(a, b, 1)

```

# Annexe A

## Autres fonctions développées

```
fun VerifStock(st:STOCK):BOOL=
if ( (length (#1 st))=0)
then false
else true

(
*****
*****FIN DU PROGRAMME*****
*****
)
fun CheckStocks [] =0
| CheckStocks(st>ListSTOCK):INT=
let
val tab=st
val s= ref 0
val j= ref ((length st) -1)
in
while ((!j>0) andalso VerifStock(List.nth(tab,(!j))))=false )
do
(
j:=(!j)-1;
s:= (!s)+1
);
!s
end

(
*****
*****FIN DU PROGRAMME*****
*****
)
fun CmpStock (SOut:STOCK_IO, SIn:STOCK_IO):BOOL=
let
val s1= #1 SIn
```



---

```

val s2= #1 SOut
val a= hd (#1 s2)
val b= hd (#Routing a)
val c= (#1 b)
  val d= (#2 s1)
val e=(#1(#2 s1))
in
if (c=e) andalso ((#2 SIn)="In") andalso ((#2 SOut)="Out")
then true
else false
end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun ReturnStock (st:STOCK_IO): STOCK_IO =
( ((tl (#1 (#1 st))),(#2 (#1 st))),#2 st)

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun type_stock(s:STOCK_IO):STRING= #2 s

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun distance_stock_transport(Tr:Transport ,s:STOCK_IO):REAL=
let
val position_transport= #position Tr
val position_stock= #2 (#2 (#1 s))
val abscisse= (#1 position_transport) - (#1 position_stock)
val ordonne= (#2 position_transport) - (#2 position_stock)
in
Math.sqrt(Math.pow(abscisse ,2.0)+Math.pow(ordonne ,2.0))
end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun taille_stock(s:STOCK_IO):REAL
= Real.fromInt( (length (#1 (#1 s))))

```

```
(*****  
*****FIN DU PROGRAMME*****  
*****)  
  
fun Position_Stock (s:STOCK_IO):RxR=(#2 (#2 (#1 s)))  
  
(*****  
*****FIN DU PROGRAMME*****  
*****)  
  
fun recherche_stock_entree(s:STOCK_IO, ls:ListSTOCK_IO):STOCK_IO=  
let  
val j= ref ((length ls)-1)  
in  
while (!j>=0) andalso (CmpStock(List.nth(ls,!j),s)=false)  
do  
(  
j:=(!j)-1  
);  
List.nth(ls,!j)  
end  
  
(*****  
*****FIN DU PROGRAMME*****  
*****)  
  
fun Definir_critere_stock(s:STOCK_IO): Critere=  
let  
val stock=(#1 s)  
val nom_du_stock= #1 (#2 stock)  
in  
(nom_du_stock, [])  
end  
  
(*****  
*****FIN DU PROGRAMME*****  
*****)  
  
fun Nom_stock (s:STOCK_IO):STRING=(#1 (#2 (#1 s)))  
  
(*****  
*****FIN DU PROGRAMME*****  
*****)  
  
fun recherche_stock(c: Critere, ls:ListSTOCK_IO):STOCK_IO=  
let
```

---

```

val nom_du_stock= #1 c
val j= ref ((length ls)-1)
in
while (!j>=0) andalso (nom_du_stock <> Nom_stock(List.nth(ls,!j)))
do
(
j:= (!j)-1
);
List.nth(ls,!j)
end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun taillestockcmp (st:STOCK_IO):ListSTOCK_IO=
if (taille_stock(st)=0.0)
then []
else [st]

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun filtre_stock_vide(ls>ListSTOCK_IO):ListSTOCK_IO=
let
val j= ref ((length ls)-1)
val tab= ref []
in
while (!j>=0)
do
(
tab:= taillestockcmp(List.nth(ls,!j)) ^^(!tab);
j:=(!j)-1
);
(!tab)
end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun IdentifierStock(s1:STOCK_IO,s2:STOCK_IO)=
if(Nom_stock (s1)=Nom_stock (s2))andalso((#2 s1)=(#2 s2))

```

