

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abou Bekr Belkaïd, Tlemcen
Faculté des Sciences de l'Ingénieur

Thèse de Doctorat en Automatique-Productique

Par

Khalid HACHEMI

Titre :

**PILOTAGE DYNAMIQUE D'UN SYSTEME
AUTOMATISE DE STOCKAGE/DESTOCKAGE A
CONVOYEUR GRAVITATIONNEL**

Thèse soutenue le 04 mai 2008

Jury :

Président : Abdelmadjid CHEIKH Professeur UABB Tlemcen, Algérie

Rapporteurs : Noureddine GHOUALI Professeur UABB Tlemcen, Algérie

Zaki SARI Maître de Conférences UABB Tlemcen, Algérie

Examineurs : Hassane ALLA Professeur Gipsa-Lab - UJF Grenoble, France

Hafid HAFFAF Professeur U. Oran, Algérie

Guy JUANOLE Professeur Emérite LAAS- UPS Toulouse, France

Remerciements

Je tiens tout d'abord à remercier vivement mon directeur de thèse Monsieur Noureddine Ghouali, Professeur à l'Université de Tlemcen pour m'avoir permis de découvrir et de travailler sur un domaine aussi passionnant que l'Automatique-Productique. En me proposant ce thème et en m'impliquant dans plusieurs projets de recherche nationaux au sein du Laboratoire d'Automatique de Tlemcen, il m'a témoigné d'une grande confiance dont je lui suis très reconnaissant.

Je tiens à exprimer toute ma gratitude à Monsieur Zaki Sari, Maître de Conférences à l'Université de Tlemcen, qui a co-encadré ces travaux. Tout au long de mon parcours de recherche, Il a toujours su me transmettre son enthousiasme et son dynamisme me permettant ainsi d'aborder le sujet avec moins d'appréhension. Je le remercie chaleureusement pour son soutien, sa disponibilité et son amitié.

L'aboutissement de ce travail de recherche a été rendu possible grâce au soutien de Monsieur Hassane Alla, Professeur à l'Université Joseph Fourier et chercheur au département d'Automatique de Grenoble (ex. LAG), qui a eu la gentillesse de m'accueillir dans son laboratoire. Je tiens à le remercier vivement pour les conditions de travail exceptionnelles qu'il a mis à ma disposition, pour s'être toujours rendu disponible et pour ses remarques et réflexions qui ont considérablement contribué à ces travaux. Je le remercie encore une fois pour avoir accepté d'examiner ce travail.

Je tiens à exprimer ma sincère gratitude à Monsieur Abdelmadjid Cheikh, Professeur à l'Université de Tlemcen pour l'honneur qu'il m'a fait en s'intéressant à ce travail et en acceptant de présider mon jury.

Je tiens également à remercier chaleureusement Monsieur Hafid Haffaf, Professeur à l'Université d'Oran pour le plaisir qu'il m'a fait en acceptant d'examiner ce travail et l'intérêt qu'il y a porté.

J'exprime mes sincères remerciements à Monsieur Guy Juanole, Professeur Emérite à l'Université Paul Sabatier et chercheur au Laboratoire d'Architecture et d'Analyse des Systèmes de Toulouse (LAAS) pour l'honneur qu'il m'a fait en acceptant de faire partie du jury.

Je tiens à exprimer ma profonde gratitude à Monsieur Ahmed Boudjani, Maître de Conférences à l'Université d'Oran, pour m'avoir impliqué dans la post-graduation Automatique, option "Contrôle de Processus Industriels et de Laboratoire" dont il était le directeur. Cela m'a permis de faire progresser mes recherches, grâce à une ambiance stimulante, et des discussions fructueuses et sans lesquels plusieurs idées n'auraient sans doute jamais vu le jour. Je lui renouvelle toute ma reconnaissance.

Je tiens également à exprimer toute ma gratitude à Madame Béatrix Besombes, Maître de Conférences à l'Université de Saint-Etienne (IUT de Roanne) pour m'avoir accueilli et permis

d'effectuer une bonne partie de mes travaux de recherche dans le Laboratoire d'Analyse de Signaux et de Processus Industriels, LASPI dans d'excellentes conditions.

Je remercie sincèrement Monsieur Eric Marcon, Professeur à l'Université de Saint-Etienne (IUT de Roanne) et responsable de l'équipe Génie Industriel au sein du LASPI qui m'a permis de trouver une orientation vers une application hospitalière de mon travail de recherche. C'était un axe nouveau pour moi et il n'a pas cessé de me faire participer à des journées d'études dans ce domaine, sans oublier ses précieuses remarques et recommandations.

Je souhaite exprimer mes sincères remerciements à Monsieur Yahia Lebbah, Maître de Conférences à l'Université d'Oran pour la collaboration fructueuse que nous avons eu et pour ses idées précieuses qui m'ont été d'un grand aide.

Mes remerciements vont également à toutes les personnes qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Pour finir, je souhaite remercier ma famille pour son soutien et ses encouragements tout au long de la réalisation de ce travail.

Introduction générale

L'environnement actuel des entreprises est caractérisé par des marchés soumis à une forte concurrence et sur lesquels les exigences et les attentes des clients deviennent de plus en plus fortes en termes de quantités, coûts et délais de livraison. Dans un tel contexte, la performance de l'entreprise se construit sur toute la chaîne logistique (*Supply chain, en anglais*) qui est le système grâce auquel les entreprises amènent leurs produits et leurs services jusqu'à leur clients. Les entrepôts de stockage constituent l'un des maillons clés de la performance de la chaîne logistique. En effet, l'amélioration du taux de service dans l'entrepôt de stockage au niveau des centres de distribution conduit à une meilleure réponse aux demandes de clients.

Pour faire face à une demande de clients croissante, et d'y répondre rapidement, les systèmes automatisés de stockage/déstockage (Automated Storage and Retrieval Systems, AS/RS) sont d'un grand apport. Un AS/RS est composé typiquement de racks de stockage, machines de stockage et déstockage, convoyeurs et de stations d'entrée/sortie où la machine de stockage/déstockage peut charger et décharger des produits. Il existe plusieurs types d'AS/RS, tels que les AS/RS à charge unitaire, les AS/RS à mini-charge et les AS/RS à convoyeur gravitationnel.

Cependant, les AS/RS utilisés en production, entrepôts et en distribution doivent être conçus de telle sorte qu'ils fournissent des temps de réponses rapides pour satisfaire efficacement la demande des clients. De ce fait, la performance principale recherchée dans les AS/RS consiste en la réduction des temps de transport de la machine de stockage/déstockage et ainsi l'amélioration du débit de sortie du système (Randhawa *et al*, 1991), c'est-à-dire le nombre de produits déstockés par unité de temps. Deux grandes approches sont utilisées pour améliorer le débit de sortie du système. La première tente d'agir sur la configuration du point d'entrée/sortie et du point de repos de la machine pour réduire le temps moyen de transport. La deuxième approche s'y prend différemment, elle s'adresse à l'ordonnancement des opérations de stockage/déstockage en vue de trouver une succession d'opérations permettant de réduire le temps de réponse du système.

Le type d'AS/RS le plus étudié est l'AS/RS à charge unitaire qui est considéré comme le modèle générique des autres. Dans ce système les produits sont stockés dans des emplacements spécifiques et qui peuvent être déstockés directement sur la face avant du rack.

Par ailleurs, lorsqu'on tente d'étudier l'AS/RS à convoyeur gravitationnel, objet de ce travail, on se confronte rapidement à une différence fondamentale avec l'AS/RS à charge unitaire, limitant ainsi l'exploitation des résultats déjà trouvés pour ce dernier. En effet, il s'agit du fait que dans l'AS/RS à convoyeur gravitationnel, les produits sont stockés en profondeur dans des casiers à plusieurs emplacements. Ainsi, pour récupérer un produit qui se trouve au milieu d'un casier, il faut déstocker, un à un tous les produits qui le précèdent et les envoyer vers le convoyeur de restockage pour être réintroduit dans le rack.

Par conséquent, il ressort que contrairement à l'AS/RS à charge unitaire, pour déstocker un produit demandé dans un AS/RS à convoyeur gravitationnel, la machine de déstockage peut nécessiter plus d'un seul cycle. Cela nous a amené à considérer cette notion de restockage qui est très coûteuse en temps, en se fixant l'objectif de réduire le nombre de cycles de restockage.

Dans ce travail, nous traitons le problème de pilotage dynamique de l'AS/RS à convoyeur gravitationnel, en ayant comme objectif l'ordonnancement optimal des opérations de déstockage satisfaisant la demande de clients tout en réalisant le moins de cycles de déstockage possible. Pour ce faire, nous développerons un modèle réseau de Pétri coloré capable de décrire la dynamique de l'AS/RS à convoyeur gravitationnel.

Ce mémoire s'articulera autour de cinq chapitres.

Dans le premier chapitre, nous donnons une présentation détaillée des systèmes automatisés de stockage/déstockage. Nous commençons par la description d'un AS/RS avec ses différents composants et nous décrivons ensuite le fonctionnement des machines de stockage/déstockage. Les avantages des AS/RS sont ensuite traités. Par la suite, nous faisons une classification des politiques de stockage utilisées dans la pratique. L'évaluation de performances des AS/RS repose sur les modèles des temps de transport de la machine de stockage/déstockage, de ce fait nous avons réservée une section au calcul de ces modèles. Nous réalisons ensuite une étude bibliographique en dressant un état de l'art des problématiques étudiées dans le domaine des AS/RS et nous nous focaliserons sur les motivations qui nous ont incités à s'intéresser au problème de pilotage de l'AS/RS à convoyeur gravitationnel.

Dans le deuxième chapitre, les réseaux de Pétri sont introduit comme un outil de modélisation des systèmes à événements discrets. Nous définissons dans la première partie de

ce chapitre les notions de base intervenant dans le formalisme classique des réseaux de Pétri. Grâce à cet outil, nous montrerons que la dynamique du système modélisé est décrite par l'évolution du marquage du réseau de Pétri. Le changement de marquage se fait par le bais de franchissement des transitions, traduisant ainsi un changement d'état du système. Après la présentation des propriétés comportementales et structurelles des réseaux de Pétri, la première partie de ce chapitre se termine par des exemples de modélisation de systèmes utilisés dans les systèmes automatisés de stockage/déstockage. Etant donné que nous avons choisi les réseaux de Petri colorés (RdPC) comme outil de modélisation de notre AS/RS à convoyeur gravitationnel, la deuxième partie de ce chapitre est consacrée à cet outil qui constitue une abréviation des réseaux de Petri ordinaires. Après une présentation intuitive, nous donnons la définition formelle d'un réseau de Petri coloré. Ensuite nous décrivons les règles de franchissement des transitions colorés régissant l'évolution du marquage. Pour ce faire, des fonctions spéciales associées aux arcs sont définies. Un exemple d'application utilisant un modèle réseau de Pétri coloré est traité avant de clore le chapitre par une conclusion.

Le troisième chapitre est consacré à la modélisation de l'AS/RS à convoyeur gravitationnel par les réseaux Pétri colorés ainsi que son exploitation dans le système de pilotage. Nous commençons par montrer l'importance de la notion de réactivité dans la conception d'un système de pilotage. Ensuite nous exposons les niveaux de décisions qui interviennent dans les différentes phases de pilotage d'un système. Cela permettra de situer le niveau dans lequel s'inscrit le problème de pilotage que nous considérons. Nous passons ensuite à la description détaillée de l'AS/RS à convoyeur gravitationnel ainsi qu'à la problématique liée au problème de déstockage vis-à-vis de la demande de clients. La structure de pilotage proposée est ensuite abordée en explicitant le rôle de chaque bloc. Cette structure s'articule autour du modèle RdPC et d'un bloc d'optimisation. Le modèle RdPC développé est détaillé dans une section avant de montrer son exploitation dans la boucle de pilotage. Nous étudions ensuite dans une section deux modes de traitement de la demande selon l'intensité de celle-ci.

Le quatrième chapitre porte sur l'optimisation du nombre de cycles de déstockage à effectuer par la machine de déstockage pour satisfaire une demande de clients. Après une présentation de la position du problème d'optimisation en question, nous abordons la programmation en nombres entiers pour la modélisation des problèmes de décision en recherche opérationnelle. En effet, le problème de déstockage sera exprimé en termes de programme en nombres entiers. Nous décrivons ensuite la méthode de résolution par

séparations et évaluations successives, qui est très utilisée pour résoudre ce type de problèmes. Ceci étant, nous formulons le problème d'optimisation du nombre de cycles de déstockage en définissant toutes les variables et relations qui le régissent. Des exemples concrets seront ensuite traités pour montrer l'efficacité de la méthode d'optimisation proposée.

Le cinquième chapitre porte sur l'intégration des dates de péremption des produits dans le pilotage de l'AS/RS à convoyeur gravitationnel. Dans un premier temps, nous présentons le contexte et les enjeux qui nous ont poussés à prendre en compte la date de péremption comme donnée supplémentaire du produit. Nous montrons qu'il devient indispensable de considérer la péremption quand il s'agit de stocker des produits périssables qui ont une date d'expiration prédéfinie. Dans un deuxième temps, nous étendons le modèle réseau de Pétri coloré pour contenir l'information relative à la péremption du produit. Le problème d'optimisation du nombre de cycles de déstockage est ensuite formulé afin de tenir en compte les péremptions des produits stockés dans le rack. Nous expliquons comment nous procédons pour satisfaire une demande en un minimum de cycles tout en minimisant, par un choix judicieux de produits, le risque de péremption dans le rack. Des exemples concrets seront ensuite traités pour illustrer l'efficacité de la méthode d'optimisation proposée.

Enfin, nous présentons dans la conclusion une synthèse des travaux que nous avons réalisés et nous proposons quelques perspectives de recherche futures.

Chapitre 1

Les Systèmes Automatisés de Stockage/Déstockage

1. 1. Introduction

Depuis les années 50, la technologie de stockage et de déstockage automatisés a été employée dans le domaine industriel. Les objets sont stockés dans des casiers ou des palettes et un mécanisme robotisé de manutention peut déstocker un casier suite à une instruction d'un opérateur. Ce système a été à l'origine conçu pour réduire le temps de marche qui avoisinait 70% du temps de déstockage manuel (Kirsch, 1999).

Les systèmes automatisés de stockage/déstockage (Automated Storage and Retrieval Systems, AS/RS) sont utilisés typiquement dans des applications où il y a un important volume de charges transitant en entrée et en sortie du stock. Ils sont largement utilisés dans les opérations de production et de distribution pour tenir et stocker le flux de matériel traversant le processus jusqu'à l'utilisateur. La plupart des systèmes fonctionnent en mode entièrement automatisé avec peu d'intervention humaine dans la manipulation du matériel. Cela conduit à une précision extrêmement élevée d'inventaire.

Un AS/RS est composé typiquement de racks de stockage, de machines de stockage et déstockage, de convoyeurs et de stations d'entrée/sortie où la machine de stockage/déstockage peut charger et décharger des produits.

Ce chapitre est consacré à la présentation des systèmes automatisés de stockage et déstockage (AS/RS), cadre de notre étude. Nous commencerons tout d'abord par présenter la configuration d'un AS/RS dans la section 1.2. Dans la section 1.3, nous exposerons les différents types d'AS/RS. Les politiques de stockages utilisés dans les AS/RS seront définies dans la section 1.4. Un état de l'art des problématiques étudiées dans la littérature des AS/RS est présenté dans la section 1.5. Finalement, nous terminons par une conclusion.

1. 2. Configuration d'un AS/RS

Un AS/RS est un système qui emploie des machines automatisées et un système de contrôle pour transporter, stocker et récupérer des produits suivant un ordre de commande. Les produits sont stockés dans des racks. Les produits à stocker ou à déposer transitent par des stations d'entrées/sorties.

1. 2. 1. Composants et terminologie utilisée

Un système automatisé de stockage/déstockage est composé :

- d'une série d'allées avec des étagères de stockage.
- de machines de stockage/déstockage, généralement une par allée, pour stocker et déstocker les produits.
- des stations d'entrées/sorties, où les produits sont déposés pour le stockage, ou délivrés pour l'utilisateur.

La figure 1.1 montre les différents composants d'un AS/RS.

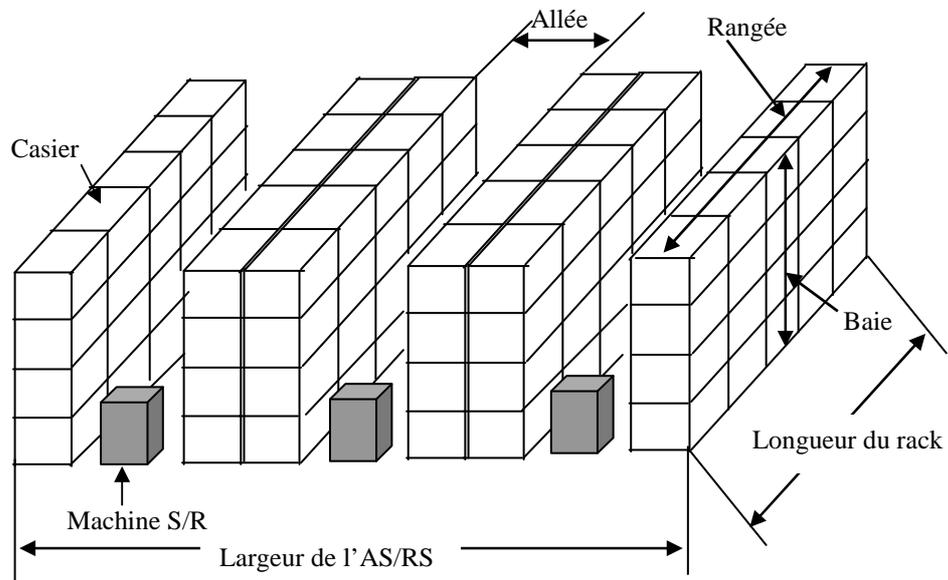


Figure 1.1. Structure générique d'un AS/RS.

- Casier : le casier est l'unité de stockage de base. C'est l'espace tridimensionnel nécessaire au stockage d'une charge unitaire ou un ensemble de charges.
- Baie : c'est la colonne formée de l'ensemble de casiers allant du sol au plafond.
- Rangée : c'est l'ensemble de casiers formés par des baies placées les unes à la suite des autres.
- Etagère : c'est la structure formant une rangée. C'est en général une structure métallique, conçue pour supporter la totalité des charges placées dans chaque casier.
- Allée : l'allée est l'espace entre deux rangées où circule la machine de stockage/déstockage.
- Machine S/R : la machine de stockage/déstockage (*Storage/Retrieval machine, S/R machine, en anglais*) déplace les produits depuis et en direction du rack. La machine S/R doit être capable de se déplacer à la fois horizontalement et verticalement. Un système de rails permet de guider la machine S/R dans son déplacement.
- Module de stockage : Ce sont des palettes ou des containers de formes standardisées pouvant être placés dans les casiers, chargés par les machines S/R et déchargés dans les stations d'entrée/sortie. Les containers peuvent être des paniers, des bacs, des boîtes de cartons...
- Station d'entrée/sortie : la station d'entrée/sortie (*Input/output station, I/O station, en anglais*) est aussi appelée station de dépôt/livraison. C'est le point par où transitent tous les produits entrant ou sortant du système. C'est aussi, l'interface entre la machine S/R et le système de transfert externe à l'AS/RS, qui est généralement réalisé par des convoyeurs à accumulation.

1. 2. 2. Fonctionnement d'un AS/RS

Pour une opération de déstockage, un opérateur sélectionne un casier à déstocker et envoyer vers la station d'entrée/sortie. Suite à cette sélection, une instruction est transmise à la machine S/R par le biais d'un système de contrôle. La machine se déplace d'abord vers l'allée concernée puis aux coordonnées (ligne et colonne) exactes du casier à déstocker. La machine récupère alors le produit de cet endroit et le transfère vers la station d'entrée/sortie.

Une fois le produit déposé à la station d'entrée/sortie, la machine S/R peut soit revenir à sa position de repos, soit entamer directement un autre cycle de stockage ou déstockage. Pour un cycle de stockage, il est similaire à un cycle de déstockage sauf que la machine doit récupérer un produit de la station d'entrée/sortie et le déplacer vers un casier déterminé. Deux modes de fonctionnement de la machine S/R peuvent être envisagés :

- Fonctionnement en simple cycle (SC) : dans ce mode, la machine S/R effectue une seule opération (stockage ou déstockage) par cycle. Ainsi, la machine S/R visite un seul casier par cycle.
- Fonctionnement en double cycle (DC) : dans ce mode, la machine S/R effectue d'abord une opération de stockage et une fois libre elle continue directement vers un autre casier pour effectuer une opération de déstockage. Ainsi, la machine S/R visite deux casiers par cycle.

1. 3. Types d'AS/RS

Il existe plusieurs classes d'AS/RS. Le plus connu est l'AS/RS à charge unitaire, il est considéré comme le modèle générique des autres types.

1. 3. 1. L'AS/RS à charge unitaire

L'AS/RS à charge unitaire est un système dans lequel l'équipement de manutention manipule les produits comme une seule unité (Fig. 1.2.). Typiquement, ceci signifie des charges sur des palettes, mais peut s'appliquer aux charges conteneurisées.

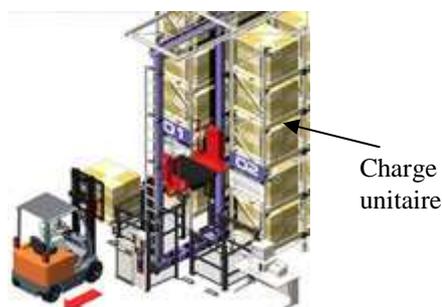


Figure 1.2. Aperçu d'un AS/RS à charge unitaire

L'AS/RS est disponible avec une variété d'options de vitesses horizontales et verticales. En conséquence, il est facile de sélectionner la bonne machine pour l'application considérée.

De même, les grues légères actuelles avec leur électronique moderne tiennent en compte des taux variables d'accélération assurant les vitesses les plus rapides tout en maintenant la stabilité de charge.

1. 3. 2. L'AS/RS à mini-charge

Un AS/RS à mini charge manipule des charges qui sont typiquement contenues dans de petits récipients ou bacs, avec un poids de charge variant typiquement dans une gamme de 100 à 500 livres (45 à 225 Kg).

Un AS/RS à mini charge permet de stocker plus de matériel en moins d'espace, particulièrement du matériel dans des bacs de manutention. Il peut également manipuler ces bacs avec beaucoup plus de vitesse que des méthodes statiques et manuelles conventionnelles. La figure 1.3 donne la configuration typique d'un AS/RS à charge mini charge.

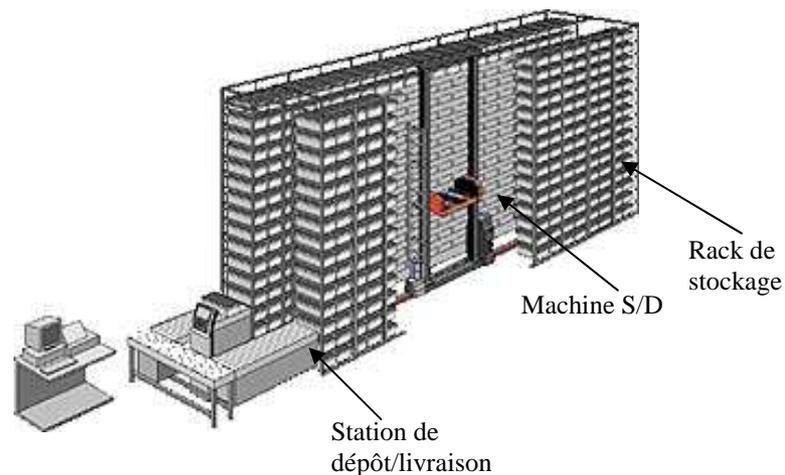


Figure 1.3. Configuration typique d'un AS/RS à min charge

L'AS/RS à mini charge offre une variété de méthodes de manipulation de charge selon l'application. Il existe des modèles avec simple ou double navette ou encore à bras manipulateur. Les vitesses peuvent varier d'un fabricant à un autre, d'un modèle à un autre. Il va de même pour les dimensions.

1.3.3. L'AS/RS à carrousel

Les carrousels sont une bonne alternative aux systèmes à mini charge. Dans ces systèmes, se sont les casiers qui se déplacent vers l'extrémité du rack ou un opérateur (personne ou robot) stocke/déstocke les produits. Le système consiste en un ensemble de casiers montés en carrousel et se déplaçant horizontalement vers une extrémité du rack où l'opération de stockage/déstockage sera réalisée (Fig. 1.4).



Figure 1.4. Aperçu d'un AS/RS à carrousel.

1. 3. 4. L'AS/RS à convoyeur gravitationnel

L'AS/RS à convoyeur gravitationnel est composé d'un rack profond composé d'un ensemble de casiers pouvant contenir plusieurs produits placés les uns à la suite des autres. Chaque casier est muni d'un convoyeur gravitationnel à base de rouleaux ou roues libres inclinées de telle façon à permettre le glissement des produits d'un bout à l'autre du casier, donc d'une face à l'autre du rack. Les produits sont stockés d'un coté du rack (face de stockage) par la machine de stockage (MS), ils glissent sur les convoyeurs gravitationnels, et sont déstockés de l'autre coté du rack (face de déstockage) par la machine de déstockage (MD). Les machines de stockage et de déstockage peuvent se déplacer simultanément sur deux axes. Ces deux axes forment un plan parallèle aux deux faces du rack. Lorsqu'une charge est déstockée, les charges qui sont derrière avancent vers l'avant. La figure 1.5 donne la structure d'un tel AS/RS. Une description plus fine sera présentée au chapitre 3.

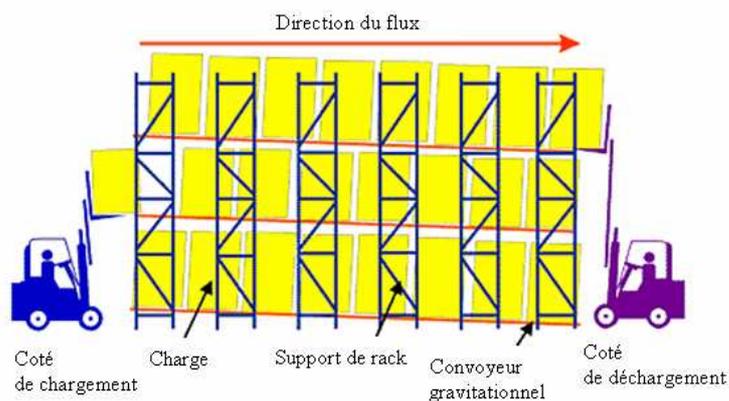


Figure 1.5. Structure d'un AS/RS à convoyeur gravitationnel.

Cet AS/RS présente plusieurs avantages par rapport aux autres AS/RS, nous pouvons citer :

- Gestion FIFO (First In First Out) : les produits se déplacent via le convoyeur gravitationnel du côté de chargement vers le côté de déchargement du système. Cette rotation constante du stock est particulièrement bénéfique pour les produits périssables, tel que les aliments et les produits pharmaceutiques.
- Réduction des coûts de maintenance : Du fait de la rotation automatique, exploitant juste la gravitation, il n'y a pas besoin d'énergie électrique ou d'autres utilités.
- Réduction de l'espace au sol du fait de la suppression des espaces réservés aux allées. Ainsi la capacité de stockage se trouve améliorée pour la même zone de stockage.
- Réduction du temps et du travail : avec uniquement deux allées, un pour le chargement et l'autre pour le déchargement, le trafic des deux machines est plus ordonné et efficace améliorant ainsi la productivité.

1. 4. Avantages des AS/RS

L'apport des AS/RS par rapport aux systèmes conventionnels réside dans plusieurs facteurs, nous pouvons citer (Singh, 1996, Sari, 2003) :

- Diminution de l'espace requis : un AS/RS peut diminuer considérablement l'espace nécessaire au stockage. la place ainsi récupérée peut être utilisée pour augmenter la production.
- Augmentation de la capacité : pour le même espace, un AS/RS augmente considérablement la capacité de stockage. Une entreprise, dont le système de stockage est saturé peut, en choisissant un AS/RS, accroître sa capacité de stockage, sans investir dans de nouveaux locaux.
- Gestion et contrôle des stocks améliorés : à n'importe quel instant, le système de contrôle de l'AS/RS peut donner avec précision les quantités types et qualités, des produits, présents en stock, celles délivrées dans l'heure, la journée, la semaine...
- Ordonnancement de production amélioré : grâce à une meilleure connaissance et à un contrôle plus précis des en-cours, l'ordonnancement de la production peut être mieux optimisé.
- Réduction des temps de réponse : les systèmes AS/RS possèdent des temps de réponse très réduits, pour localiser, stocker ou déstocker des articles, par rapport aux systèmes classiques de stockage.
- Réduction des stocks : due à une meilleure gestion et contrôle des stocks, et à des temps de réponse réduits il est possible de réduire, la quantité de produits en stock.
- Réduction des risques de rupture de stocks : grâce à une meilleure gestion de stocks, qui peut se faire en temps réel. Les ruptures de stocks sont pratiquement inexistantes, ou de très faibles durées lorsqu'elles sont inévitables.
- Réduction de la main d'œuvre : l'automatisation de la majorité des opérations dans les AS/RS permet de diminuer l'intervention humaine, que ce soit dans l'administration ou l'exécution.
- Réduction des détériorations : l'automatisation permet la diminution des risques de casse des produits.
- Diminution des coûts de fonctionnement : étant donné le haut degré d'automatisation, les coûts de fonctionnement sont réduits au maximum (réduction du personnel, de la lumière, du chauffage...)

- Augmentation du rendement dans la production : grâce à l'augmentation de la productivité des stations de travail, les AS/RS permettent d'augmenter le rendement du système de production.
- Augmentation de la sécurité : les AS/RS étant des aires autonomes et fermées, sans intervention humaines, la sécurité est grandement améliorée.
- Réduction des vols : les AS/RS étant des aires autonomes et fermées, sans intervention humaines, le vol de produit par le personnel, ou autres, est très réduit.
- Grande flexibilité de fonctionnement : les AS/RS peuvent s'accommoder d'une grande variété d'articles à stocker. De plus, ils sont facilement interfaçables avec des systèmes comme les AGVS, les FMS, les systèmes d'inspection...
- Production juste à temps (JIT) : les AS/RS permettent d'avoir le bon produit, outils, palette et support au bon endroit au bon moment.

1. 5. Politiques de stockage

Il existe trois méthodologies de stockage : Le stockage dédié où chaque type de produit possède sa propre aire de stockage ; le stockage aléatoire (ouvert) où chaque produit est assigné à un casier de manière aléatoire ; et le stockage aléatoire par classe où les produits sont divisés généralement en plusieurs classes suivant leur temps et leur fréquence de stockage.

1.5.1. Stockage dédié

Dans le stockage dédié, l'espace de stockage est partagé en blocs, chacun est utilisé pour stocker un produit déterminé. Cette méthode de stockage présente des avantages et des inconvénients. Les principaux avantages sont:

- Chaque produit ayant une aire de stockage prédéterminée, le contrôle du système est grandement simplifié.

- La gestion des stocks est simplifiée : la quantité de chaque produit peut être contrôlée visuellement.
- Chaque aire de stockage peut être conçue pour le stockage du produit correspondant (casier de taille variables, allées plus ou moins larges,...).

Les principaux inconvénients sont :

- L'espace de stockage est augmenté parce qu'il faut donner suffisamment d'espace pour les stocks maximum de chaque produit. La somme des stocks maximum de chaque produit étant supérieure au stock maximum de tous les produits. Puisque les stocks maxima pour chaque produit sont de l'ordre du double des stocks moyens, le taux de remplissage du système de stockage sera relativement bas.
- Le système est difficilement modifiable : l'ajout de nouveaux produits non prévus à l'origine est très complexe.

1.5.2. Stockage aléatoire ouvert

Dans le stockage aléatoire, n'importe quel produit est stocké dans n'importe quel casier. Un système de contrôle élaboré garde en mémoire les adresses de chaque type de produit. Le principal avantage est que ce type de stockage utilise moins d'espace que le stockage précédent. L'espace total de stockage est équivalent au stock maximum de tous les produits (Askin, 1993). Ceci permet une réduction importante d'espace. Cependant, ce système nécessite un contrôle très élaboré.

1.5.3. Stockage aléatoire par classe

Le stockage aléatoire permet de réduire l'espace de stockage par rapport au stockage dédié. Le stockage par classe s'attaque au temps de cycle. Des réductions du temps de cycle peuvent atteindre 40% (Askin, 1993). Dans ce type de stockage les produits stockés dans un AS/RS suivent leur durée de stockage. Si les produits qui restent stockés peu de temps sont placés près de la station de livraison et les produits qui restent plus longtemps sont placés loin de la station, le rendement du système peut être grandement amélioré. Le stockage par classe

subdivise les casiers en classes par rapport à leurs distances du point de livraison. Les casiers les plus proches formant la première classe, les plus loin formant la dernière (Fig. 1.6).

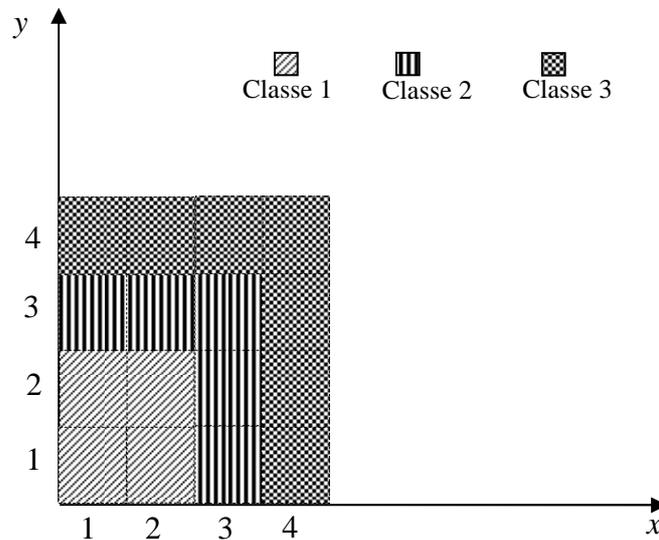


Figure 1.6. Stockage selon trois classes.

1. 6. Modèles des temps de transport dans les AS/RS

L'expression de Tchebychev permet de calculer les temps de transport de la machine S/R nécessaires pour atteindre n'importe quel point sur la face du rack. Dans ce qui suit, nous allons représenter la face du rack sur un plan xy et nous supposons que la machine S/R se déplace simultanément dans le sens horizontal et vertical. Le point d'entrée/sortie depuis lequel la machine S/R démarre est localisé à la position (0, 0).

1. 6. 1. Distribution des temps de transport

Le temps de transport est donné par l'expression suivante (Aschayeri, 2002) :

$$F(x, y) = \max (x/Sh, y/Sv) ; \text{ le temps de transport de Tchebychev (s).} \quad [1]$$

Avec :

x : l'abscisse (horizontale) de la position de l'emplacement dans le rack (m).

y : l'ordonnée (verticale) de la position de l'emplacement dans le rack (m).

Sh : est la vitesse horizontale de la machine S/R (m/s).

Sv : est la vitesse verticale de la machine S/R (m/s).

L'expression [1] permet de représenter la distribution des temps de transport de la machine (S/R) en fonction des coordonnées (x , y) et des valeurs des vitesses Sh et Sv .

Afin de montrer l'influence des vitesses Sh et Sv sur les temps de transport, nous présentons sur les figures 1.7, les trois cas dans lesquels : $Sh > Sv$, $Sh = Sv$ et $Sh < Sv$.

✓ **Cas1** : $Sh > Sv$ avec :

$Sh = 10$ m/s.

$Sv = 1.7$ m/s.

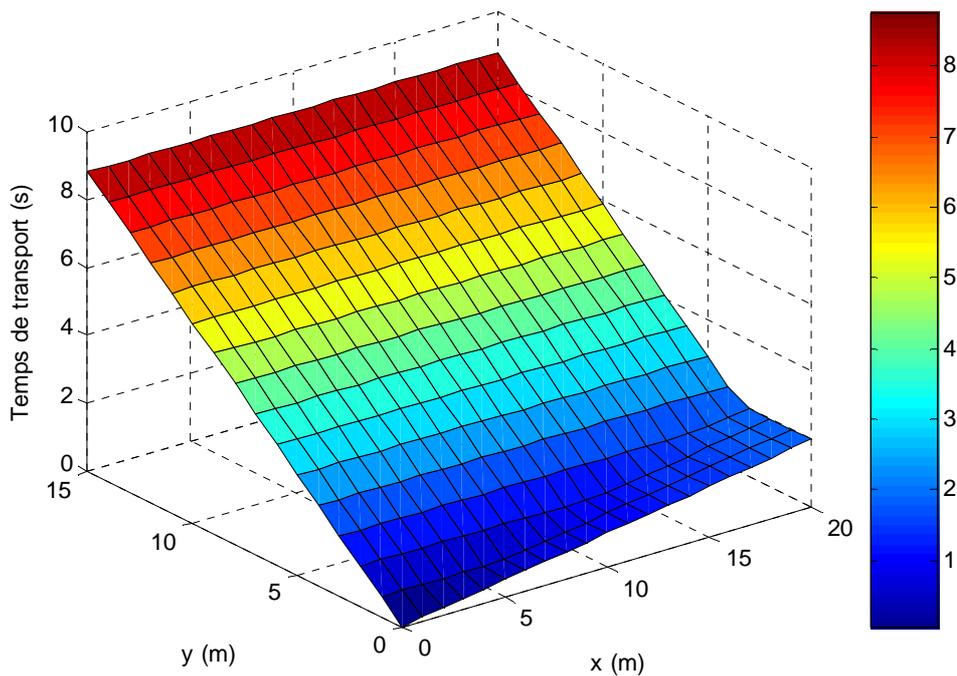


Figure 1.7.a. Distribution des temps de transport pour $Sh > Sv$.

✓ **Cas2 :** $Sh = Sv$ avec :

$Sh = 10$ m/s.

$Sv = 10$ m/s.

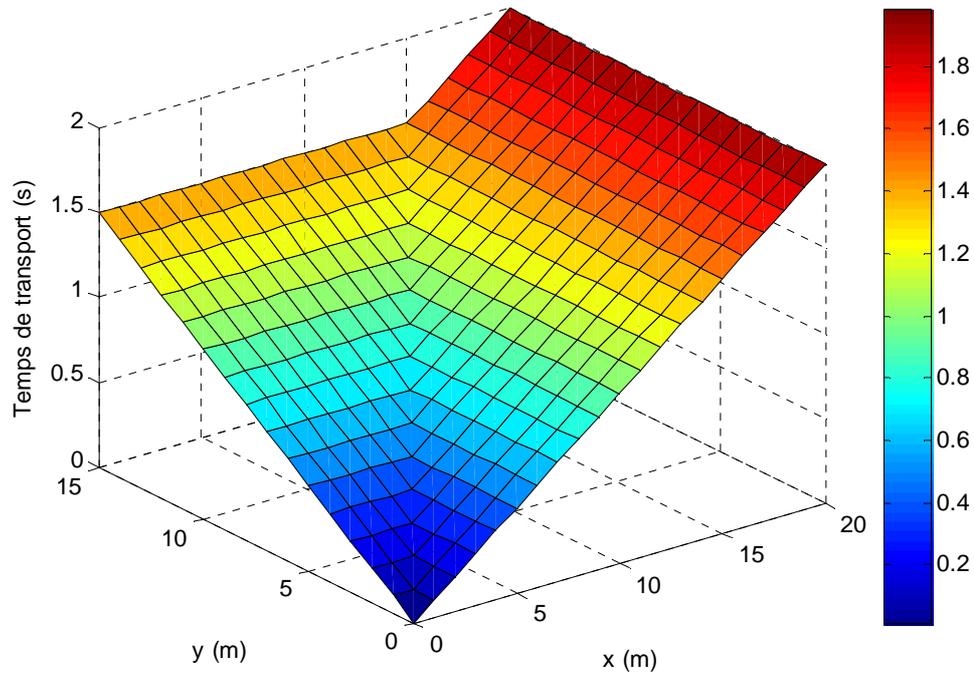


Figure 1.7.b. Distribution des temps de transport pour $Sh = Sv$.