```
then true
else false
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun remplissage_liste_stock_entree(s:STOCK_IO):ListSTOCK_IO=
if type_stock(s)="In"
then [s]
else []
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun remplissage_liste_stock_sortie(s:STOCK_IO):ListSTOCK_IO=
if type_stock(s)="Out"
then [s]
else []
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun ReturnStock (st:STOCK_IO):STOCK_IO= ( ((t1 (#1 (#1 st))),(#2 (#1 st))),)
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun ReponseStock(s1:STOCK_IO,s2:STOCK_IO)=
if IdentifierStock(s1,s2)
then ReturnStock (s1)
else s1
```

```
(* c est la fonction li '\e au transporteur *)
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun Ajouter_produit_stock(i:ITEM,s:STOCK):STOCK=
let
```

---

```

val Nom_stock= #1 (#2 s)
val position_stock= #2 (#2 s)
val capacite_stock =(#3 (#2 s))
val produit_en_stock= #1 s
in
([i]^produit_en_stock ,(Nom_stock , position_stock , capacite_stock))
end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun nieme_element_stock (s:STOCK_IO, n:INT):ITEM=
List.nth ((#1 (#1 s)),n-1)

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun VerifStock_IO (st:STOCK_IO):BOOL=
if ( (length (#1(#1 st)))>0)
then true
else false

```

```

(est ce qu'un stock est vide ou pas)

```

```

(*****
*****FIN DU PROGRAMME*****
*****)

```

```

fun CheckStock_IO [] =0
| CheckStock_IO (st :ListSTOCK_IO):INT=
let
val tab=st
val s= ref 0
val j= ref (length st)
in
while((!j>0) andalso VerifStock_IO(List.nth(tab ,(!j)-1))=false )
do
(
j:=(!j)-1;
s:= (!s)+1
);
if(!s)<(length st)

```

```
then 1
else 0
end
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun RetourStockChoix (stock1:STOCK_IO, Stock_choisi:STOCK_IO):STOCK_IO=
if IdentifierStock(stock1, Stock_choisi)=true
then ReturnStock(stock1)
else stock1
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun Stock_prev_change (s1:STOCK_IO, n:INT):STOCK_IO=
```

```
let
```

```
val stock=(#1 s1)
```

```
val typestock=(#2 s1)
```

```
val liste_produit= (#1 stock)
```

```
val nom= (#1 (#2 stock))
```

```
val position_stock= (#2 (#2 stock))
```

```
val stock_prev=n+(#3 (#2 stock))
```

```
in
```

```
(
( (liste_produit), ((nom, position_stock, stock_prev)) ) ,
typestock
)
```

```
end
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun Retour_info_ss (s1:STOCK_IO, s2:STOCK_IO, n:INT):STOCK_IO=
```

```
let
```

```
val produit_en_tete= nieme_element_stock(s2, 1)
```

```
val Prochaine_machine= NextMachine(produit_en_tete)
```

```
in
```

```
if (Prochaine_machine=Nom_stock(s1))
```

```
then Stock_prev_change(s1, n)
```

```
else s1
```

```
end
```

```
(*****
```

---

```

*****FIN DU PROGRAMME*****
*****)
fun Find_Stock(c:STRING, ls>ListSTOCK_IO):STOCK_IO=
let
  val nomdustock =c
  val j= ref ((length ls)-1)
in
  while(!j>=0) andalso (nomdustock <> Nom_stock(List.nth(ls,!j)))
  do
    (
    j:= (!j)-1
    );
    List.nth(ls,!j)
  end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)
fun TabIn []=[]
|TabIn(st>ListSTOCK_IO):ListSTOCK_IO=
let
  val tab=st
  val stock_entree= ref []
  val s= ref 0
  val j= ref ((length st) -1)
in
  while((!j>=0))
  do
    (
    stock_entree:= remplissage_liste_stock_entree(List.nth(st,!j))^(!stock_e
    j:=(!j)-1
    );
    !stock_entree
  end

```

```

(*****
*****FIN DU PROGRAMME*****
*****)
fun TabOut []=[]
|TabOut(st>ListSTOCK_IO):ListSTOCK_IO=
let
  val tab=st
  val stock_sortie= ref []

```

```
val s= ref 0
val j= ref ((length st) -1)
in
while ((!j >=0))
do
(
stock_sortie:= remplissage_liste_stock_sortie(List.nth(st,(!j)))^(!stock_
j:=(!j)-1
);
!stock_sortie
end
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun ajout_critere(c: Critere, x: REAL): Critere=
let
val id= #1 c
val table_critere= #2 c
val nouvelle_table= table_critere^^[x]
in
(id, nouvelle_table)
end
```

```
(*****
*****FIN DU PROGRAMME*****
*****)
```

```
fun Table_de_priorite ([], Tr: Transport) = []
|
Table_de_priorite(ls: ListSTOCK_IO, Tr: Transport): ListCriteres=
let
val tab_des_entrees = TabIn(ls)
val tab_des_sorties = TabOut(ls)
val tab_sorties_non_vides = filtre_stock_vide(tab_des_sorties)
val j= ref(length(tab_sorties_non_vides)-1)
val k= ref(length(tab_des_entrees)-1)
val Tables_des_choix = ref []
in
while (!j >=0)
do
(
while (!k >=0)
do
(
if (CmpStock((List.nth(tab_des_entrees, !k)), List.nth(tab_sorties_non_vides,
then !Tables_des_choix^^[ (Nom_stock(List.nth(tab_sorties_non_vides, !j)),
```



---

```

[ taille_stock(List.nth(tab_des_entrees,!k)),taille_stock(List.nth(tab_sorties,!j))
else !Tables_des_choix;
k:=(!k)-1
);
j:=(!j)-1
);
(!Tables_des_choix)
end
(*****
*****FIN DU PROGRAMME*****
*****)
fun FillCritere(SIn: STOCK_IO,SOut:STOCK_IO): ListCriteres=
(entree et ensuite sortie)
let
val a= CmpStock(SOut,SIn)
val x= Nom_stock(SOut)
val y=taille_stock(SIn)
val z=taille_stock(SOut)
in
if (a=true)

then [ (x,[y ,z])]
else []
end

(*****
*****FIN DU PROGRAMME*****
*****)
fun ExtCritere(cr: Critere ,n:INT):REAL= List.nth((#2 cr),n)
(*****
*****FIN DU PROGRAMME*****
*****)

fun Table_de_priorites([],Tr:Transport)=[]
| Table_de_priorites (ls:ListSTOCK_IO,Tr:Transport): ListCriteres=
let
val tab_des_entrees = TabIn(ls)
val tab_des_sorties = TabOut(ls)
val tab_sorties_non_vides =filtre_stock_vide(tab_des_sorties)
val j= ref(length(tab_sorties_non_vides))
val k= ref(length(tab_des_entrees))
val Tables_des_choix = ref []
val counter= ref (0)
in
while (!j>0)
do

```