✓ **Cas 3 :** $Sh < Sv$ avec :

$Sh = 1$ m/s.

$Sv = 14$ m/s.

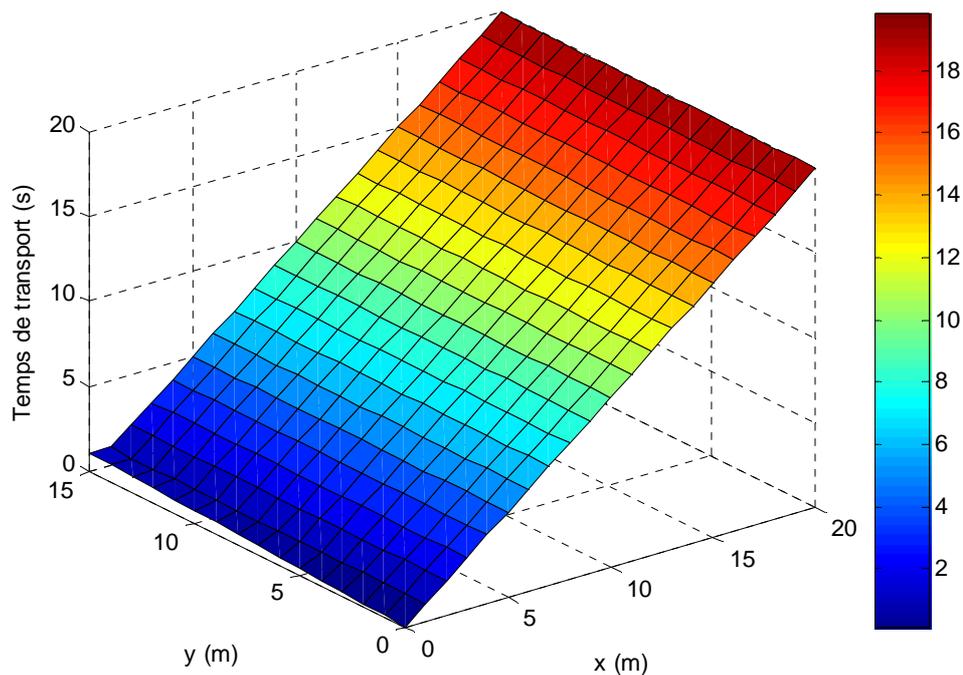


Figure 1.7.c. Distribution des temps de transport pour $Sh < Sv$.

1.6.2. Temps moyens de transport

Bozer et White (Bozer, 1984) ont développé des expressions analytiques du temps moyen de simple et double cycle dans les AS/RS à charge unitaire. Ils ont utilisé une approche statistique basée principalement sur l'approximation des distances des casiers du rack en fonction continue. Ils ont considéré pour leur étude les hypothèses suivantes :

- Une approximation continue de la face discrète du rack.
- Le système consiste en une seule machine S/R qui peut fonctionner soit en simple soit en double cycle.
- La machine S/R est capable de se déplacer à la fois horizontalement et verticalement à des vitesses constantes. Le temps nécessaire pour atteindre n'importe quel emplacement du rack est donné par l'expression de Tchebychev.
- La position du point d'entrée/sortie est située au coin bas et gauche du rack.
- Les temps de chargement et déchargement de la machine S/R sont ignorés.

- Les produits sont aléatoirement stockés ou déstockés. Tous les points de la face du rack sont équiprobables pour subir une opération de stockage/déstockage.

Ils ont défini d'abord les facteurs suivant :

- t_h : le temps de déplacement horizontal de la station d'entrée/sortie à la baie la plus éloignée.
- t_v : le temps de déplacement vertical de la station d'entrée/sortie à la ligne la plus éloignée.
- T : le facteur de normalisation avec $T = \max(t_h, t_v)$.
- b : le facteur de forme avec $b = \min(t_h/T, t_v/T)$, ce qui implique que :

$$0 < b \leq 1.$$

Avec la supposition $t_h > t_v$, ce qui implique que $T = t_h$ et $b = t_v/T$. En raison du stockage aléatoire, les points de stockage/déstockage sont aléatoirement distribués entre 0 et 1 horizontalement selon l'axe des abscisses et entre 0 et b verticalement selon l'axe des ordonnées. Le temps de transport normalisé entre deux points sur la face du rack représenté par (x_1, y_1) et (x_2, y_2) est alors égal à $\max(|x_1 - x_2|, |y_1 - y_2|)$. Ils ont considéré $G(z)$ comme étant la probabilité que le temps de transport de $(0, 0)$ à (x, y) est inférieur ou égal à z . En supposant que les coordonnées (x, y) sont indépendamment générées on a :

$$G(z) = Pr(x \leq z) \cdot Pr(y \leq z) \quad [2]$$

Avec des points de stockage/déstockage uniformément distribués, on a :

$$Pr(x \leq z) = z \quad [3]$$

$$Pr(x \leq y) = \begin{cases} z/b & \text{si } 0 < z \leq b \\ 1 & \text{si } b < z \leq 1 \end{cases} \quad \text{d'où : } G(z) = \begin{cases} z^2/b & \text{si } 0 < z \leq b \\ z & \text{si } b < z \leq 1 \end{cases} \quad [4]$$

$$\text{La densité de probabilité décrivant } G(z) \text{ est : } g(z) = \begin{cases} 2z/b & \text{si } 0 < z \leq b \\ 1 & \text{si } b < z \leq 1 \end{cases} \quad [5]$$

$$\text{Le temps moyen de simple cycle est donc : } \overline{E(SC)} = 2 \int_{z=0}^1 zg(z)dz \quad [6]$$

Après résolution de l'équation [6] et multiplication par le facteur de normalisation T , le temps moyen de simple cycle devient donc : $\overline{E(SC)} = T \left(\frac{b^2}{3} + 1 \right)$ [7]

En suivant la même démarche, le temps moyen de double cycle sera égal à :

$$\overline{E(DC)} = T \left(\frac{4}{3} + \frac{b^2}{2} - \frac{b^3}{30} \right) \quad [8]$$

1.7. Etat de l'art

Il existe plusieurs types d'AS/RS tel que l'AS/RS à charge unitaire, à mini-charge, et l'AS/RS à convoyeur gravitationnel. Le type d'AS/RS le plus étudié est l'AS/RS à charge unitaire qui constitue le modèle générique, les autres étant des variantes de celui-ci. Ce système est composé d'un rack à charge unitaire dont les produits à charge unitaire sont stockés dans des emplacements accessibles sur la face avant du rack. En revanche, l'AS/RS à convoyeur gravitationnel (Flow-rack AS/RS) possède un rack profond, où les produits sont stockés en profondeur dans des casiers à plusieurs emplacements. Une machine de stockage introduit les produits de la face arrière du rack, tandis qu'une machine de déstockage les récupère de la face avant, de ce fait les opérations de stockage et déstockage sont indépendantes. Cette différence fondamentale entre les deux AS/RS dû au stockage en profondeur induit une grande difficulté d'étude et de commande.

Comme performances principales poursuivies dans les AS/RS, on peut citer la réduction du temps moyen de transport de la machine de stockage/déstockage (S/D) et l'amélioration du débit du système (*Throughput*). Le débit du système est considéré comme la plus importante performance des AS/RS par (Randhawa *et al.*, 1991). Pour atteindre cet objectif d'amélioration du débit de sortie, les chercheurs ont développé plusieurs approches. Dans (Bozer *et al.*, 1984), les auteurs développent des expressions pour les temps moyens de simple et double cycle pour un rack à charge unitaire rectangulaire. Ils ont utilisé une approximation continue du rack avec une politique de stockage aléatoire en explorant plusieurs configurations du point d'entrée/sortie et du point de repos de la machine S/D. Dans (Ashayeri, 2002) les auteurs proposent une approche géométrique pour calculer le temps de cycle de la machine de stockage/déstockage (S/R) fonctionnant en simple et double cycle. Le rack traité peut être carré en temps ou non carré en temps. D'autres auteurs comme (Egbelu,

1991), (Peters *et al.*, 1996), (Chang *et al.*, 1997) ont traité l'impact de la position de repos de la machine (S/D) sur le temps de réponse du système. Cependant, dans (Meller *et al.*, 2005), les auteurs ont fait une investigation par simulation sur l'impact du point de repos sur le temps moyen de transport et ils ont trouvé que le choix du point de repos de la machine S/D sous un taux élevé d'utilisation n'affecte pas significativement le temps de réponse du système. En s'inspirant du travail de (Bozer *et al.*, 1984), (Sari *et al.*, 2005) ont développé deux modèles, approché et exact, du temps de transport pour l'AS/RS à convoyeur gravitationnel. Le modèle approché est basé sur une approche continue, où le rack est supposé continu. Le modèle exact est basé sur une approche discrète. Les expressions des deux approches sont comparées via la simulation. Les auteurs concluent qu'il n'y a pas de différence significative entre les résultats obtenus par les deux approches mais l'approche continue est plus pratique, à cause de sa simplicité de calcul par rapport à l'approche discrète qui est extrêmement complexe.

Un autre axe de recherche dans les AS/RS concerne l'ordonnancement des opérations de stockage/déstockage dans le but d'améliorer le temps de réponse du système. Linn et Wysk (1990a, 1990b) ont développé un système expert qui est capable d'adapter les règles d'ordonnancement pour atteindre des buts spécifiques. Ils concluent sur le besoin de développer des techniques appropriées d'ordonnancement de déstockage en tant qu'élément de la commande stratégique et tactique globale pour améliorer les performances du système.

Une pratique commune dans l'ordonnancement des demandes de stockage et déstockage consiste à considérer un traitement selon la règle du premier arrivé premier servi (*First-Come-First served, FCFS*). Le choix de la règle FCFS est raisonnable pour le stockage, du fait que la disposition des produits destinés au stockage présente un flux physique dont il est difficile de modifier l'ordre (dans un convoyeur par exemple). Cependant, le choix de la règle FCFS pour le déstockage est moins justifiée du fait que les demandes de déstockage sont juste des messages électroniques (flux d'information) et peuvent être facilement re-séquencées. Han *et al.* (1987) ont étudié des méthodes d'ordonnancement pour les opérations de déstockage fonctionnant en commande double cycle. Ils ont trouvé que quand il y a un seul emplacement disponible pour le stockage, le problème est formulé comme le problème du voyageur de commerce. Lorsqu'il y a plusieurs emplacements disponibles pour le stockage, ils considèrent que le problème devient plus difficile et proposent deux heuristiques.

Dans (Lee *et al.*, 1996), les auteurs ont étudié un problème d'ordonnancement des demandes de déstockage dans un AS/RS à charge unitaire. Dans le but de réduire le temps

total de transport de la machine S/R, ils ont introduit une méthode qui trouve une solution optimale ou quasi-optimale pour des problèmes de taille moyenne.

Dans une étude de (Kim *et al.*, 2003), les auteurs proposent un nouveau processus de remplissage de racks interchangeables en fonction des demandes de clients. L'objectif est de minimiser les temps de préparation et de lancement des produits demandés. Dans (Van den Berg *et al.*, 1999), pour réduire le temps de réponse du système, les auteurs se sont intéressés à l'ordonnancement des opérations de stockage/déstockage en utilisant l'approche d'ordonnancement par bloc avec un stockage dédié. L'objectif était de trouver un chemin avec un temps total de transport minimum. Le problème est équivalent au problème du voyageur de commerce et ils ont montré que dans le cas particulier avec une politique de stockage dédié, le problème peut être résolu en un temps polynomial.

L'utilisation des réseaux de Petri pour la modélisation et la commande des AS/RS a suscité quelques intérêts. Les auteurs dans (Amato *et al.*, 2005) développent deux algorithmes de commande pour un AS/RS modélisé par des réseaux de Petri colorés temporisés. Ils ont introduit un système d'optimisation en temps réel, dans l'architecture de contrôle, afin d'améliorer le débit du système. (Dotoli *et al.*, 2005) ont proposé une modélisation modulaire par les réseaux de Petri colorés temporisés d'un AS/RS à charge unitaire multi-allées, servi par des véhicules guidés par rails. Le modèle a servi pour tester plusieurs politiques de contrôle et de gestion et montrer comment le modèle simulé peut aider à améliorer les performances du système.

Dans ce travail, nous traitons le problème de pilotage dynamique d'un AS/RS à convoyeur gravitationnel et nous nous intéressons à la réduction du temps de réponse du système et par conséquent à l'amélioration du débit de sortie, qui correspond au nombre de produits délivrés par unité de temps. Sachant que la majorité des travaux ont porté sur l'AS/RS à charge unitaire et vu la particularité de l'AS/RS à convoyeur gravitationnel où les produits désirés peuvent se trouver en profondeur à l'intérieur du rack, nous nous attacherons également à la réduction des opérations de restockage très coûteuse en temps.

1. 8. Conclusion

Dans ce chapitre, nous avons présenté les systèmes automatisés de stockage/déstockage avec leurs différentes caractéristiques. Ces systèmes apportent un certain nombre d'avantages

tel que l'élimination de l'attente, de la marche et le temps de recherche des produits. Ils offrent une meilleure exactitude dans l'inventaire par sa vérification en temps réel et avec des rapports instantanés, d'où une meilleure réactivité.

Un état de l'art a été dressé pour montrer les différentes problématiques traitées et les axes de recherches dans ce domaine. Il en ressort qu'une des plus importantes performances des AS/RS, que les chercheurs proposent d'améliorer est le débit de sortie du système, c'est-à-dire le nombre de produits déstockés par unité de temps. Pour ce faire, ils suivent plusieurs approches.

Dans ce travail, nous allons traiter le problème du pilotage dynamique du système automatisé de stockage/déstockage à convoyeur gravitationnel. Dans le prochain chapitre nous aborderons les réseaux de Pétri et leurs notions de bases. Cet outil avec son abréviation coloré va servir pour la modélisation du système de stockage/déstockage à convoyeur gravitationnel.

Chapitre 2

Notions de base sur les réseaux de Petri

2.1. Introduction

Les réseaux de Petri ont été introduits pour la première fois en 1962 par le mathématicien Carl Adam Petri pour décrire les relations existant entre des conditions et des événements, de modéliser le comportement de systèmes dynamiques à événements discrets. Les réseaux de Petri sont un outil graphique et mathématique qui s'applique à un grand nombre de domaines où les notions d'événements et d'états discrets sont importantes.

Le terme "réseaux de Petri" recouvre plusieurs classes de réseaux, des réseaux ordinaires jusqu'au réseau coloré et à prédicats. Les premiers sont simples à comprendre et relativement riches en propriétés mathématiques, ce qui favorise l'analyse de leur comportement. Ils ont cependant l'inconvénient de conduire à des modèles de grande taille. Les seconds permettent la mise en place de modèles compacts, entraînant une lisibilité moindre. Les réseaux de Petri présentent deux caractéristiques intéressantes. Premièrement ils permettent de modéliser et de visualiser des comportements comportant du parallélisme, de la synchronisation, et du partage de ressource. Deuxièmement, les résultats théoriques qui les concernent sont abondants ; les propriétés de ces réseaux ont été et sont encore largement étudiées.

Dans le cadre de ce chapitre, nous commençons par présenter l'ensemble des notions de bases régissant le fonctionnement des réseaux de Petri. Nous passons ensuite à la définition des réseaux de Petri colorés qui sont une abréviation des premiers réseaux et qui seront adoptés pour la modélisation de notre AS/RS. Nous terminons ensuite par une conclusion.

2.2. Les notions de base des réseaux de Petri

Un réseau de Petri (RdP) est un graphe biparti fait de deux types de sommets : les places et les transitions. Des arcs orientés relient certaines places à certaines transitions, ou certaines transitions à certaines places. Un arc ne relie jamais deux sommets de même nature.

Généralement, les places sont représentées par des cercles et les transitions par des rectangles (ou des barres). Chaque place peut contenir un ou plusieurs jetons, représentés par des points. Comme nous le verrons plus loin, ces jetons vont permettre de modéliser la dynamique du système. Le marquage d'un RdP est un vecteur à composantes entières positives ou nulles et dont la dimension est égale au nombre de places. La $n^{\text{ème}}$ composante de ce vecteur représente le nombre de jetons qui figurent dans la place numéro n du RdP.

Pour illustrer les définitions qui précèdent, considérons le RdP représenté par la figure 2-1. Ce RdP comporte quatre places notées P_1 , P_2 , P_3 et P_4 , et cinq transitions notées T_1 , T_2 , T_3 , T_4 et T_5 . Son marquage est représenté par le vecteur : $M_0 = [1.2.4.3]$.

A chaque arc est associé un poids entier positif. Lorsque ce poids n'est pas porté sur l'arc, on admet qu'il est égal à 1: c'est le cas de tous les arcs du RdP représenté par la figure 2.1.

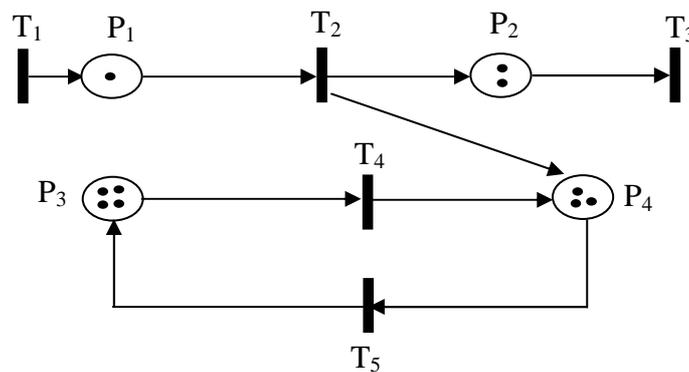


Figure 2.1. Un réseau de Petri.

De manière plus formelle (Proth et Xie, 1995), un RdP est un 5-uple $PN = (P, T, A, W, M_0)$ où :

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

$T = \{t_1, t_2, \dots, t_m\}$ est un ensemble fini de transitions,

$A \subseteq (P \times T) \cup (T \times P)$ est l'ensemble fini des arcs,

$W : A \rightarrow \{1, 2, \dots\}$ est la fonction poids attachée aux arcs,

$M_0 : P \rightarrow \{0, 1, 2, \dots\}$ est le marquage initial.

Notons que $P \cap T = \emptyset$.

Le RdP sans marquage initial est noté $N = (P, T, A, W)$. Donc $PN = (N, M_0)$.

Lorsque tous les poids des arcs sont égaux à 1, le RdP est dit ordinaire. Si M_0 est le marquage initial d'un RdP PN , $M_0(p)$ est le nombre de jetons contenus dans la place P de PN .

0T L'ensemble des places d'entrée de la transition T , c'est-à-dire l'ensemble des places P telles que $(P,T) \in A$.

T^0 L'ensemble des places de sortie de la transition T , c'est-à-dire l'ensemble des places P telles que $(T,P) \in A$.

0P L'ensemble des transitions d'entrée de la place P , c'est-à-dire l'ensemble des transitions T telles que $(T,P) \in A$.

P^0 L'ensemble des transitions de sortie de la place P , c'est-à-dire l'ensemble des transitions T telles que $(P,T) \in A$.

2. 2. 1. Marquage

La figure 2.2.b représente un réseau de Petri marqué. Chaque place contient un nombre entier positif ou nul de marques ou jetons. Le nombre de marques contenu dans une place P_i sera noté soit $M(P_i)$ soit m_i . Pour l'exemple considéré, on a $m_1 = m_3 = 1$, $m_6 = 2$ et $m_2 = m_4 = m_5 = m_7 = 0$. Le marquage du réseau, M , est défini par le vecteur de ces marquages, c'est-à-dire $M = (m_1, m_2, m_3, m_4, m_5, m_6, m_7)$. Le marquage du RdP de la figure 2.2.b est donc $M = (1, 0, 1, 0, 0, 2, 0)$. Le marquage à un certain instant définit l'état du RdP, ou plus précisément l'état du système décrit par le RdP. L'évolution de l'état correspond donc à une évolution du marquage, évolution qui se produit par le franchissement de transitions.