```
(
  k:= (length(tab_des_entrees));
  while (!k>0)
    do
      (
        Tables_des_choix:=FillCritere( List.nth(tab_des_entrees,
        k:=(!k)-1
      );
      j:=(!j)-1;
      k:= (length(tab_des_entrees))
    );
  (!Tables_des_choix)
end
```

# Bibliographie Personnelle

- [1] J. Eloundou, D. Baudry, A. Bensrhair, A. Louis, B. Mazari, *et al.*, “Flexible job shop models for solving scheduling and layout problems using coloured petri nets,” in *Applied Mechanics and Materials*, vol. 598, pp. 638–642, Trans Tech Publ, 2014.
- [2] J. Eloundou, M. Sahnoun, A. Louis, D. Baudry, and A. Bensrhair, “Evaluation of the routing flexibility of a manufacturing system,” in *Xème Conférence Internationale : Conception et Production Intégrées*, 2015.
- [3] J. ELOUNDOU, M. Sahnoun, D. Baudry, A. Bensrhair, and A. Louis, “Variability of production in flexible manufacturing systems :petri net modelling and simulation,” in *Xème Conférence Internationale : Conception et Production Intégrées*, (Tanger, Morocco), Dec. 2015.

# Bibliographie

- [1] Y. o. M. Koren, U. o. S. Heisel, F. d. M. Jovane, T. U. Moriwaki, G. o. S. Pritschow, G. o. M. Ulsoy, and H. U. leuven) Van Brussel, “Reconfigurable Manufacturing Systems,” *CIRP Annals-manufacturing technology*, vol. 48, no. 2, pp. 527–540, 1999.
- [2] J. W. Fowler and O. Rose, “Grand challenges in modeling and simulation of complex manufacturing systems,” *Simulation*, vol. 80, no. 9, pp. 469–476, 2004.
- [3] K. Tamani, “Développement d’une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production,” *Thèse de Doctorat*, 2008.
- [4] A. Bratcu, “Détermination systématiques des graphes de précedence et équilibrage des lignes d’assemblage,” *Thèse de Doctorat*, 2001.
- [5] G. Javel, *Organisation et gestion de la production-4e édition : Cours, exercices et études de cas*. Dunod, 2010.
- [6] V. Giard, “Gestion de le production et des flux ; production et techniques quantitatives appliquées à la gestion ; ed,” *Economica*, 2003.
- [7] A. Courtois, C. Martin-Bonnefous, M. Pillet, and M. Pillet, *Gestion de production*. Les Ed. d’Organisation, 2003.
- [8] R. Biteau and S. Biteau, *Maîtriser les flux industriels : les outils d’analyse*. Ed. d’Organisation, 1998.
- [9] L. Gu, “Modèles Déterministe , Stochastique et Multicritère pour l’ Equilibrage de Lignes d’ Assemblage,” *Thèse de doctorat*, 2008.
- [10] Y. Koren and M. Shpitalni, “Design of reconfigurable manufacturing systems,” *Journal of Manufacturing Systems*, vol. 29, pp. 130–141, Oct. 2011.
- [11] Y. Koren, “General RMS characteristics. Comparison with dedicated and flexible systems,” in *Reconfigurable manufacturing systems and transformable factories*, pp. 27–45, Springer, 2006.
- [12] A. Sethi and S. Sethi, “Flexibility in manufacturing : A survey,” *International Journal of Flexible Manufacturing Systems*, vol. 2, pp. 289–328, July 1990.
- [13] J. Browne, D. Dubois, K. Rathmill, S. P. Sethi, and K. E. Stecke, “Classification of flexible manufacturing systems,” *The FMS magazine*, vol. 2, no. 2, pp. 114–117, 1984.

- [14] V. Kumar, “Entropic measures of manufacturing flexibility,” *International Journal of Production Research*, vol. 25, no. 7, pp. 957–966, 1987.
- [15] D. Gerwin, “An agenda for research on the flexibility of manufacturing processes,” *International Journal of Operations & Production Management*, vol. 7, no. 1, pp. 38–49, 1987.
- [16] F. Carter, M., “Designing flexibility into automated manufacturing systems,” in *Proc. 2nd ORSA/TIMS Conference on Flexible Manufacturing systems :OR Models and Applications*, Ann Arbor, pp. 107–118, 1986.