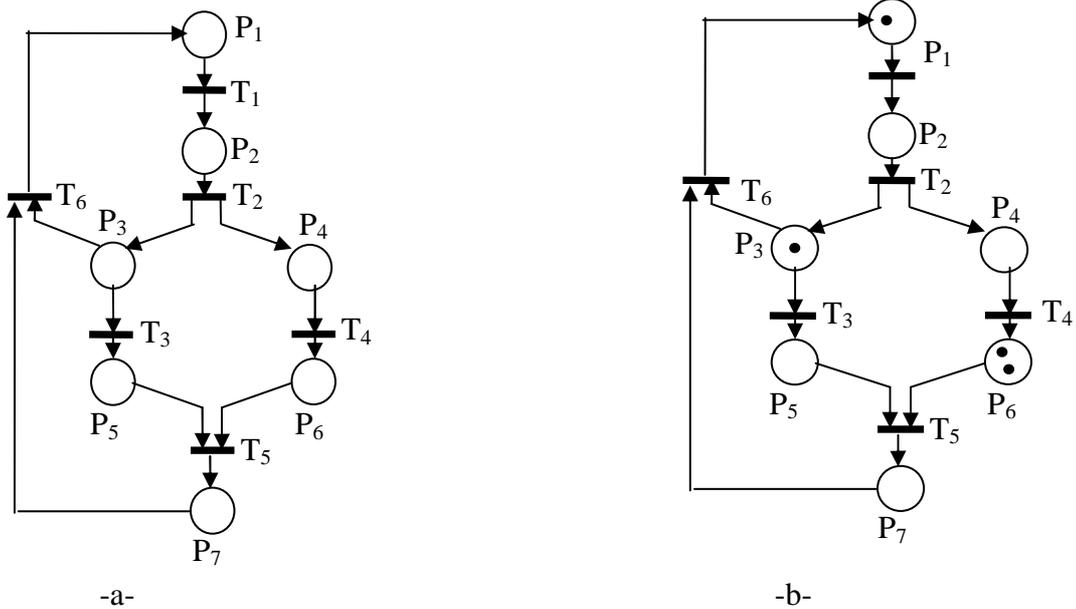


Figure 2.2. Réseau de Petri -a- non marqué –b- marqué.

Nous considérons pratiquement toujours des réseaux de Petri marqués. Nous les appellerons simplement réseaux de Petri. Selon le besoin, nous considérons le réseau de Petri marqué ou non marqué.

2. 2. 2. Franchissement des transitions

Le franchissement d'une transition ne peut s'effectuer que si chacune des places en amont de cette transition contient au moins une marque. On dit alors que la transition est franchissable, ou validée. Une transition source est donc toujours validée. Les transitions des figures 2.3.a, b et c (avant franchissement) sont validées parce que dans chaque cas les places P_1 et P_2 contiennent au moins une marque. Ce n'est pas le cas pour l'exemple de la figure 2.3.d sur laquelle la transition T_1 n'est pas validée, parce que P_1 ne contient aucune marque.

Le franchissement (ou tir) d'une transition T_j consiste à retirer une marque dans chacune des places en amont de la transition T_j et à ajouter une marque dans chacune des places en aval de la transition T_j . Ceci est illustré sur la figure 2.3.b, on remarque qu'il y a 2 jetons dans la place P_3 après franchissement parce qu'il y en avait déjà un avant. Sur la figure c on observe qu'il reste une marque dans la place P_1 , après franchissement.

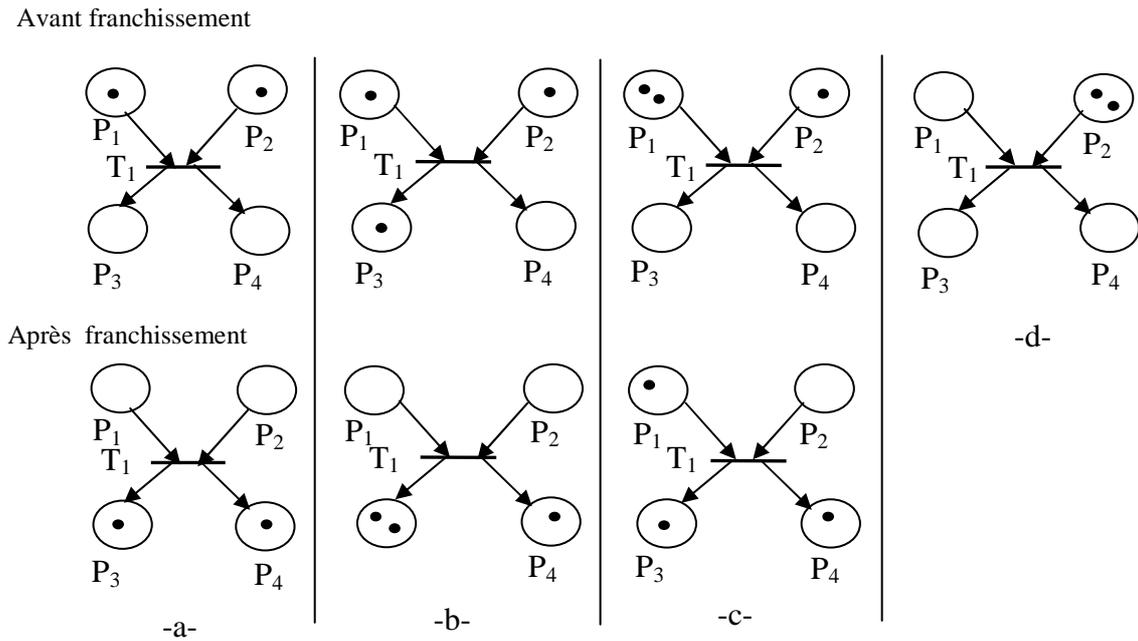


Figure 2.3. Exemple de franchissement d'une transition.

Lorsqu'une transition est validée, cela n'implique pas qu'elle sera immédiatement franchie. Ce n'est qu'une possibilité. Sur la figure 2.3.b, il y a 2 transitions qui sont validées, T_1 et T_3 . On ne sait pas quand ces transitions seront franchies, mais on sait que la prochaine évolution du marquage correspondra soit à un franchissement de T_1 soit à un franchissement de T_3 (donc qu'aucune autre évolution n'est possible).

Le franchissement d'une transition est indivisible, Bien que la notion de durée n'intervienne pas dans un RdP (s'il n'est ni temporisé ni synchronisé), il est pratique de considérer que le franchissement d'une transition a une durée nulle, pour faciliter la compréhension de la notion d'indivisibilité, (David et Alla, 1997).

2. 3. Quelques définitions particulières

2. 3. 1. Réseau de Petri autonomes et non autonomes

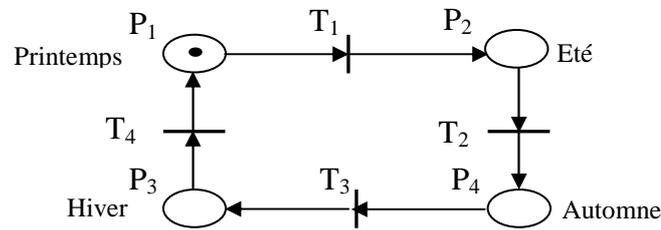


Figure 2.4. Réseau de Petri autonome.

La figure 2.4 représente le cycle des saisons. A chaque saison est associée une place, et à chaque transition le passage d'une saison à la suivante. Le marquage de la figure 2.4 correspond au printemps. On voit sur ce RdP qu'il n'y a qu'une transition validée, T_1 , et que le prochain franchissement sera donc celui de cette transition. Le marquage suivant correspondra ainsi à l'été mais on n'a aucune indication sur le moment où se produira le franchissement. Ce RdP décrit de façon qualitative le cycle des saisons. Quand un RdP décrit le fonctionnement d'un système évoluant de façon autonome, c'est-à-dire dont les instants de franchissement ne sont pas connus ou pas indiqués, on peut dire que c'est un réseau de Petri autonome. Bien que le mot autonome ne soit pas nécessaire, il permet de distinguer clairement un tel réseau d'un réseau non autonome.

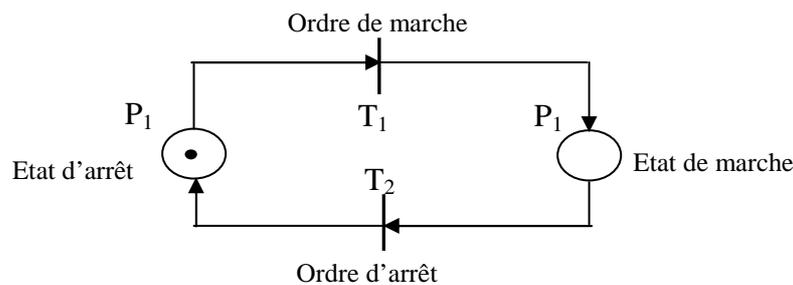


Figure 2.5. Réseau de Petri non autonome.

La figure 2.5 représente le cycle des états d'un système, un moteur par exemple, qui est arrêté, puis en marche, puis arrêté, etc. Dans l'état correspondant au marquage de la figure 2.5, le système est arrêté, et la seule transition validée est T_1 . Mais ici le franchissement de cette transition se produira quand l'événement extérieur qu'est l'ordre de démarrage sera produit. C'est un réseau de Petri non autonome.

Un réseau de Petri non autonome décrit le fonctionnement d'un système dont l'évolution est conditionné par des événements externes ou par le temps. Un RdP non autonome est synchronisé et/ou temporisé, (David et Alla, 1997).

2. 3. 2. Transition source et transition puits

Une transition sans place d'entrée est appelée transition source. Une telle transition est toujours franchissable. Dans la figure 2.6, la transition T_1 est une transition source. Une transition sans place de sortie est appelée transition puits. Une telle transition peut être tirée si elle est franchissable. Si une transition est tirée, les jetons sont prélevés dans les places d'entrées suivant les règles habituelles, mais aucun jeton n'est produit en sortie. Dans la figure 2.6, la transition T_3 est une transition puits, (Proth et Xie, 1995)

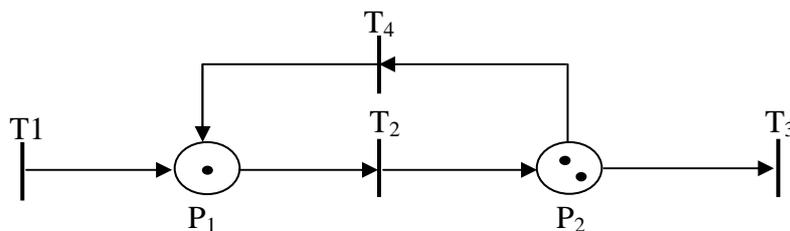


Figure 2.6. Réseau de Petri avec une transition source et transitions puits.

2. 3. 3. Circuits élémentaires et boucles

Nous utilisons souvent la notion de circuit élémentaire. Un circuit élémentaire est un chemin orienté qui part d'un sommet (place ou transition) du RdP et y revient sans jamais rencontrer plus d'une fois le même sommet. La figure 2.7 comporte, par exemple, deux circuits élémentaires : $S_1 = \{T_1, P_1, T_2, P_3, T_4, P_5, T_1\}$ et $S_2 = \{T_1, P_1, T_2, P_2, T_3, P_4, T_4, P_5, T_1\}$

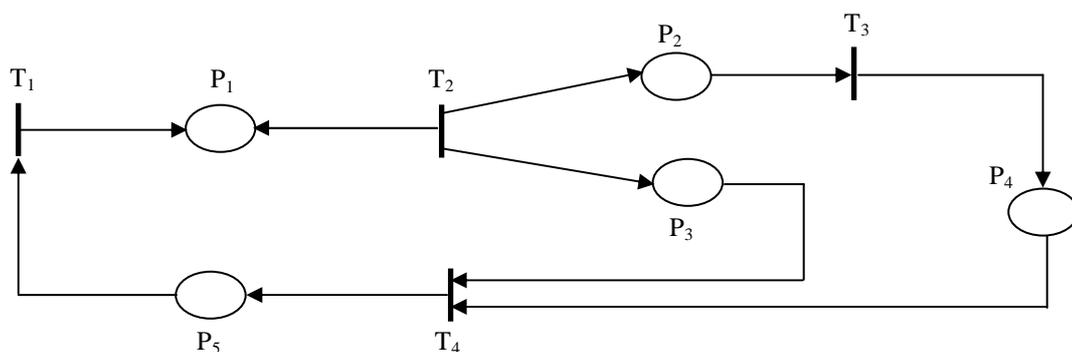


Figure 2.7. Circuits élémentaires.

Une boucle (P, T) est telle que $P \in {}^0T$ et $P \in T^0$. En d'autres termes, P est à la fois une place d'entrée et une place de sortie de T . Les boucles sont d'une grande utilité car elles permettent de modéliser le fait qu'une pièce ne peut entrer dans une machine si une autre pièce est déjà en cours de transformation sur cette machine. Dans la figure 2.8, (P_2, T_2) est une boucle.

Un RdP est dit pur s'il ne contient pas de boucle.

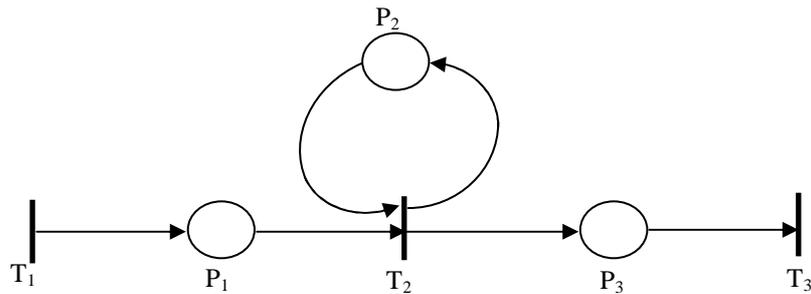


Figure 2.8. Réseau de Petri contenant une boucle.

2.3.4. Réseaux de Petri à capacité finie

Un RdP à capacité finie est un RdP dans lequel le marquage de chaque place P est limité à une quantité $Q(P)$. Dans un RdP à capacité finie, une transition T est franchissable si et seulement si, en plus des conditions habituelles, les marquages des places de sortie n'excèdent pas leur capacité après tirage.

Par exemple, la transition T de la figure 2.9 n'est pas franchissable car, après tirage, le marquage de la place P_3 serait porté à 5, alors que la capacité est 4.

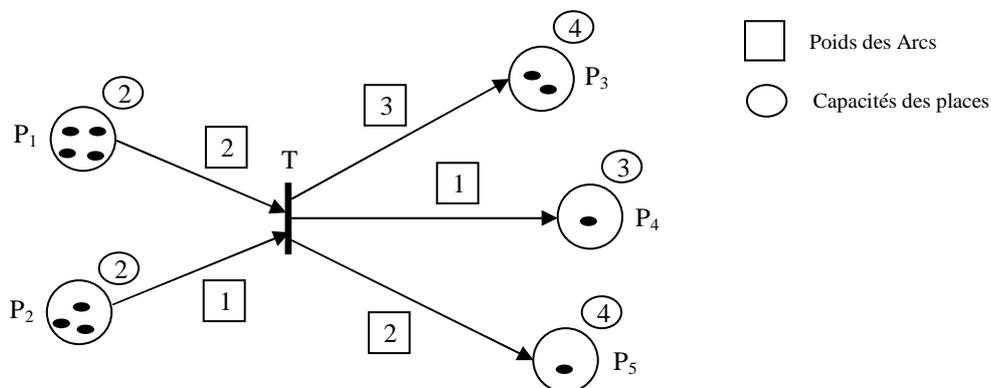


Figure 2.9. Transition non franchissable dans un réseau à capacité finie.

2. 3. 5. Verrou et trappe

Dans un RdP ordinaire (c'est-à-dire où chaque arc est pondéré à 1), un ensemble P_s de places est un verrou (ou impasse) si toute transition qui a une place de sortie dans P_s a au moins une place d'entrée dans P_s . La figure 2.10 schématise cette situation, dans laquelle P_s constitue un verrou. Par conséquent, tout franchissement de transition qui introduit un jeton dans une place du verrou en retire un d'une place du verrou. Par contre il est possible que certaines transitions aient une place d'entrée dans le verrou sans pour autant y avoir une place de sortie. Dans ce cas, le verrou peut se vider de ses jetons, (Proth et Xie, 1995).

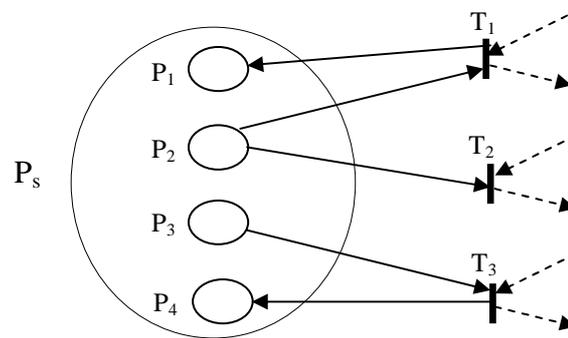


Figure 2.10. Exemple de verrou.

2. 4. Arbre des marquages atteignables, arbre de recouvrement et graphe de recouvrement

Considérons un RdP $PN = (N, M_0)$. L'objet de l'arbre des marquages atteignables est de découvrir tous les marquages que l'on peut atteindre à partir de M_0 . L'arbre de recouvrement est dérivé du précédent. Il permet, moyennant l'acceptation de certaines pertes d'information, de limiter la taille de l'arbre lorsque le nombre d'états du système n'est pas fini. L'arbre des marquages atteignables et l'arbre de recouvrement sont donc des outils d'analyse des RdP. Il s'agit d'arborescences dont les nœuds sont les marquages atteignables à partir de M_0 , et dont chaque arc représente le tirage d'une transition. La racine de l'arborescence représente M_0 .

Pour illustrer cette notion, considérons l'exemple simple de la figure 2.11 dans lequel tous les arcs sont pondérés à 1, (Proth et Xie, 1995).

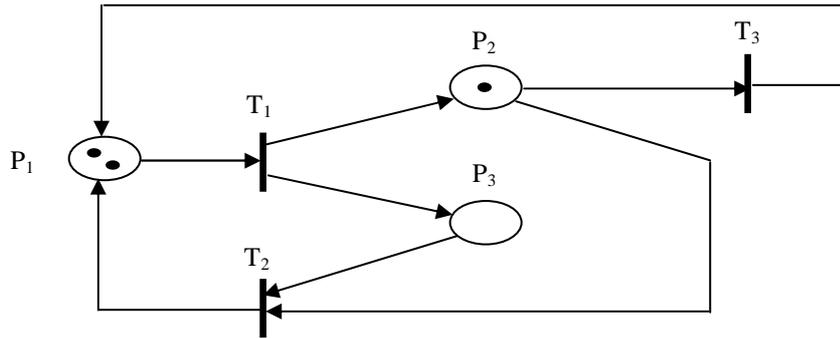


Figure 2.11. Réseau de Petri ordinaire.

Le marquage initial de ce réseau est $M_0 = [2,1,0]$. Partant de ce marquage, les transitions T_1 et T_3 peuvent être tirées. Ces tirages conduisent respectivement aux marques $M_1^1 = [1,2,1]$ et $M_1^2 = [3,0,0]$: c'est le premier niveau de l'arbre des marquages atteignables. Partant de M_1^1 , il est possible de franchir T_1 , T_2 ou T_3 qui conduisent respectivement aux marquages $M_1^2 = [0,3,2]$, $M_2^2 = [2,1,0]$ et $M_1^2 = [2,1,1]$. Partant de M_1^2 , on ne peut franchir que T_1 , ce qui conduit au marquage $M_2^4 = [2,1,1]$, qui est en fait le marquage M_2^3 . Nous venons de construire le second niveau de l'arbre des marques atteignables. On procède de même pour le troisième niveau. Enfin, on peut déduire qu'un arbre des marquages atteignables peut, dans la plupart des cas, se développer indéfiniment.

2. 5. Matrice d'incidence et équation d'état

La matrice d'incidence $U = [u_{ij}]$, $i = 1, 2, \dots, n$; $j = 1, 2, \dots, q$, d'un RdP pur est définie de la manière suivante :

$$u_{ij} = \begin{cases} W(t_j, p_i) & \text{Si } t_j \in p_i^0 \\ -W(p_i, t_j) & \text{Si } t_j \in p_i^0 \\ 0 & \text{Si non} \end{cases}$$

n est le nombre de places et m est le nombre de transitions du RdP considéré. $W(x,y)$ représente le poids de l'arc (x,y) .