- [17] M. Larabi, *Le problème de job-shop avec transport : modélisation et optimisation*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2010.
- [18] P. Baptiste, M. Flamini, and F. Sourd, “Lagrangian bounds for just-in-time job-shop scheduling,” *Computers & Operations Research*, vol. 35, no. 3, pp. 906–915, 2008.
- [19] Z. Guo, W. Wong, S. Leung, J. Fan, and S. Chan, “Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment : A case study based on the apparel industry,” *Computers & Industrial Engineering*, vol. 50, pp. 202–219, July 2006.
- [20] A. Scholl and S. Voß, “Simple assembly line balancing heuristic approaches,” *Journal of Heuristics*, vol. 2, no. 3, pp. 217–244, 1997.
- [21] J. Carlier, “Ordonnements a contraintes disjonctives,” *RAIRO-Operations Research-Recherche Opérationnelle*, vol. 12, no. 4, pp. 333–350, 1978.
- [22] J. Carlier and É. Pinson, “An algorithm for solving the job-shop problem,” *Management science*, vol. 35, no. 2, pp. 164–176, 1989.
- [23] A. Manceaux, H. B. El-Haouzi, A. Thomas, J.-F. Péting, and T. Munier, “Reconfiguration dynamique d’une ligne d’assemblage par recherche d’atteignabilité sur un réseau d’automates communicants,” in *10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM’14*.
- [24] M. Mabed, F. Bousseau, and M. Bourcerie, “Modélisation d’une ligne de galvanoplastie par les rdpt/p : cas multi-produits,” in *MOSIM’01, 3e Conférence Francophone de Modélisation et Simulation*, p. xx, 2001.
- [25] M. S. Anand and S. K. Sindhe, “Modeling and simulation of job shop scheduling using petri-nets,” *International Journal of Enginnering Research and Applications*, vol. 3, no. 2, pp. 492–496, 2013.
- [26] M. Piera and G. Mušič, “Coloured Petri net scheduling models : Timed state space exploration shortages,” *Mathematics and Computers in Simulation*, vol. 82, pp. 428–441, Nov. 2011.
- [27] J. P. Caballero-Villalobos, G. E. Mejía-Delgadillo, and R. G. García-Cáceres, “Scheduling of complex manufacturing systems with Petri nets and genetic algorithms : a case on plastic injection moulds,” *The International Journal of Advanced Manufacturing Technology*, vol. 69, pp. 2773–2786, Aug. 2013.

- [28] A. Mhalla and B. Mohamed, "Contribution to the Maintenance of Manufacturing Systems with Time Constraints Using Fuzzy Petri Nets," *Proceedings of the International Conference on Control, Engineering & Information Technology (CEIT'14)*, pp. 297–309, 2014.
- [29] H. Haleh, A. Bahari, and B. Moody, "Performance Analysis of Manufacturing Systems Using Deterministic and Stochastic Petri Nets," vol. 11, pp. 1–12, 2014.
- [30] C. Girault and J.-F. Pradat-Peyre, "Les réseaux de petri de haut-niveau," *Les réseaux de Petri*, 2001.
- [31] M. Barad and D. Sipper, "Flexibility in manufacturing systems : definitions and petri net modelling," *The International Journal Of Production Research*, vol. 26, no. 2, pp. 237–248, 1988.
- [32] J. Ezpeleta, J. M. Colom, and J. Martinez, "A petri net based deadlock prevention policy for flexible manufacturing systems," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 2, pp. 173–184, 1995.
- [33] M. Zhou and K. Venkatesh, *Modeling, simulation, and control of flexible manufacturing systems : a Petri net approach*, vol. 6. World Scientific, 1999.
- [34] C. Girault and R. Valk, *Petri nets for systems engineering : a guide to modeling, verification, and applications*. Springer Science & Business Media, 2013.
- [35] O. Kilincci and G. M. Bayhan, "A petri net approach for simple assembly line balancing problems," *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 11-12, pp. 1165–1173, 2006.
- [36] J. Eloundou, D. Baudry, A. Bensrhair, A. Louis, B. Mazari, *et al.*, "Flexible job shop models for solving scheduling and layout problems using coloured petri nets," in *Applied Mechanics and Materials*, vol. 598, pp. 638–642, Trans Tech Publ, 2014.
- [37] P. Thomas, "Petri net control of a kanban loop To cite this version :," 2011.
- [38] K. Saitou, S. Malpathak, and H. Qvam, "Robust design of flexible manufacturing systems using, colored petri net and genetic algorithm," *Journal of intelligent manufacturing*, vol. 13, no. 5, pp. 339–351, 2002.
- [39] X. Meng, "Modeling of reconfigurable manufacturing systems based on colored timed object-oriented Petri nets," *Journal of Manufacturing Systems*, vol. 29, pp. 81–90, July 2010.
- [40] M. Zhou and F. DiCesare, *Petri net synthesis for discrete event control of manufacturing systems*, vol. 204. Springer Science & Business Media, 2012.
- [41] a. P. A. TIANLONG GU and G. CAI., "Timed petri-net based formulation and an algorithm for the optimal scheduling of batch plants," *International Journal Applied Mathematics and computer sciences*, vol. 13, no. 4, pp. 527–536, 2003.

- [42] B. Sharda and A. Banerjee, “Robust manufacturing system design using multi objective genetic algorithms, petri nets and bayesian uncertainty representation,” *Journal of Manufacturing Systems*, vol. 32, no. 2, pp. 315–324, 2013.
- [43] P. David Stotts, R. W. Newcomb, and Z. Ning Cai, “Modelling the logical structure of flexible manufacturing systems with Petri-nets,” *Computer Communications*, vol. 12, pp. 193–203, Aug. 1989.
- [44] H. Thomas W, “Colored Petri Net Modeling of Metal Distribution in a Job Shop Iron Foundry : Improving Flow in a Pull System with Perishable In-Process Inventory,” pp. 66–77, 2003.
- [45] A. da Silva Ribeiro, E. M. M. Costa, and E. J. Lima II, “Flexible manufacturing systems modelling using high level petri nets,” 2008.
- [46] T. Aized, “Modelling and performance maximization of an integrated automated guided vehicle system using coloured Petri net and response surface methods,” *Computers & Industrial Engineering*, vol. 57, pp. 822–831, Oct. 2009.
- [47] P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin, “Petri nets compositional modeling and verification of Flexible Manufacturing Systems,” *2011 IEEE International Conference on Automation Science and Engineering*, pp. 588–593, Aug. 2011.
- [48] H. Hosseini-nasab and A. Sadri, “Using Stochastic Colored Petri Nets for Designing Multi-Purpose Plants,” vol. 2012, no. October, pp. 655–661, 2012.
- [49] B. Barzegar, H. Motameni, and H. Bozorgi, “Solving Flexible Job-Shop Scheduling Problem Using Gravitational Search Algorithm and Colored Petri Net,” *Journal of Applied Mathematics*, vol. 2012, pp. 1–20, 2012.
- [50] F. G. Quintanilla, O. Cardin, A. L’Anton, and P. Castagna, “A petri net-based methodology to increase flexibility in service-oriented holonic manufacturing systems,” *Computers in Industry*, vol. 76, pp. 53–68, 2016.
- [51] Y. Li, W. Wu, S. Wang, and C. Wang, “A literature review of algorithms to compute minimal siphons based on petri nets,” *International Journal of Advancements in Computing Technology*, vol. 5, no. 1, 2013.
- [52] Z. Li, N. Wu, and M. Zhou, “Deadlock control of automated manufacturing systems based on petri nets—a literature review,” *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, vol. 42, no. 4, pp. 437–462, 2012.
- [53] K. Jensen, *Coloured Petri nets : basic concepts, analysis methods and practical use*, vol. 1. Springer Science & Business Media, 2013.
- [54] G. Fleury, P. Lacomme, and A. Tanguy, *Simulation à événements discrets*. Eyrolles, 2006.

- [55] E. Yücesan and J. W. Fowler, “Logistics systems, simulation analysis of simulation analysis of manufacturing and logistics systems,” in *Encyclopedia of production and manufacturing management*, pp. 687–697, Springer, 2000.