2. 6. Propriétés comportementales et structurelles

Nous distinguons les propriétés comportementales et les propriétés structurelles. Les premières dépendent à la fois de la structure du RdP et du marquage initial, les secondes dépendent uniquement de la structure du RdP. En termes de systèmes de production, nous pouvons dire que les propriétés comportementales dépendent de la structure du système et de son état initial, alors que les propriétés structurelles ne dépendent que de la structure du système.

2. 6. 1. Propriétés comportementales

2. 6. 1. 1 Atteignabilité

La dynamique du système de production modélisé par les réseaux de Petri est représentée par l'évolution des marquages obtenus, à partir du marquage initial M_0 , par le franchissement de transitions, $R(M_0)$ représente l'ensemble des marquages qu'il est possible d'atteindre (nous dirons "atteignables") à partir de M_0 .

Dans l'étude d'un système dynamique, il est souvent souhaitable de savoir si un état donné peut être atteint ou, à l'inverse, si un état non désiré risque d'être atteint. Ce problème est appelé problème d'atteignabilité.

Définition 1

Etant donné un RdP $PN = (P, T, A, W, M_0)$ et un marquage M , le problème d'atteignabilité consiste à vérifier s'il existe une séquence de transitions σ qui transforme M_0 en M ou plus simplement si $M \in R(M_0)$.

2. 6. 1. 2. Vivacité et blocage

Dans un système de production, il est fréquent que des activités manufacturières se déroulent en parallèle. Cela exige la synchronisation de ces activités et le partage des ressources. Nous savons qu'une mauvaise synchronisation et un partage inadapté des ressources conduisent à une utilisation inefficace du système, mais peuvent également se

traduire par un blocage total ou partiel des activités. Un système de production bien conçu doit pouvoir éviter ces problèmes.

Définition 2

Une transition T d'un RdP $PN = (P, T, A, W, M_0)$ est dite vivante si elle peut être franchie quel que soit le marquage atteint, i.e. :

$$\forall M \in R(M_0), \exists M' \in R(M) \text{ tel que } t \text{ soit franchissable pour } M'.$$

Définition 3

Un RdP $PN = (P, T, A, W, M_0)$ est dit vivant si chacune de ses transitions est vivante.

Définition 4

Un marquage atteignable M d'un RdP $PN = (P, T, A, W, M_0)$ est un blocage si aucune transition n'est franchissable à partir de M .

Définition 5

Un RdP $PN = (P, T, A, W, M_0)$ est dit sans blocage si aucun de ses marquages atteignables $M \in R(M_0)$ n'est un blocage.

2. 6. 1. 3. Bornitude

Certaines places d'un RdP, modèle d'un système de production, représentent des zones de stockage. D'autres places contiennent des jetons qui représentent des ressources de fabrication. Il est souvent souhaitable de savoir si le nombre de jetons dans ces places est limité : cela permet de dimensionner le système de production correspondant, ou de découvrir certaines erreurs de conception.

Définition 6

a) Une place P d'un RdP $PN = (P, T, A, W, M_0)$ est dite k -bornée si le nombre de jetons dans cette place ne dépasse jamais k , i.e. $M(p) \leq k, \forall M \in R(M_0)$.

b) Une place P est dite bornée si elle est k -bornée pour un certain nombre entier $k > 0$.

Définition 7

a) Un RdP $PN = (P, T, A, W, M_0)$ est dit k -borné si le nombre de jetons dans chaque place ne dépasse pas k , i.e. $M(P') \leq k, \forall P' \in P$ et $\forall M \in R(M_0)$.

b) Un RdP est dit borné s'il est k -borné pour un certain nombre entier $k > 0$.

2. 6. 1. 4. Réversibilité et états d'accueil

La gestion des aléas est particulièrement importante pour le bon fonctionnement d'un système de production. En pratique, les aléas, sont nombreux et fréquents. Citons par exemple les pannes des machines et des moyens de transport, la rupture des outils, les dysfonctionnements dus à la qualité insuffisante de la matière première, les retards de livraison, etc....

Après avoir solutionné ces dysfonctionnements, on se ramène habituellement à un état précis qui permet de relancer le système. En termes de RdP, ce problème a trait à la propriété de réversibilité.

Définition 8

Un RdP $PN = (P, T, A, W, M_0)$ est dit réversible si l'on peut toujours revenir au marquage initial quelque soit le marquage atteint, i.e. $M_0 \in R(M)$ quel que soit $M \in R(M_0)$.

Définition 9

Un marquage M_a d'un RdP $PN = (P, T, A, W, M_0)$ est dit état d'accueil s'il peut être atteint à partir de tous les marquages atteignables, i.e. $M_a \in R(M)$ quel que soit $M \in R(M_0)$.

2. 6. 2. Propriétés structurelles

Les propriétés structurelles ne dépendent que de la structure du RdP. En termes de systèmes de production, les propriétés structurelles dépendent de la configuration du système, et non de la manière dont il géré.

Définition 10

Un RdP $N = (P, T, A, W)$ est dit structurellement vivant s'il existe un marquage initial M_0 tel que le RdP marqué PN = (N, M_0) soit vivant.

D'après cette définition, un RdP vivant est structurellement vivant. Mais la réciproque est fausse.

Définition 11

Un RdP $N = (P, T, A, W)$ est dit structurellement borné si le RdP marqué PN = (N, M_0) est borné quelque soit le marquage initial M_0 .

Suivant cette définition, un RdP structurellement borné est borné. Mais la réciproque n'est pas vraie.

Définition 12

Un RdP $N = (P, T, A, W)$ est dit conservatif s'il existe un vecteur colonne X qui associe à chaque place P' un poids entier positif $x(P') \geq 0$ de telle sorte que :

$$X^t \cdot M = X^t \cdot M_0, \quad \forall M \in R(M_0), \quad \text{où } X = [x(P_1), x(P_2), \dots, x(P_q)]^t.$$

Définition 13

Un RdP $N = (P, T, A, W)$ est dit répétitif s'il existe un marquage initial M_0 et une séquence σ de transitions franchissables dans laquelle chaque transition apparaît un nombre limité de fois.

D'après cette définition, un RdP structurellement vivant est répétitif. Mais la réciproque est fausse. Cette remarque implique que la répétitivité est une condition nécessaire de la vivacité structurelle. Par conséquent, elle est également une condition nécessaire de la vivacité.

Définition 14

Un RdP est dit consistant s'il existe un marquage initial M_0 et une séquence de transitions franchissables σ qui contient au moins une fois chaque transition et dont le franchissement conduit à nouveau au marquage initial M_0 , i.e. $M_0 \xrightarrow{\sigma} M_0$.

D'après cette définition, un RdP vivant et réversible est consistant. Mais la réciproque n'est pas vraie.

2. 7. Exemples de modélisation

Nous allons présenter dans cette section des exemples de modélisation par les RdP de quelques systèmes intervenant dans les systèmes automatisés de stockage/déstockage, tel que les convoyeurs et les zones de stockages.

2. 7. 1. Convoyeurs

Les convoyeurs sont fréquemment utilisés en pratique pour les déplacements des produits lourds ou des produits à fort débit. La figure 2.12 représente un système de production utilisant un convoyeur entre deux machines.

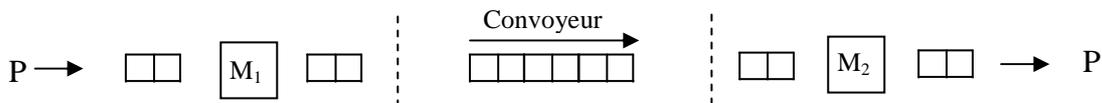


Figure 2.12. Deux machines reliées par un convoyeur.

Un convoyeur peut être décomposé en cantons. Un canton est une portion de convoyeur qui peut être occupée par au plus un produit à la fois. Les produits quittent le convoyeur dans l'ordre dans lequel ils arrivent (First-In-First-Out).

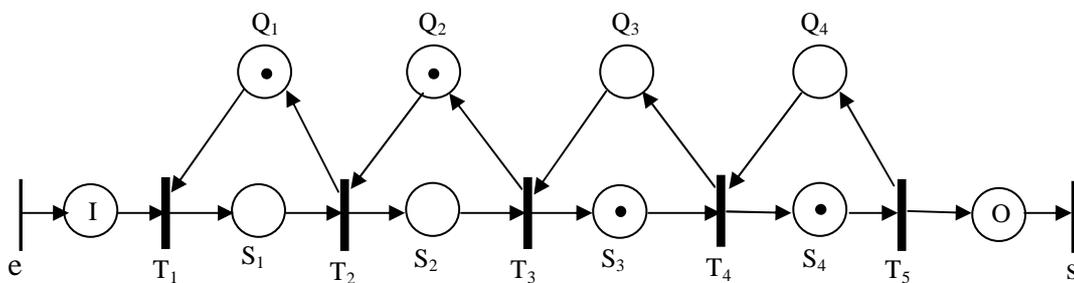


Figure 2.13. Modèle RdP d'un convoyeur comportant quatre cantons.

La figure 2.13 donne un modèle RdP d'un convoyeur qui comporte quatre cantons. Dans ce modèle, les transitions T_1 , T_2 , T_3 , T_4 et T_5 représentent les déplacements des produits d'un canton à un autre. Chaque couple de places (Q_i, S_i) contient un seul jeton. Le canton i est libre si le jeton figure dans la place Q_i et il ne l'est pas si le jeton figure dans S_i . Le marquage de la figure 2.13 correspond au cas où les cantons 3 et 4 sont occupés et les cantons 1 et 2 sont libres.

2. 7. 2. Système de stockage

Les systèmes de stockage sont utilisés dans les systèmes de fabrication afin d'absorber les aléas et de réguler la fabrication. Les systèmes de stockage les plus utilisés sont les stocks tampons dédiés, les zones de stockage partagées et les stocks FIFO (First-In-First-Out).

2. 7. 2. 1. Stocks tampons dédiés

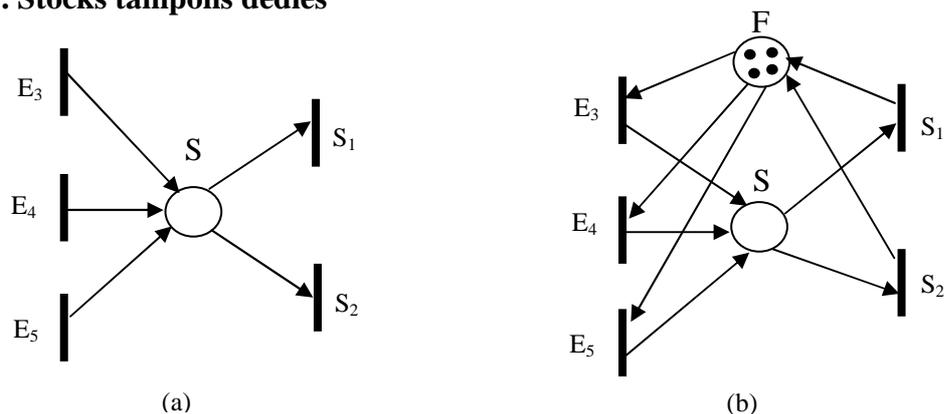


Figure 2.14. Modèles RdP de deux stocks tampons dédiés.

Ils sont utilisés pour stocker les produits entre deux opérations consécutives. La figure 2.14.a est le modèle d'un stock tampon avec capacité illimitée tandis que la figure 2.14.b représente celui d'un stock tampon à capacité limitée. Cette capacité est de quatre unités dans le cas de la figure b. Dans ce modèle, le nombre de jetons figurant dans la place F indique la capacité de stockage restante et le nombre de jetons figurant dans la place S représente le nombre de produits stockés.

2. 7. 2. 2. Zone de stockage commune

Une zone de stockage commune peut être utilisée par plusieurs produits différents. La capacité globale est limitée. La figure 2.15 représente le modèle RdP d'une zone de stockage partagée entre deux produits. Nous remarquons que le modèle de la figure 2.15 se généralise aisément dans le cas où les volumes des produits sont différents et la capacité de stockage est exprimée en volume.

Dans ce modèle, le nombre de jetons figurant dans la place S_1 (resp. S_2) correspond au nombre de produits P_1 (resp. P_2) dans la zone de stockage.

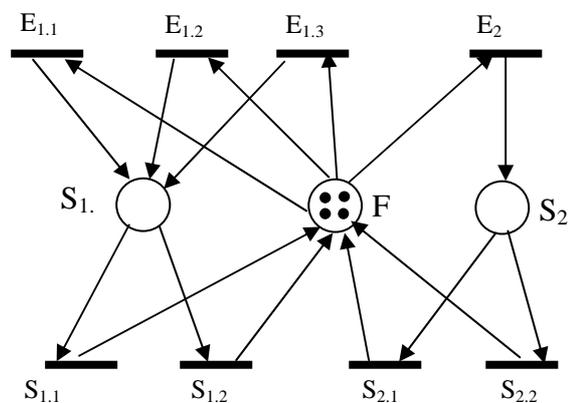


Figure 2.15. Modèle RdP d'une zone de stockage commune.

2. 7. 2. 3. Stocks FIFO

Dans les stocks FIFO, le produit qui arrive le premier est déstocké le premier. Le convoyeur peut également être considéré comme un stock FIFO. La figure 2.16 donne le modèle RdP d'un stock FIFO dédié avec une capacité limitée à 4 et partagé par deux types de produits. Initialement, les places Q_1 , Q_2 , Q_3 et Q_4 contiennent chacune un jeton, ce qui signifie que le stock est vide. Supposons que la transition $T_{1.1}$ soit franchie (i.e. un produit P_1 est stocké). Le jeton dans Q_1 disparaît et un jeton apparaît dans $S_{1.1}$. La transition $T_{2.1}$, représentant le stockage d'un produit P_2 , ne peut être franchie qu'après le franchissement de $T_{1.2}$. Pour la même raison, la transition $T_{2.2}$ (resp. $T_{2.3}$, $T_{2.4}$) ne peut être franchie qu'après le franchissement de $T_{1.3}$ (resp. $T_{1.4}$, $T_{1.5}$). Le franchissement de $T_{2.4}$ permet alors le franchissement de $T_{2.5}$.

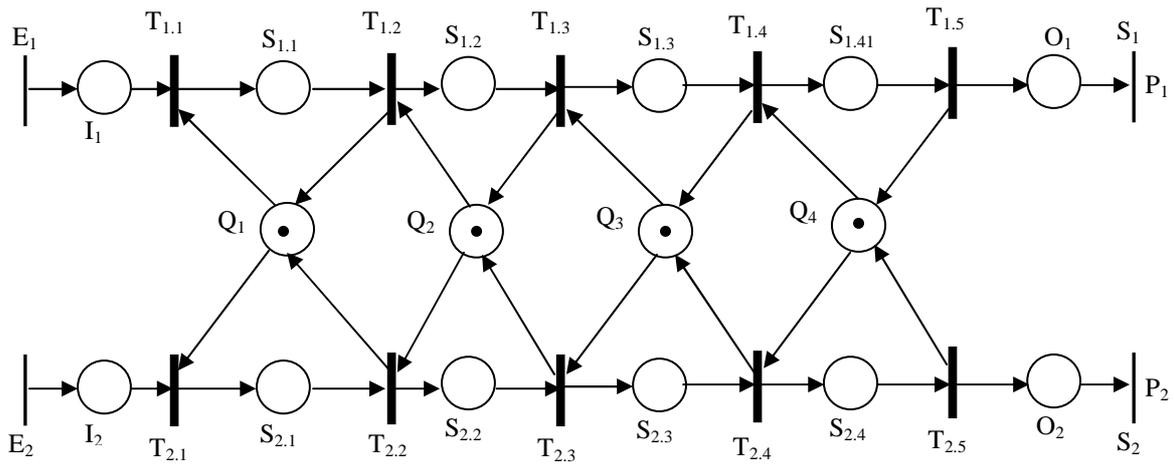


Figure 2.16. Modèle RdP d'un stock FIFO partagé par deux produits.

2. 8. Abréviations et extensions des Réseaux de Petri

Les abréviations correspondent à des représentations simplifiées, utile pour alléger le graphisme, mais auxquelles on peut toujours faire correspondre un réseau de Petri ordinaire. Les extensions correspondent à des modèles auxquels des règles de fonctionnement ont été ajoutées, afin d'enrichir le modèle initial, ce qui permet d'aborder un plus grand nombre d'application. Plusieurs types de RdP ont été développé tel que les RdP continus, les RdP hybrides et les RdP colorés. Nous allons nous focaliser dans la suite sur les réseaux de Petri colorés étant donné qu'ils seront utilisés pour la modélisation du système de stockage/déstockage (chapitre 3).

2. 9. Réseaux de Petri colorés

La modélisation d'un système peut mener à des réseaux de Petri de taille trop importante rendant leur manipulation et/ou leur analyse difficile. La question est alors de modifier (étendre) la modélisation par RdP de façon à obtenir des modèles RdP de plus petite taille. Cette question a motivé l'introduction des abréviations telles que dans les réseaux de Petri colorés (RdPC). En effet, une taille trop importante peut découler du fait que l'on ne peut pas distinguer entre elles les différentes marques d'une place. Plusieurs marques dans une place peuvent modéliser un certain nombre de pièces identiques dans un stock. Si le stock contient plusieurs types de pièces, des places supplémentaires doivent être introduites pour sa

modélisation. L'idée de base utilisée dans les réseaux de Petri colorés est d'associer alors un identificateur ou une "couleur" à chaque marque de la place et l'information est représentée par l'ensemble place-couleur.

2. 9. 1. Présentation intuitive et notion de couleur

La figure 2. 17. représente deux systèmes identiques. Dans chaque système, un chariot peut se déplacer soit vers la gauche, soit vers la droite. Dès que le chariot atteint une extrémité, il repart vers l'autre.

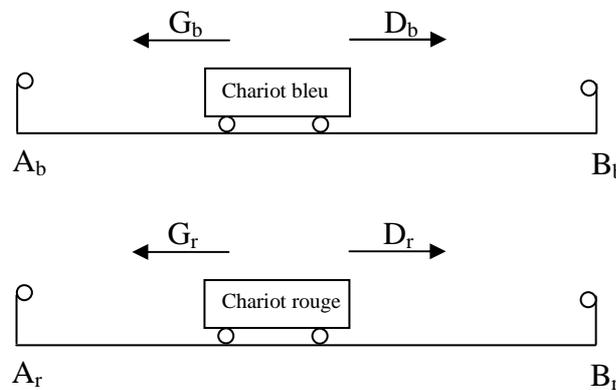


Figure 2.17. Déplacement de deux chariots.



Figure 2.18. Les réseaux de Petri associés aux chariots a- bleu – b- rouge.

En observant le modèle de la figure 2.18, on voit que celui-ci est composé de deux RdP identiques. Si l'on fusionne les deux RdP en un seul, on obtient le modèle de la figure 2.19.a.

Ce réseau ne décrit pas correctement le fonctionnement des deux chariots, car la connaissance du marquage est insuffisante pour donner l'état de chacun d'eux. Par exemple, lorsque les deux places P_1 et P_2 sont marquées, on ne sait pas quel chariot se déplace vers la droite et quel chariot se déplace vers la gauche. Pour modéliser correctement le système des deux chariots, il faut pouvoir distinguer les marques entre elles dans le réseau de la figure 2.19.a. On associe une couleur (il s'agit d'un identificateur qu'on appelle couleur) à chaque marque, soit la marque r pour le chariot rouge et la marque b pour le chariot bleu (figure 2.19.b). Les places P_1 et P_2 peuvent alors contenir des marques de couleur appartenant à l'ensemble $\{b, r\}$. La transition T_1 est validée par rapport à la couleur r s'il y a une marque r dans la place P_1 . Le franchissement de T_1 consiste à retirer une marque r de la place P_1 et à ajouter une marque r à la place P_2 . La transition peut être également être franchie par rapport à la couleur b . On associe à la transition T_1 l'ensemble de couleurs $\{b, r\}$.



Figure 2.19. -a- Modèle ne décrivant pas correctement le fonctionnement –b- RdP coloré associé aux deux chariots.

2. 9. 2. Définition des réseaux de Petri colorés

Un RdP Coloré N est un sextuplet $N = \langle P, T, \text{Pré}, \text{Post}, M_0, C \rangle$. P est l'ensemble des places, T est l'ensemble des transitions, et $C = \{C_1, C_2, \dots\}$ est l'ensemble des couleurs. Pré et Post sont des fonctions relatives aux couleurs de franchissement. M_0 est le marquage initial. Un RdPC se différencie d'un RdP généralisé par l'adjonction d'un ensemble de couleurs. Une couleur $C_k = \langle c_{k1}, c_{k2}, \dots, c_{kn} \rangle$ pourra être notée soit globalement C_k , soit par le n -uplet qui la définit.

2. 9. 3. Evolution du marquage dans un réseau de Petri coloré

Le marquage $M(P_i)$ d'une place P_i représente le nombre de marques de chaque couleur contenue dans la place P_i . Le marquage initial de la figure 2.19.b est donné par le marquage des places P_1 et P_2 , soit $M(P_1) = \langle b \rangle + \langle r \rangle$ et $M(P_2) = 0$. Cela signifie que le marquage de P_1 se compose d'une marque de couleur $\langle b \rangle$ et une marque de couleur $\langle r \rangle$.

2. 9. 3. 1. Transition validée

Soit $C(T_j)$ l'ensemble des couleurs associées à la transition T_j . Cette transition peut être franchie par rapport à l'une quelconque de ces couleurs (le nombre de couleurs dans $C(T_j)$ correspond au nombre de transitions du réseau déplié, c'est-à-dire du réseau non coloré équivalent, qui ont été regroupées en une seule). Soit C_k une couleur quelconque de $C(T_j)$, et soit M un marquage courant du RdP coloré. La transition T_j est validée par rapport à la couleur C_k pour le marquage M , si et seulement si, le nombre de marques contenues dans toute place P_i en amont de T_j est supérieur ou égal à $\text{Pré}(P_i, T_j/C_k)$.

c'est-à-dire :

$$M(p_i) \geq \text{Pré}(P_i, T_j/C_k) \quad \forall P_i \in {}^0T_j$$

$\text{Pré}(P_i, T_j/C_k)$ est l'image de la couleur C_k par la fonction poids de l'arc reliant la place P_i à la transition T_j . 0T_j est l'ensemble des places d'entrée de la transition T_j .

Exemple : Dans la figure 2.19.b la transition T_1 est validée par rapport à la couleur $\langle b \rangle$ car $\text{Id}(\langle b \rangle) = \langle b \rangle$ et la place P_1 contient une marque de couleur $\langle b \rangle$. Elle est également validée par rapport à la couleur $\langle r \rangle$.

2. 9. 3. 2. Franchissement d'une transition validée

Une transition T_j validée par rapport à une couleur C_k peut être franchie. Son franchissement qui sera notée T_j/C_k , consiste à effectuer simultanément les opérations suivantes :

1. On retranche à toute place P_i en amont de T_j une quantité de marques égale à $\text{Pré}(P_i, T_j / C_k)$.
2. On ajoute à toute place P_i en aval de T_j une quantité de marques égale à $\text{Post}(P_i, T_j / C_k)$.

Le marquage M' obtenu après le franchissement de la transition T_j par rapport à la couleur C_k se déduit du marquage M par la relation suivante :

$$M'(p) = M(P_i) + \text{Post}(P_i, T_j / C_k) - \text{Pré}(P_i, T_j / C_k) \quad \forall P_i$$

Dans l'exemple de la figure 2.19.b, le franchissement de la transition T_1 qui est validée par rapport à la couleur $\langle b \rangle$ conduit d'une part à retirer une marque de couleur $\langle b \rangle$ de la place P_1 car $\text{Id}(\langle b \rangle) = \langle b \rangle$ et d'autre part à ajouter une marque de couleur $\langle b \rangle$ à la place P_2 car $\text{Id}(\langle b \rangle) = \langle b \rangle$.

2. 9. 3. 3. Séquence de franchissements

Le franchissement d'une transition T_1 par rapport à une couleur C_{h1} (noté T_1/C_{h1}), à partir d'un marquage M_1 , donne un marquage M_2 . Ce marquage M_2 peut également valider une transition T_2/C_{h2} dont le franchissement donne un marquage M_3 , et ainsi de suite. On obtient alors une séquence de franchissement $S = T_1/C_{h1}. T_2/C_{h2} \dots T_k/C_{hk}$ qui fait passer du marquage M_1 au marquage M_{k+1} que l'on calcule par l'expression suivante :

$$M_{k+1}(P_i) = M_1(P_i) + \sum_{j=1}^k \text{Post}(P_i, T_j / C_{hj}) - \text{Pré}(P_i, T_j / C_{hj})$$

Exemple : Pour le RdP de la figure de 2.20, la séquence $S1 = T_1/C_1. T_2/C_1. T_2/C_2$ est une séquence de franchissements. Cette séquence peut être franchie une infinité de fois puisqu'elle ramène au marquage initial.

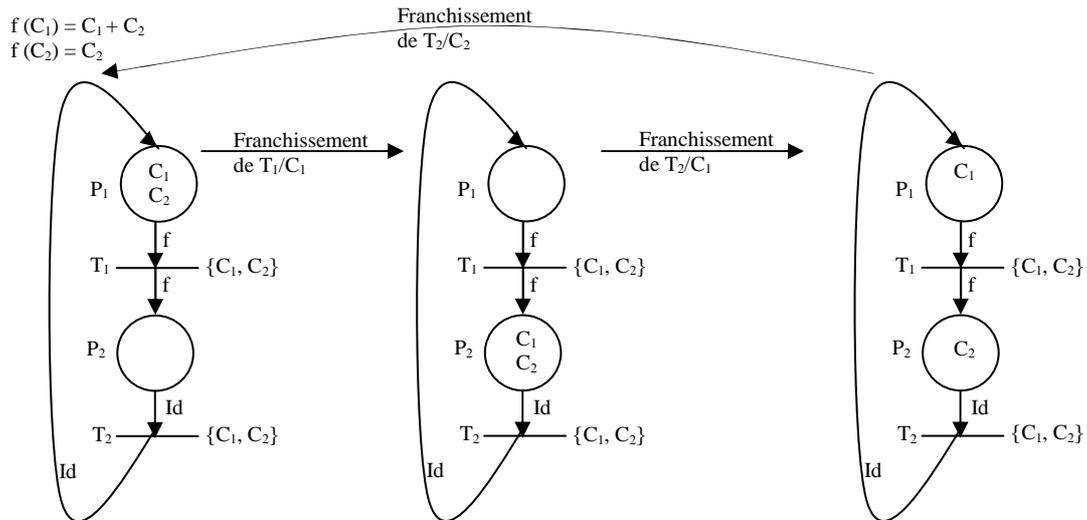


Figure 2.20. Séquence de franchissements.

2. 9. 3. 4. Fonctions

Les fonctions réalisent une transformation linéaire des couleurs associées à une transition. Une fonction quelconque est définie en calculant l'image de chaque couleur de la transition. Mais une fonction peut être également prédéfinie, comme nous le verrons dans la suite. Il existe un noyau de fonctions élémentaires rencontrées dans une grande diversité de modèles. Ces fonctions peuvent intervenir sous leur forme simple, mais il est également possible de construire de nouvelles fonctions en composant des fonctions simples entre elles. Un catalogue de fonctions prédéfinies est présenté dans le tableau de la figure 2.21.

Les fonctions Id, Déc, Succ et Préc sur la figure 2.21 ont toutes pour argument des couleurs simples. On peut aussi définir des fonctions dont l'argument est une couleur complexe. C'est le cas par exemple des fonctions Succ_j et Préc_j. Ces deux fonctions correspondent respectivement à incrémenter et décrémenter la j^{ème} composante. Il arrive que, dans une place donnée, on n'ait pas besoin de toute l'information apportée par la couleur complexe ; on utilise alors une fonction projection, ce qui consiste à supprimer une ou plusieurs composantes. Par exemple pour un doublet $\langle c_i, c_j \rangle$, on peut définir une projection supprimant la première composante, soit $\text{Proj}_1(\langle c_i, c_j \rangle) = \langle c_j \rangle$ ou une projection supprimant la deuxième composante, soit $\text{Proj}_2(\langle c_i, c_j \rangle) = \langle c_i \rangle$, etc.

Ensemble de départ	Ensemble d'arrivée	Fonction	Définition	Commentaire
C	C	Id	$\text{Id}(\langle c_i \rangle) = \langle c_i \rangle$	Identité
C	$\langle \bullet \rangle$	Déc	$\text{Déc}(\langle c_i \rangle) = \langle \bullet \rangle$	Décoloration
C	C	Succ	$\text{Succ}(\langle c_i \rangle) = (\langle c_{i+1} \rangle) \quad [n]$	Successeur modulo n
C	C	Préc	$\text{Préc}(\langle c_i \rangle) = (\langle c_{i-1} \rangle) \quad [n]$	Précédent modulo n
C^m	C^m	Succ_j	$\text{Succ}_j(\langle c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im} \rangle)$ $= \langle c_{i1}, c_{i2}, \dots, c_{(i+1)j}, \dots, c_{im} \rangle [n]$	Précédent pour un m-uplet
C^m	C^m	Préc_j	$\text{Préc}_j(\langle c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im} \rangle)$ $= \langle c_{i1}, c_{i2}, \dots, c_{(i-1)j}, \dots, c_{im} \rangle [n]$	Précédent pour un m-uplet
C^m	C^{m-1}	Proj_j	$\text{Proj}_j(\langle c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im} \rangle)$ $= \langle c_{i1}, \dots, c_{i(j-1)}, c_{i(j+1)}, \dots, c_{im} \rangle$	Projection sur la $j^{\text{ème}}$ composante
Remarque : C est un ensemble de n couleurs : $C = \{ \langle c_1 \rangle, \langle c_2 \rangle, \dots, \langle c_i \rangle, \dots, \langle c_n \rangle \}$ et C^m est le produit cartésien à l'ordre m de C				

Figure 2.21. Catalogue de fonctions prédéfinies.

2. 9. 3. 5. Exemple d'application des réseaux de Petri colorés

Nous considérons un système de transport à structure FIFO (First In First Out) comme représenté dans la figure 2.22.

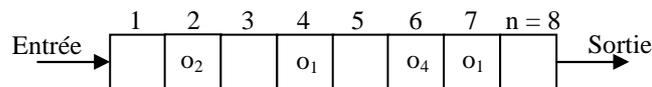


Figure 2.22. Système de transport FIFO.

Le convoyeur linéaire représenté à la figure 2.22 peut transporter simultanément q types d'objets appelés o_1, o_2, \dots, o_q , la capacité du convoyeur étant n (ce qui correspond à n emplacements). Pour cet exemple $n = 8$. Il y a un objet o_2 sur l'emplacement $n^{\circ}2$, un objet o_1 sur l'emplacement $n^{\circ}4$, un objet o_4 sur l'emplacement $n^{\circ}6$ et un objet o_1 sur l'emplacement $n^{\circ}7$. Nous nous proposons de représenter par un RdPC l'état du système physique.

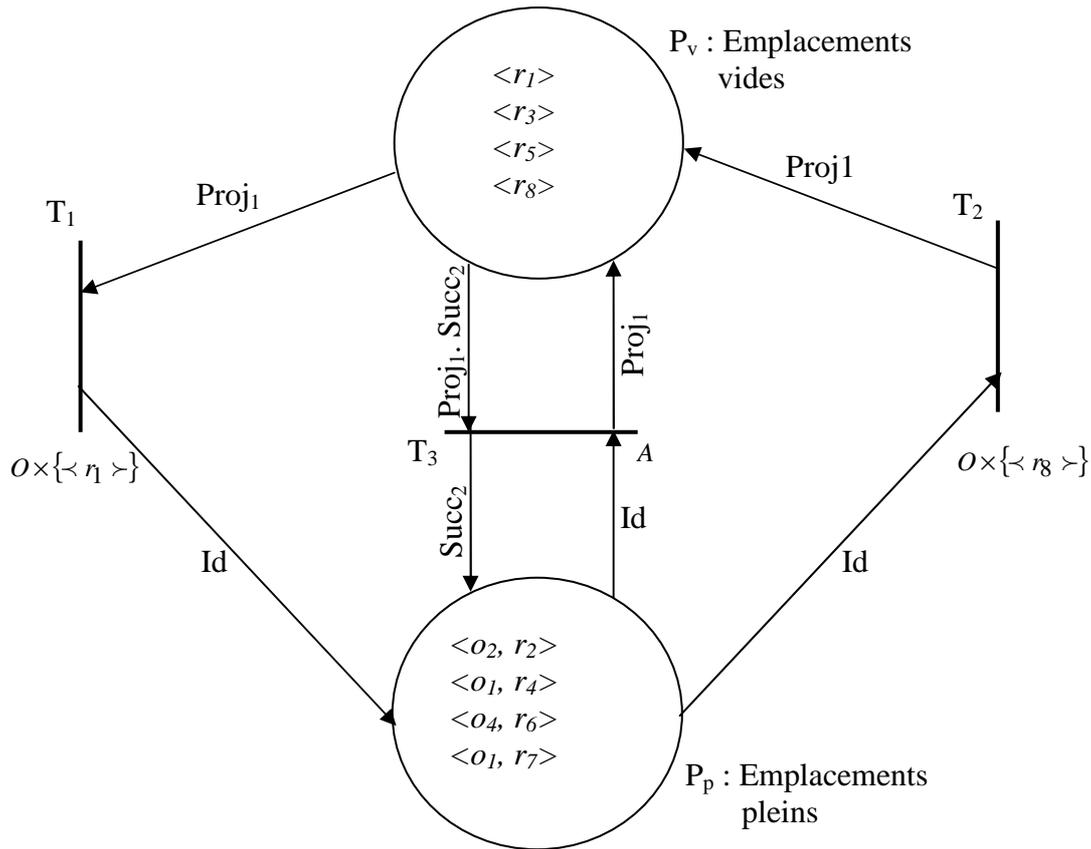


Figure 2.23. RdP d'un système de transport FIFO.

Le RdPC représentant ce système est donné à la figure 2.23. Les ensembles de couleurs simples sont les suivants : $O = \{ \langle o_j \rangle, j \in [1, q] \}$ ensemble des objets, et $R = \{ \langle r_i \rangle, i \in [1, n] \}$ ensemble des emplacements du convoyeur. L'ensemble de couleurs A associé à la transition T_3 est : $O \times \{ r_1, \dots, r_{n-1} \}$. Une marque de couleur $\langle o_j, r_i \rangle$ dans la place P signifie qu'il y a un objet j dans l'emplacement i du convoyeur. Dans le modèle de ce système on constate l'intervention d'une fonction composée : $\text{Proj}_1.\text{Succ}_2$. Celle-ci provient de la composition de la fonction Succ_2 qui gère la progression des objets dans le système et de la fonction Proj_1 qui gère les emplacements. L'utilisation de fonctions composées permet de réduire l'ensemble des fonctions utilisées tout en conservant leur signification physique.

Le marquage initial correspondant à un convoyeur vide est :

$$M_0(P_p) = 0 \quad \text{et} \quad M_0(P_v) = \sum_{i=1}^n \langle r_i \rangle$$

Le marquage indiqué sur la figure 2.23 correspond à l'état indiqué sur la figure 2.22. A partir de ce marquage la transition d'entrée T_1 du système de transport est franchissable par rapport à la couleur $\langle o_j, r_1 \rangle$ quel que soit l'indice j car la place P_v contient la marque $\text{Proj}_1(\langle o_j, r_1 \rangle) = \langle r_1 \rangle$, c'est-à-dire que l'emplacement 1 du système est vide. Le franchissement de T_1 dépose une marque de couleur $\langle o_j, r_1 \rangle$ dans la place P_p et retire une marque $\langle r_1 \rangle$ de la place P_v ; l'emplacement 1 du système contient maintenant un objet j . Cet objet sera transféré d'un emplacement à un autre par le franchissement de la transition T_3 . Pour le marquage de la figure 4.23, la place P_p contient une marque $\langle o_2, r_2 \rangle$, la transition T_3 peut être franchie par rapport à cette couleur si la place P_v contient une marque de couleur $\text{Proj}_1.\text{Succ}_2(\langle o_2, r_2 \rangle) = \langle r_3 \rangle$, ce qui signifie que l'emplacement 3 doit être vide (ce qui est le cas dans l'exemple). Le franchissement de T_3 dépose une marque de couleur $\text{Proj}_1(\langle o_2, r_2 \rangle) = \langle r_2 \rangle$ dans la place P_v (l'emplacement 2 est libéré) et une marque $\text{Succ}_2(\langle o_2, r_2 \rangle) = \langle o_2, r_3 \rangle$ dans la place P_p (l'emplacement 3 es occupé par un objet o_2). Ce transfert va se poursuivre tant qu'il y a de la place dans le système et jusqu'à ce que l'objet arrive dans l'emplacement n . Il est alors évacué du système par le franchissement de la transition de sortie T_2 . Ce système modélisé par un RdP non coloré nécessiterait $(q + 1).n$ places et $q.(n + 1)$ transitions. Le modèle obtenu est, dans sa structure, indépendant de la longueur du convoyeur et de la nature des objets qu'il transporte.

2. 10. Conclusion

Dans ce chapitre nous avons présenté les notions de base des réseaux de Petri et de leur application dans la modélisation des systèmes de production. Ces réseaux de Petri permettent de modéliser et de visualiser des comportements incluant du parallélisme, de la synchronisation et du partage de ressources. Nous avons présenté un certains nombre de leurs propriétés et définitions. Il existe un grand nombre de résultats théoriques sur les réseaux de Petri les rendant ainsi un outil très puissant de modélisation et d'analyse. Cependant quand le nombre de places augmente le réseau devient plus compliqué à étudier. Dans ce contexte, les réseaux de Petri colorés permettent une description compacte et structurée en associant plus d'information au réseau de Petri. Ce modèle n'est qu'une abréviation du RdP ordinaire, donc il est possible de vérifier toutes ses propriétés. On dit qu'on le valide. La deuxième partie de ce chapitre a été consacrée à ces réseaux de Petri colorés qui seront utilisés pour la modélisation du système automatisé de stockage/déstockage dans le chapitre 3.

Chapitre 3

Modélisation et pilotage du système automatisé de stockage/déstockage à convoyeur gravitationnel

3. 1. Introduction

L'évolution croissante des besoins d'une entreprise fait que la conception d'un système de stockage/déstockage est de plus en plus orientée vers des familles de produits et non vers un seul type de produit. Pour répondre concrètement aux problèmes qui se posent dans de tels systèmes, il est indispensable de disposer d'un modèle reflétant la dynamique réelle de ces systèmes et d'un système de pilotage permettant d'atteindre des objectifs désirés.

Ce chapitre porte sur la modélisation et le pilotage du système automatisé de stockage/déstockage à convoyeur gravitationnel. Dans ce cadre, nous commençons par la définition de la réactivité d'un système. Ensuite, nous présentons le système de décision avec ses différents niveaux. Nous passons ensuite à la description détaillée du fonctionnement de l'AS/RS à convoyeur gravitationnel. Nous traitons ensuite la modélisation du système, basée sur les réseaux de Petri colorés déjà introduits dans le chapitre précédent. Ce modèle sera intégré dans un système de contrôle en boucle fermée. Nous terminons ensuite par une conclusion.

La modélisation et le contrôle de l'AS/RS utilisant les RdPC, présentée dans ce chapitre, figurent dans la publication au *Journal Européen des Systèmes Automatisés* (Hachemi et Alla, 2007a).

3. 2. Réactivité des systèmes de stockage/déstockage

Une exigence importante du client est de recevoir sa livraison dans les délais impartis et ce quel que soit le carnet de commande (variable). Satisfaire une telle exigence impose au système de stockage d'être réactif, c'est-à-dire capable de répondre rapidement et

économiquement à un changement (introduction d'une nouvelle commande, changement des types de produits etc.)

La réactivité d'un système de production (le système de stockage/déstockage en est un cas particulier) est définie comme l'aptitude à répondre (réagir) dans un temps requis aux changements de son environnement interne ou externe (situation nouvelle, perturbation, sollicitation, ...).