- [56] Y. Kuo, T. Yang, C. Cho, and Y.-C. Tseng, “Using simulation and multi-criteria methods to provide robust solutions to dispatching problems in a flow shop with multiple processors,” *Mathematics and Computers in Simulation*, vol. 78, no. 1, pp. 40–56, 2008.
- [57] H.-H. Huang, W. Pei, H.-H. Wu, and M.-D. May, “A research on problems of mixed-line production and the re-scheduling,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 3, pp. 64–72, 2013.
- [58] L.-N. Xing, Y.-W. Chen, and K.-W. Yang, “Multi-objective flexible job shop schedule : design and evaluation by simulation modeling,” *Applied Soft Computing*, vol. 9, no. 1, pp. 362–376, 2009.
- [59] L. Zhuo, D. Chua Kim Huat, and K. H. Wee, “Scheduling dynamic block assembly in shipbuilding through hybrid simulation and spatial optimisation,” *International Journal of Production Research*, vol. 50, no. 20, pp. 5986–6004, 2012.
- [60] J. ELOUNDOU, M. Sahnoun, D. Baudry, A. Bensrhair, and A. Louis, “Variability of production in flexible manufacturing systems :petri net modelling and simulation,” in *Xème Conférence Internationale : Conception et Production Intégrées*, (Tanger, Morocco), Dec. 2015.
- [61] M. Gourgand, N. Grangeon, and S. Norre, “A contribution to the stochastic flow shop scheduling problem,” *European Journal of Operational Research*, vol. 151, no. 2, pp. 415–433, 2003.
- [62] F. Dugardin, F. Yalaoui, and L. Amodeo, “New multi-objective method to solve reentrant hybrid flow shop scheduling problem,” *European Journal of Operational Research*, vol. 203, no. 1, pp. 22–31, 2010.
- [63] G.-C. Lee, “Estimating order lead times in hybrid flowshops with different scheduling rules,” *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1668–1674, 2009.
- [64] V. Vinod and R. Sridharan, “Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system,” *International Journal of Production Economics*, vol. 129, no. 1, pp. 127–146, 2011.
- [65] O. Joseph and R. Sridharan, “Effects of flexibility and scheduling decisions on the performance of an fms : simulation modelling and analysis,” *International Journal of Production Research*, vol. 50, no. 7, pp. 2058–2078, 2012.
- [66] L. W. Schruben and T. M. Roeder, “Fast simulations of large-scale highly congested systems,” *Simulation*, vol. 79, no. 3, pp. 115–125, 2003.



- [67] G. Deleuze, N. Brinzei, and T. Hodicq, “Estimation of common cause failures probabilities in control systems by colored, hierarchical petri nets,” in *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFE-PROCESS’2015*, 2015.
- [68] K. Negrichi, *Approche intégrée pour l’analyse de risques et l’évaluation des performances : application aux services de stérilisation hospitalière*. PhD thesis, Université Grenoble Alpes, 2015.
- [69] T. Berger, D. Deneux, T. Bonte, E. Cocquebert, and D. Trentesaux, “Arezzo-flexible manufacturing system : A generic flexible manufacturing system shop floor emulator approach for high-level control virtual commissioning,” *Concurrent Engineering*, vol. 23, no. 4, pp. 333–342, 2015.
- [70] M. M. Tseng and S. J. Hu, “Mass customization,” in *Cirp Encyclopedia of Production Engineering*, pp. 836–843, Springer, 2014.
- [71] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, A. Marino, and D. Gerbasio, “Automated synthesis of hybrid petri net models for robotic cells in the aircraft industry,” *Control Engineering Practice*, vol. 31, pp. 35–49, 2014.
- [72] <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html>.