La réactivité d'un système de stockage/déstockage impose une vision dynamique des événements qui se passent dans le système. Afin d'assurer cette propriété de réactivité du système de stockage/déstockage, trois fonctions annexes s'avèrent nécessaires :

- **Une fonction d'observation** qui collecte les variables nécessaires au suivi, afin de connaître l'état courant du système (disponibilité et état des produits).
- **Une fonction de surveillance** qui détecte (suite au résultat d'une observation) et interprète les écarts et les changements entre le plan prévisionnel et le plan courant par anticipation.
- **Une fonction de correction** qui tente à tout instant de corriger les écarts entre ces plans, ce qui implique un ordonnancement dynamique.

3. 3. Système de décision

Le système de décision définit des cadres de décisions dans lesquels il précise les opérations à accomplir, les grandeurs sur lesquelles il est possible d'agir, les méthodes à suivre ainsi que les limites. Il s'articule autour d'une organisation en boucle (figure 3.1) constituée des quatre activités suivantes :

- **La planification**, qui consiste à mettre en œuvre des techniques d'ordonnancement. Intégrée dans le processus global de conduite, la planification propose une affectation pour les différentes opérations, dans le temps et l'espace. C'est un moyen unique et incontournable pour s'assurer du respect des objectifs fixés. Néanmoins les méthodes de calcul employées pour sa mise en œuvre sont souvent pénalisantes dans le temps.

- **Le lancement**, qui répartit et transmet les ordres au système en tenant compte de l'état des entités du système.
- **Le suivi**, qui recueille l'ensemble des événements survenant dans système et qui met à jour une image interne du système opérant.
- **La réaction**, qui ajuste les actions à prendre par rapport aux changements induits.

La conduite en temps réel d'un système de stockage/déstockage est une tâche complexe qui requiert des connaissances dans les domaines de l'informatique, de l'automatique, de la production, de la communication homme - machine etc. Sa difficulté est due :

- A la complexité du problème d'ordonnement.
- A la nécessité de résoudre le problème d'ordonnement dans un contexte dynamique.
- Au problème du suivi temps réel.
- A l'intégration du système de conduite (pilotage) dans le système global de gestion de stock.

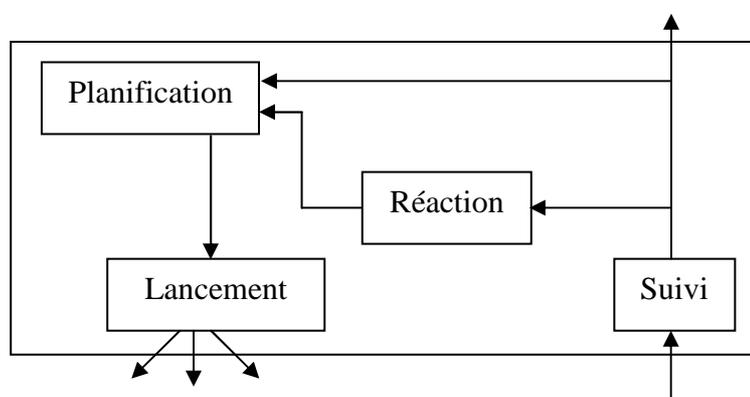


Figure 3.1. Le modèle conceptuel du système de décision

3. 3. 1. Niveaux décisionnels

Dans le cadre de la gestion des chaînes logistiques, et plus largement des systèmes industriels, trois niveaux de décisions sont généralement définis (Anthony, 1965). Il s'agit des

niveaux stratégique, tactique et opérationnel. Ces niveaux sont repris par un grand nombre d'auteurs (Ganeshan, 1999 ; Shapiro, 1999 ; Huang et al., 2003 ...).

3. 3. 1. 1. Niveau stratégique

Ganeshan (1999) décompose le niveau stratégique suivant les problèmes étudiés, à savoir :

1. La définition d'objectifs communs pour l'ensemble des acteurs de la chaîne.
2. La conception ou structure physique de la chaîne (choix des partenaires, délocalisation,...).
3. La relance de la compétitivité, par exemple par la planification stratégique.
4. L'évolution de la nature stratégique du management de la chaîne logistique.

D'après Miller (2001), les décisions de planification de la chaîne logistique concernant le niveau stratégique sont les suivantes :

- Localisations, missions et relations des usines et entrepôts. (i.e., conception des infrastructures et réseaux).
- Positionnement des nouvelles usines et fermetures de sites.
- Positionnement de nouveaux entrepôts et fermetures.
- Niveaux de capacité des usines et entrepôts.
- Acquisitions de biens technologiques et d'équipements pour usines et entrepôts.
- Conception d'usines et d'entrepôts.
- Répartition entre biens achetés et utilisation de ressources d'un tiers (par ex. décisions de sous-traiter).
- Réseaux de transports et prestataires de transport.
- Choix du type de gestion de production (par ex. fabrication à la commande, fabrication sur stock).

Ces décisions sont de première importance pour l'entreprise et concernent une vision à long terme. Ainsi, compte tenu de l'évolution rapide des marchés, ces décisions vont nécessiter une grande prise de risque de la part de l'entreprise. En effet, le choix d'accroître ou non sa capacité de production en installant une nouvelle usine sera réellement judicieux si la demande, au moment où l'usine sera en fonctionnement, correspond à celle prévue au moment de la prise de décision.

3. 3. 1. 2. Niveau tactique

Le niveau décisionnel tactique va s'intéresser aux décisions à moyen et long terme qui devront être mises en application pour développer la stratégie décidée par l'entreprise.

D'après Miller (2001) les décisions de planification de la chaîne logistique concernant le niveau tactique sont les suivantes :

- Affectation des capacités de production aux familles de produits par usine, souvent en considérant des périodes temporelles de taille « moyenne » (par exemple, trimestriellement).
- Taux d'utilisation des capacités planifiées en fabrication, par usine et au niveau du réseau.
- Besoins en main d'œuvre (niveaux nominal et heures supplémentaires), allocation des ressources d'approvisionnement aux usines, centres de distribution et détaillants par région ou pays.
- Gestion des transferts intersites (par ex. entre centres de distribution),
- Plans d'investissements et de déploiement des stocks.
- Modes de transports et choix des transporteurs.

L'ensemble de ces décisions ne représente qu'une partie de toutes les décisions pouvant être prises au niveau tactique. Il est toutefois possible de dégager un certain nombre de caractéristiques communes entre ces décisions comme, par exemple, la portée temporelle donnée à chacune de ces décisions qui est d'environ 12 mois et qui peut s'étendre jusqu'à 2 ans (Galasso, 2007). Elle tient compte essentiellement des délais liés au cycle complet de

fabrication, englobant le délai maximum d'approvisionnement, de production, et de mise sur le marché.

3. 3. 1. 3. Niveau opérationnel

Les décisions prises au *niveau opérationnel* auront une portée plus limitée dans l'espace et dans le temps. A ce niveau, les décisions tactiques vont être déclinées de manière à ce qu'elles soient applicables au niveau d'un site de fabrication ou un entrepôt.

De plus, Giard (2003) inclut dans le cadre des décisions opérationnelles les décisions liées au suivi de la production en *Temps Réel*. Ces décisions concernent l'évolution, le suivi et le contrôle d'éléments du système de production jouissant d'une certaine autonomie, tels que des magasins automatiques, des machines outils à commande numériques etc.

De manière plus exhaustive, Miller (2001) propose l'ensemble de décisions associées au niveau opérationnel suivant :

- Ordonnancement quotidien et hebdomadaire au niveau des unités de stockages (*Stock Keeping Units, SKU*) incluant la gestion des priorités.
- Equilibrage et correction des stocks à court terme.
- Traitement et ordonnancement des commandes clients.
- Ordonnancement et gestion des entrepôts.
- Ordonnancement de la main d'œuvre pour la fabrication et l'entreposage,
- Ordonnancement des tournées de véhicules.
- Sélection des transporteurs pour les chargements non groupés.
- Supports logistiques pour les lancements individuels (par ex. lancements d'approvisionnements directs spécifiques).

Au regard de ces différents niveaux de décisions, notre problématique nous porte vers le niveau décisionnel opérationnel. Ainsi, nous nous intéressons à l'ordonnancement dynamique

et temps réel du système de stockage/déstockage et nous excluons de notre étude les décisions stratégiques et tactiques.

3. 4. Ordonnancement des opérations de déstockage

L'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles et de contraintes portant sur l'utilisation et la disponibilité des ressources requises par les tâches. Un ordonnancement décrit l'ordre d'exécution des tâches et l'allocation des ressources au cours du temps afin de satisfaire un ou plusieurs critères d'optimisation (Lopez, 2001). L'ordonnancement en temps réel consiste en un ensemble de décisions de lancements de fabrication des produits et de séquençement d'exécution des opérations au niveau de postes. Cette fois la décision doit être prise en temps réel. Le niveau commande doit gérer les informations reçues et maîtriser la conduite des robots, outils, ou convoyeurs. Comme nous nous plaçons dans le contexte des systèmes de stockage, les notions de systèmes de production présentées ci-après (Duta, 2006), restent valables. Dans ce cas on parlera d'opération de stockage/déstockage au lieu d'opération de fabrication. Dans notre système, il s'agit d'ordonner les opérations de déstockages, c'est-à-dire définir l'ensemble des opérations de déstockages effectuées par la machine de déstockage pour répondre aux demandes de clients.

3. 4. 1. Les éléments d'ordonnancement

D'après la définition citée ci-dessus, il ressort qu'un problème d'ordonnancement est constitué principalement de quatre éléments: *les tâches, les ressources, les contraintes et les critères d'optimisation.*

3. 4. 1. 1. Les tâches

Les tâches sont toutes les opérations élémentaires de travail à effectuer pour la fabrication d'un produit.

Une tâche est un travail élémentaire dont la réalisation nécessite une certaine durée et un nombre de ressources. Une tâche *i* est localisée dans le temps par une date de début *si* (start

time) ou de fin ci (completion time) dont la réalisation est caractérisée par une durée ti . On a ainsi $ci=si+ti$. Certaines contraintes techniques ou économiques peuvent conduire à associer aux tâches des dates de début au plus tôt ri (ready time) ou des dates de fin au plus tard di (due date).

3. 4. 1. 2. Les ressources

L'exécution des différentes tâches nécessite la mise en œuvre d'un ensemble de moyens techniques et d'opérateurs humains. Cet ensemble représente donc les ressources indispensables à la réalisation des tâches durant les intervalles de disponibilité. Une ressource est un moyen, technique ou humain, dont la disponibilité limitée ou non est connue *a priori*. Une ressource est ainsi caractérisée par sa capacité supposée constante pendant le travail. Il y a plusieurs types de ressources. Si après son utilisation la ressource est à nouveau disponible avec la même capacité on parle d'une ressource renouvelable. Par contre, si après son utilisation la ressource est disponible avec une capacité inférieure ou nulle on parle de ressource consommable. Certaines ressources sont capables de réaliser plusieurs tâches en parallèle, elles sont dites cumulatives (une station de travail composée de plusieurs machines est une ressource cumulative). Les ressources sont supposées disjonctives si elles ne permettent de réaliser qu'une seule opération à la fois (un ouvrier s'occupant de plusieurs machines est une ressource disjonctive).

3. 4. 1. 3. Les contraintes

Les contraintes expriment des restrictions sur les valeurs des variables de décision. Leur prise en compte permet d'avoir un ordonnancement réalisable. Les contraintes qui sont liées directement au système de production et à ses performances sont les contraintes temporelles et les contraintes de ressources.

Les contraintes temporelles reflètent la non disponibilité continue des ressources et la nécessité de délivrer les produits en respectant les délais. Les contraintes temporelles peuvent aussi définir des relations de précedence entre les opérations. Ce sont surtout des contraintes technologiques qui imposent un ordre d'exécution aux opérations.

Les caractéristiques des ressources peuvent induire des contraintes indiquant les conditions de passage des tâches. Dans le cas d'une ressource consommable, seules les tâches ayant une consommation inférieure ou égale à la capacité de la ressource à l'instant considéré peuvent être exécutées. Pour une ressource renouvelable ce type de contraintes exprime la limitation de sa capacité. Pour assurer le respect des contraintes de ressources on doit éviter le chevauchement temporel de certaines tâches. Ce type d'action est appelé *séquencement*. Le problème de séquencement est induit par le problème d'ordonnancement initial.

3. 4. 1. 4. Les critères d'optimisation

L'objectif de l'ordonnancement est d'optimiser (maximiser ou minimiser) une fonction d'évaluation en respectant un certain nombre de contraintes. Les critères d'optimisation dans l'ordonnancement classique sont le coût, le délai de livraison et la qualité de processus.

Le coût des ressources est un critère assez rare parce que le coût machine est invariant.

Ce coût est pris en compte dans la phase de conception du système. Le coût de production est souvent utilisé comme critère d'optimisation (minimisation). Les arrêts et les temps d'inactivité augmentent les coûts de production. L'objectif de l'ordonnancement est ainsi de minimiser le temps mort entre les opérations et donc d'assurer une activité continue pour chaque poste de travail ou chaque machine.

Le délai évalue en termes de temps les performances d'un système. Respecter le délai consiste à définir un ordre de fabrication durant une période donnée.

La qualité est rarement associée à l'évaluation d'un ordonnancement. Toutefois, la fluidité d'un flux de fabrication est associée à un facteur de qualité. La qualité d'un processus de fabrication est obtenue en minimisant le nombre des produits inachevés pendant une période de production. Malheureusement, ce critère est rarement utilisé dans l'ordonnancement théorique.

3. 4. 2. Les étapes d'ordonnancement

L'objectif du problème d'ordonnancement est de fixer la date de début *si* pour chaque tâche *i*. Pour cela il faut déterminer l'ordre de passage de l'ensemble des travaux ou séquence

sur chaque machine. A partir de ces séquences plusieurs ordonnancements admissibles (localisant de manière absolue les opérations dans le temps) peuvent être obtenus selon le critère à optimiser. L'ordonnancement comme processus de décision a plusieurs étapes. Les plus importantes sont l'affectation des tâches aux postes et leur séquençement.

L'étape de **l'affectation des tâches** consiste à répartir les tâches aux divers postes de travail en fonction de ressources disponibles et des précédences entre les tâches. Dans cette étape l'affectation des tâches aux ressources n'est pas fixée a priori. A chaque tâche i est associé un ensemble de ressources capable de la réaliser. Pour chaque ressource la durée de réalisation t_i de la tâche est déterminée. Cette durée peut être variable ou fixe.

L'étape de **séquençement** est la **détermination d'un ordre de passage** (ou séquence) qui consiste à établir un ordre dans lequel les différentes tâches seront exécutées sur chaque poste de travail. Pour ce faire, on doit connaître la gamme complète et les durées opératoires pour chaque tâche. L'évaluation de l'ordre de passage repose sur plusieurs critères: la minimisation des coûts de travail et des stocks ou la maximisation des taux d'utilisation des postes de travail. Souvent il est préférable que cette étape définisse un ordre de passage qui donne des résultats satisfaisants dans des conditions réalistes, plutôt qu'un ordre qui donne des résultats théoriques optimaux, mais trop éloignés de la réalité.

Une fois réalisées l'affectation des tâches à chaque poste et la détermination du séquençement, on peut alors passer à la troisième étape qui consiste à **établir le calendrier de déstockage**. Ce dernier donne une date précise pour le lancement de chaque opération à chaque poste de travail.

La quatrième étape de l'ordonnancement est le **lancement du déstockage**, soit l'exécution du calendrier.

La phase de **contrôle de l'ordonnancement** permet le suivi de l'exécution du travail par le biais de la collecte de données sur les commandes en cours.

3. 5. Système automatisé de stockage/déstockage à convoyeur gravitationnel

L'AS/RS à convoyeur gravitationnel est composé d'un rack profond regroupant un ensemble de casiers pouvant contenir plusieurs produits placés les uns à la suite des autres.

Chaque casier est muni d'un convoyeur gravitationnel à base de rouleaux ou roues libres inclinées de façon à permettre le glissement des produits d'un bout à l'autre du casier, donc d'une face à l'autre du rack. Les produits sont stockés d'un côté du rack (face de stockage) par la machine de stockage (MS), ils glissent sur les convoyeurs gravitationnels, et ils sont déstockés de l'autre côté du rack (face de déstockage) par la machine de déstockage (MD). Les machines de stockage et de déstockage peuvent se déplacer simultanément sur deux axes. Ces deux axes forment le plan x - y parallèle aux deux faces du rack. Une station de dépôt est située sur la face de stockage, où la machine de stockage récupère les produits à stocker. Une station de livraison est située sur la face de déstockage, où la machine de déstockage dépose les produits pour la livraison. Un convoyeur de re-stockage incliné en sens inverse relie la machine de déstockage à la machine de stockage. Il permet de transférer les produits à restocker vers la station de dépôt. La figure 3.2 donne la structure d'un tel AS/RS industriel.

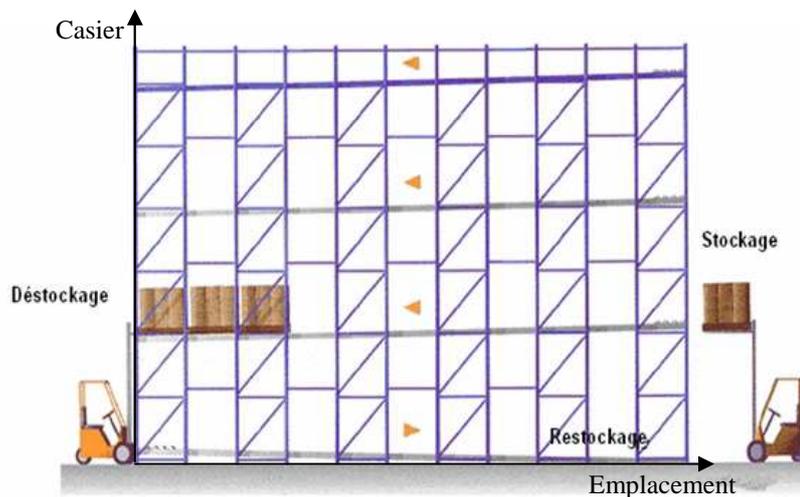


Figure 3.2. Vue de profil d'un AS/RS à convoyeur gravitationnel

La figure 3.3 présente la constitution détaillée d'un rack à convoyeur gravitationnel. Le rack est composé de plusieurs casiers, et chaque casier contient plusieurs emplacements dans lesquels les produits peuvent être placés. Les casiers du rack sont numérotés de 1 à m et les emplacements d'un casier sont numérotés de 1 à q . La figure 3.4 illustre la disposition des produits à l'intérieur d'un casier du rack. Par exemple, pour déstocker un produit demandé qui se trouve au quatrième emplacement d'un casier, il faut d'abord déstocker un à un, les trois produits qui le précèdent et les envoyer au convoyeur de re-stockage pour être réintroduits dans le rack. Le produit demandé sera déposé dans la station de livraison.