- [73] K. Jensen, L. M. Kristensen, and L. Wells, “Coloured petri nets and cpn tools for modelling and validation of concurrent systems,” *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, 2007.
- [74] A. W. Appel and D. B. MacQueen, “Standard ml of new jersey,” in *Programming Language Implementation and Logic Programming*, pp. 1–13, Springer, 1991.
- [75] R. Milner, *The definition of standard ML : revised*. MIT press, 1997.
- [76] M. Dahane, M. Sahnoun, B. Bettayeb, D. Baudry, and H. Boudhar, “Impact of spare parts remanufacturing on the operation and maintenance performance of offshore wind turbines : a multi-agent approach,” *Journal of Intelligent Manufacturing*, pp. 1–19, 2015.
- [77] R. G. Sargent, “Verification and validation of simulation models,” *Journal of simulation*, vol. 7, no. 1, pp. 12–24, 2013.
- [78] R. W. Pew and S. Baron, “Perspectives on human performance modelling,” *Automatica*, vol. 19, no. 6, pp. 663–676, 1983.
- [79] B. P. Zeigler, *Object-oriented simulation with hierarchical, modular models : intelligent agents and endomorphic systems*. Academic press, 2014.
- [80] R. G. Sargent, “Model verification and validation,” in *Modeling and Simulation in the Systems Engineering Life Cycle*, pp. 57–65, Springer, 2015.
- [81] J. Gattorna, *Dynamic supply chains*. Pearson Education Limited, 2015.

- [82] S. B. Mena, "Introduction aux méthodes multicritères d'aide à la décision," *Biotechnologie, Agronomie, Société et Environnement*, vol. 4, no. 2, pp. 83–93, 2000.
- [83] C. Sourisse, "L'élaboration des cahiers des charges des équipements de production industriels : La transitique : automatiser flux et procédés," *REE. Revue de l'électricité et de l'électronique*, no. 6, pp. 18–22, 1995.
- [84] O. Devise, "Modélisation orientée objet de la transitique dans un environnement fms," *Revue d'automatique et de productique appliquées*, vol. 8, no. 2-3, pp. 479–484, 1995.
- [85] M. Larabi, *Job-shop with transport : its modelling and optimisation*. Theses, Université Blaise Pascal - Clermont-Ferrand II, Dec. 2010.
- [86] P. Vincke, B. Roy, *et al.*, *L' aide multicritère à la décision*. Ed. de l' Université de Bruxelles [etc., 1989.
- [87] Y. Collette and P. Siarry, *Optimisation multiobjectif : Algorithmes*. Editions Eyrolles, 2011.
- [88] J. Figueira, V. Mousseau, and B. Roy, "Electre methods," in *Multiple criteria decision analysis : State of the art surveys*, pp. 133–153, Springer, 2005.
- [89] R. Caillet *et al.*, "Analyse multicritère : Étude de comparaison des méthodes existantes en vue d'une application en analyse de cycle de vie," tech. rep., CIRANO, 2003.
- [90] Ü. Bilge, M. Firat, and E. Albey, "A parametric fuzzy logic approach to dynamic part routing under full routing flexibility," *Computers & Industrial Engineering*, vol. 55, no. 1, pp. 15–33, 2008.
- [91] F. Zammori, M. Braglia, and M. Frosolini, "A measurement method of routing flexibility in manufacturing systems," *International Journal of Industrial Engineering Computations*, vol. 2, no. 3, pp. 593–616, 2011.
- [92] O. Joseph and R. Sridharan, "Evaluation of routing flexibility of a flexible manufacturing system using simulation modelling and analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 56, no. 1-4, pp. 273–289, 2011.
- [93] A. Chang, "On the measurement of labor flexibility," in *Engineering Management Conference, 2004. Proceedings. 2004 IEEE International*, vol. 1, pp. 163–167, IEEE, 2004.
- [94] A.-Y. Chang, "On the measurement of routing flexibility : a multiple attribute approach," *International Journal of Production Economics*, vol. 109, no. 1, pp. 122–136, 2007.
- [95] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European journal of operational research*, vol. 2, no. 6, pp. 429–444, 1978.