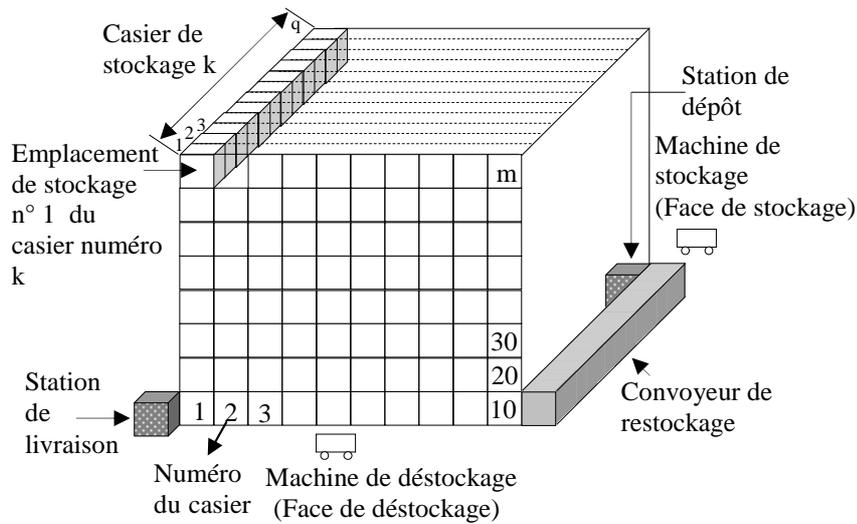


Figure 3.3. Constitution typique d'un rack à convoyeur gravitationnel

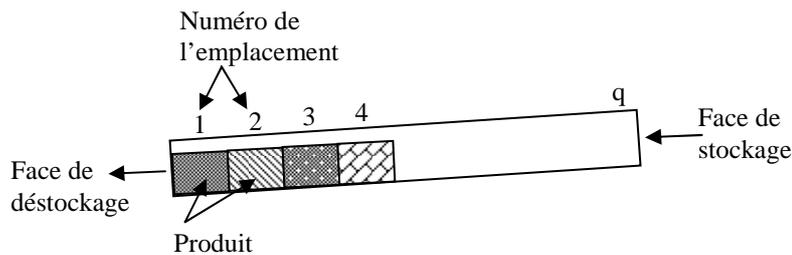


Figure 3.4. Disposition des produits à l'intérieur d'un casier

Chaque casier contient plusieurs types de produits stockés selon une politique de stockage. Nous considérons, dans ce travail, le stockage aléatoire. C'est-à-dire qu'un produit peut être stocké dans n'importe quel emplacement du rack. Etant donné des demandes de clients, le problème consiste à déterminer les emplacements de produits à déstocker afin de les satisfaire en un temps minimum.

L'élaboration de la demande clients consiste à regrouper plusieurs demandes en un seul lot que nous représenterons formellement par un vecteur. Chaque élément de ce vecteur représente la somme de toutes les quantités de produits pour le type de produit correspondant.

Par exemple, pour deux demandes de clients Q_1 et Q_2 telles que :

$$Q_1 = [q_{11} \ q_{21} \ \dots \ q_{n1}]^T$$

$Q_2 = [q_{12} \ q_{22} \ \dots \ q_{n2}]^T$ où n est le nombre de types de produits existants.

Le vecteur de demande résultant sera égal à :

$$Q = Q_1 + Q_2 = [q_{11} + q_{12} \ q_{21} + q_{22} \ \dots \ q_{n1} + q_{n2}]^T = [q_1 \ q_2 \ \dots \ q_n]^T$$

Avec :

q_{ij} : représente la quantité du produit de type i demandé par le client j .

q_i : représente la quantité totale du produit de type i pour l'ensemble des clients.

Le regroupement des demandes en un seul lot permet de rechercher une trajectoire optimale en terme de nombre de cycles de déstockage, englobant tout les produits demandés et en ne choisissant pas forcément les produits qui ont l'indice d'emplacement le plus faible. Il est évident que dans le cas où les demandes sont traitées une par une, le choix optimal se porte forcément sur le produit dont l'indice d'emplacement est le plus faible. Cette différence est due au fait que pour l'optimisation dans un lot, on fait intervenir la fonction coût par casier (Chapitre 4).

Deux modes de fonctionnement peuvent être envisagés. Dans le premier mode, l'élaboration de la demande se fait à une période fixe. Dans ce cas la taille du lot est constante. Dans le deuxième mode, l'élaboration de la demande se fait à une période adaptative afin de permettre plus de réactivité au système quand le niveau de demande est élevé. Dans ce cas, la taille du lot traité est variable. Ces deux modes sont détaillés dans la sous-section 3. 6. 3.

3. 6. Structure du système de pilotage

La structure de pilotage de l'AS/RS à convoyeur gravitationnel permet de générer les décisions de déstockage à partir de la demande des clients et de l'état du rack. Pour modéliser l'état du rack nous avons développé un modèle Réseau de Petri Coloré capable de suivre les mouvements des produits à l'intérieur des casiers. Les décisions de contrôle sont calculées par un bloc d'optimisation. Selon l'intensité de la demande, l'élaboration de la demande se fait soit à période fixe soit à période adaptative.

3. 6. 1. Modèle Réseau de Petri coloré

Les Réseaux de Petri Colorés (RdPC) sont une abréviation des réseaux de Petri qui permet de modéliser des systèmes complexes, comme décrit dans le chapitre 2. Dans un RdPC, aux places sont associées des identificateurs ou "couleurs". Chaque transition peut être franchie de différentes manières représentées par les différentes couleurs de franchissement que l'on associe à la transition. Des fonctions sont associées aux arcs, elles permettent de définir la couleur des jetons à retirer des places d'entrée et celle des jetons à ajouter aux places de sortie. Deux types de relations sont utilisés :

1. La première entre la couleur du jeton à retirer, contenu dans la place d'entrée de la transition et la couleur de franchissement associée à cette dernière.
2. La deuxième entre la couleur de franchissement de la transition et la couleur du jeton à déposer dans la place de sortie de celle-ci.

Afin de représenter la dynamique de l'AS/RS à convoyeur gravitationnel, un modèle réseau de Petri coloré a été développé. En effet, le système est de gestion complexe, du fait que l'état à chaque instant du rack peut être modifié par trois opérations différentes : le stockage d'un nouveau produit, le déstockage d'un produit demandé, et éventuellement le restockage des produits non demandés. Dans le modèle, les produits à restocker seront considérés comme des produits à stocker, une fois arrivés à la station de dépôt via le convoyeur de restockage. De plus, la taille du système est importante et il faut prendre en compte les déplacements des produits d'un emplacement à un autre à l'intérieur des casiers. Le modèle RdPC du système est présenté dans la figure 3.5. L'état du système est donné à tout instant par le marquage du RdPC.

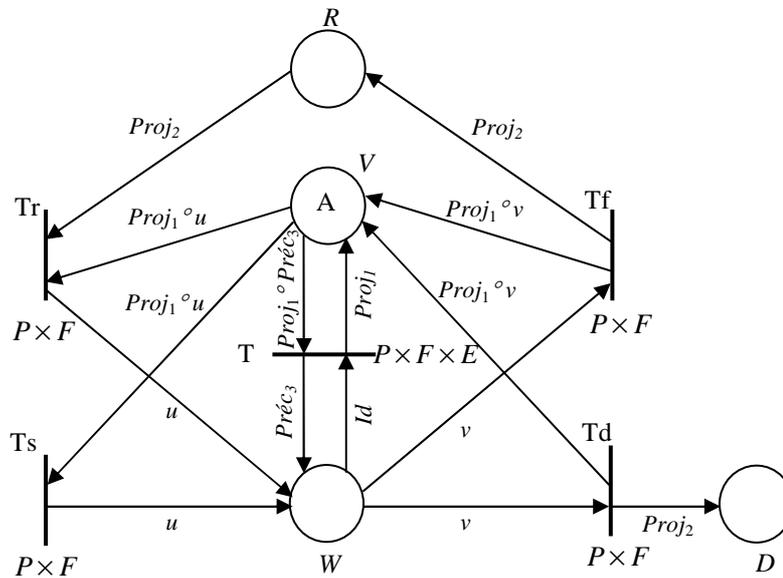


Figure 3.5. Le modèle RdPC du système de stockage/déstockage à Convoyeur gravitationnel

Nous tenons à préciser que le développement d'un modèle RdPC est motivé par deux objectifs. Le premier est que les RdPC permettent bien de représenter la dynamique complexe du système. Le deuxième consiste dans le fait que le formalisme des réseaux de Petri permet de vérifier et valider le modèle (vivacité, réversibilité, atteignabilité par exemple). Ce deuxième objectif nous a semblé très important et c'est le besoin de pouvoir établir des preuves formelles de propriétés qui nous a conduit au choix des réseaux de Petri. D'autant plus que la validation des modèles FIFO existe abondamment dans la littérature, on peut citer par exemple (Alla, 1987). Nous garantissons ainsi le fonctionnement correct du modèle en dehors de toute simulation.

Le modèle RdPC de la figure 3.5 représente les casiers du rack comme des files FIFO. En effet, le fonctionnement d'un casier peut être assimilé à une file FIFO, du fait que le produit qui entrera le premier dans le casier sortira en premier, même s'il est conduit une nouvelle fois au restockage. Les ensembles de couleurs qui gèrent le RdPC sont :

$P = \{ \langle pi \rangle, i \in [1, n] \}$ Ensemble de types de produits.

$F = \{ \langle fk \rangle, k \in [1, m] \}$ Ensemble des casiers du rack.

$E = \{ \langle ej \rangle, j \in [1, q] \}$ Ensemble des emplacements d'un casier

Le marquage initial $A = M_0(V) = \sum_{k=1}^m \sum_{j=1}^q \langle fk, ej \rangle$

Aux arcs sont associés des fonctions simples ou des fonctions composées. Elles sont définies comme suit :

La fonction u introduit un produit pi dans le dernier emplacement d'un casier fk :

$$u : P \times F \rightarrow P \times F \times \{eq\}$$

$$(pi, fk) \mapsto u(pi, fk) = \langle pi, fk, eq \rangle$$

La fonction v extrait un produit pi du premier emplacement d'un casier fk :

$$v : P \times F \rightarrow P \times F \times \{e1\}$$

$$(pi, fk) \mapsto v(pi, fk) = \langle pi, fk, e1 \rangle$$

Dans le RdPC de la figure 3.5 la transition T_s est validée par rapport à la couleur $\langle pi, fk \rangle$ car la place V contient la couleur $\text{Proj}_1^\circ u(\langle pi, fk \rangle) = \text{Proj}_1(pi, fk, eq) = \langle fk, eq \rangle$, l'emplacement q du casier est vide. Le tir de T_s dépose un jeton de couleur $u(\langle pi, fk \rangle) = \langle pi, fk, eq \rangle$ dans la place W et retire un jeton $\langle fk, eq \rangle$ de la place V , ainsi l'emplacement q du casier fk contient un produit de type i . Ce produit sera transféré d'un emplacement à un autre par le tir de la transition T . Supposons que la place W contienne un jeton $\langle pi, fk, ej \rangle$, la transition T sera franchie selon cette couleur si la place V contient un jeton de couleur $\text{Proj}_1^\circ \text{Préc}_3(\langle pi, fk, ej \rangle) = \langle fk, e_{j-1} \rangle$, l'emplacement $(j-1)$ doit être vide. Le tir de T dépose un jeton $\text{Proj}_1(\langle pi, fk, ej \rangle) = \langle fk, ej \rangle$ dans la place V (l'emplacement j est libéré) et un jeton $\text{Préc}_3(\langle pi, fk, ej \rangle) = \langle pi, fk, e_{j-1} \rangle$ dans la place W (l'emplacement $j-1$ est occupé). Ce transfert va continuer tant qu'il y a de la place dans le casier jusqu'à ce que le produit arrive à l'emplacement 1.

La transition T_r décrit un fonctionnement similaire à celui de la transition T_s mais pour le restockage d'un produit. Un produit peut être évacué d'un casier soit vers la station de livraison, soit vers le restockage (recyclage). Le tir de T_f dépose un jeton $\text{Proj}_2(pi, fk) = pi$ dans la place R et un jeton $\text{Proj}_1^\circ v(\langle pi, fk \rangle) = \text{Proj}_1(pi, fk, e1) = \langle fk, e1 \rangle$ dans la place V . Il retire un jeton $v(\langle pi, fk \rangle) = \langle pi, fk, e1 \rangle$ de la place W . Le tir de T_d évacue un produit vers la station de livraison. Ce tir dépose un jeton $\text{Proj}_2(pi, fk) = pi$ dans la place D et un jeton $\text{Proj}_1^\circ v(\langle pi, fk \rangle) = \text{Proj}_1(pi, fk, e1) = \langle fk, e1 \rangle$ dans la place V . En même temps il retire un jeton $v(\langle pi, fk \rangle) = \langle pi, fk, e1 \rangle$ de la place W .

L'interprétation des places et des transitions du RdPC est résumée dans le tableau de la figure 3.6.

Place	Couleur	Interprétation
<i>V</i>	$\langle fk, ej \rangle$	L'emplacement <i>ej</i> du casier <i>fk</i> est vide
<i>W</i>	$\langle pi, fk, ej \rangle$	L'emplacement <i>ej</i> du casier <i>fk</i> contient un produit de type <i>pi</i>
<i>D</i>	$\langle pi \rangle$	La station de livraison contient un produit de type <i>pi</i>
<i>R</i>	$\langle pi \rangle$	Un produit de type <i>pi</i> est recyclé
Transition	Couleur	Interprétation
<i>Ts</i>	$\langle pi, fk \rangle$	Introduction d'un produit <i>pi</i> dans l'emplacement <i>q</i> du casier <i>fk</i>
<i>Tr</i>	$\langle pi, fk \rangle$	Introduction d'un produit recyclé <i>pi</i> dans l'emplacement <i>q</i> du casier <i>fk</i>
<i>Td</i>	$\langle pi, fk \rangle$	Evacuation d'un produit de type <i>pi</i> de l'emplacement 1 du casier <i>fk</i> vers la station de livraison
<i>Tf</i>	$\langle pi, fk \rangle$	Evacuation d'un produit de type <i>pi</i> de l'emplacement 1 du casier <i>fk</i> vers le convoyeur de restockage
<i>T</i>	$\langle pi, fk, ej \rangle$	Déplacement d'un produit <i>pi</i> de l'emplacement <i>j</i> à l'emplacement <i>j-1</i>

Figure 3.6. Interprétation des places et transitions du modèle RdPC

3. 6. 2. Pilotage en boucle fermée

La figure 3.7 montre la boucle de pilotage de l'AS/RS. L'entrée de référence du système de contrôle est la demande clients élaborée périodiquement. Les valeurs q_1 à q_n représentent les quantités de produits (1 à n) demandées après le regroupement en lot.

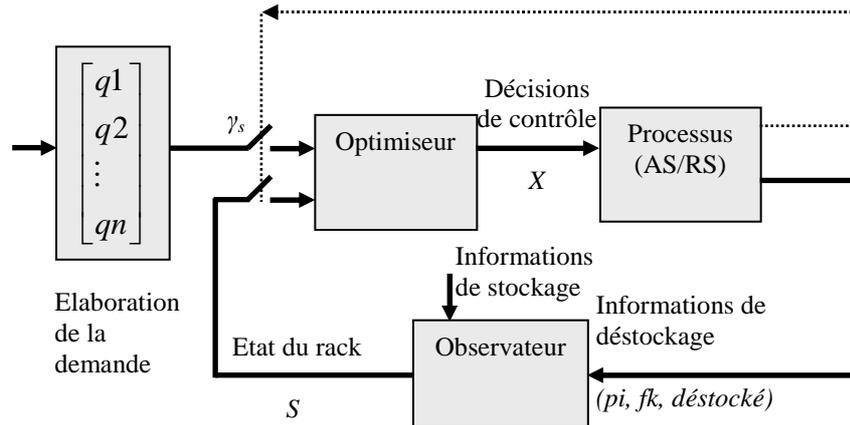


Figure 3.7. Schéma de contrôle de l'AS/RS

Le système de contrôle utilise un bouclage pour lire les informations issues des capteurs par le biais d'un observateur modélisé par un RdPC. Ceci permet l'actualisation de l'état du système après chaque opération induisant un changement d'état ; c'est à dire une opération de stockage, déstockage ou de restockage. Le déplacement des produits d'un emplacement à un autre est très rapide et en tous cas de durée négligeable par rapport à celles des machines de stockage et déstockage. Ainsi, la situation où la machine est prête alors que le produit désiré n'est pas encore arrivé sur la face de déstockage n'est pas possible. Une autre boucle (indiquée en pointillés dans la figure 3.7) est éventuellement utilisée pour adapter la période de traitement des demandes γ_s .

L'observateur reçoit les informations via les capteurs installés dans le rack. L'information $(p_i, f_k, stocké)$ indique une opération de stockage ou de restockage d'un produit de type p_i dans le casier f_k . Une entrée externe en direction de l'observateur permet de recevoir les informations concernant le stockage de nouveaux produits. Le bloc d'optimisation (optimiseur) reçoit en entrée, la demande périodique et l'état du rack S fourni par le modèle RdPC, et génère les décisions de contrôle en sortie.

Le rôle principal de l'optimiseur est d'ordonner les demandes de produits en déterminant une séquence spécifique des emplacements de casiers à déstocker. Ceci en tenant en compte de la demande des clients et de l'état du rack. Par conséquent, l'élément de commande, par exemple un Automate Programmable Industriel reçoit la séquence de déstockage générée par l'optimiseur pour exécuter les déplacements de la machine de déstockage.

Le temps de calcul de l'optimiseur doit être faible afin de réaliser un contrôle dynamique et temps réel. Pour chaque acquisition d'un lot de demandes, l'optimiseur réagit aux changements d'état dans le système en recalculant la variable de décision X . La méthode de génération de la variable de décision X est détaillée dans la section suivante.

3. 6. 3. Mode de traitement des demandes

En fonction du niveau de la demande, nous envisageons deux modes de fonctionnement. Le premier mode fonctionne à période d'échantillonnage des demandes fixe. Dans le deuxième mode, l'élaboration de la demande se fait à une période adaptative pour mieux réagir à un niveau de demande important.

3. 6. 3. 1. Fonctionnement à période d'échantillonnage fixe

Dans ce mode de fonctionnement, la période de traitement des demandes (période d'échantillonnage) γ_s est fixe et elle est choisit suffisamment grande par rapport à la somme du temps de calcul de l'optimiseur et du temps total nécessaire pour satisfaire le lot de demandes. Cette période est calculée sur la base de la limite supérieure du temps de transport de la machine de déstockage. Cette limite correspond au temps que met la machine pour atteindre le casier le plus éloigné de la station de livraison.

La période d'échantillonnage doit donc être $\gamma_s \geq qg \cdot (tm + tlu) + tc$.

$tm = \max\left(\frac{xm}{sh}, \frac{ym}{sv}\right)$, est le temps de transport de Tchebychev maximal (s), correspondant au casier le plus éloigné de la station de livraison, voir (Bozer et al., 1984).

x_m est l'abscisse (maximale) du dernier casier (m),

y_m est l'ordonnée (maximale) du dernier casier (m),

s_h est la vitesse horizontale de la machine de déstockage (ms^{-1}),

s_v est la vitesse verticale de la machine de déstockage (ms^{-1}).

q_g est la taille du lot, c'est-à-dire $q_g = q_1 + q_2 + \dots + q_n$. Dans ce mode, la taille du lot est considérée constante.

t_{lu} est le temps de chargement/déchargement d'un produit, supposé constant pour toutes les opérations de déstockage.

t_c est le temps de calcul du module d'optimisation.

Par conséquent, pour ce mode, il suffit de choisir $\gamma_s = q_g \cdot (tm + tlu) + t_c$.

3. 6. 3. 2. Fonctionnement à période d'échantillonnage adaptative

Dans ce cas la période d'échantillonnage γ_s de la demande est adaptée en fonction du temps effectif mis par la machine de déstockage pour terminer toutes les opérations de déstockage du lot. En fait, dans la majorité des cas, un lot de demande est satisfait dans un temps inférieur à la période γ_s du fait que γ_s est calculée sur la base de la limite supérieure du temps de déstockage d'un lot à taille fixe de demande. Afin d'adapter la période d'échantillonnage aux temps effectifs des tâches de contrôle, il est plus intéressant d'exploiter le temps restant jusqu'au prochain « top » de la période γ_s pour traiter un nouveau lot de demande surtout si le niveau de demande est important. Dans ce cas le retour en pointillés sur la figure 3.7 est utilisé pour activer une nouvelle acquisition d'un lot de demande sans attendre la fin de la période γ_s et par conséquent permettre de traiter plus de demandes par unité de temps. Dans ce mode la taille du lot de demande n'est pas constante. Ce second mode permet une meilleure réactivité du système de contrôle.

3. 7. Algorithme d'ordonnement

Le problème de séquençement des opérations de déstockage repose tout d'abord sur l'utilisation d'un critère d'optimisation qui sera détaillé dans le chapitre suivant. Après la

détermination des emplacements à déstocker satisfaisant la commande clients, l'ordre des opérations de déstockage se fera casier par casier selon l'algorithme 1.

Algorithme 1

```

For k = 1 to n do
  Begin
    If  $c_k <> 0$  then do
      begin
        For j=1 to  $c_k$  do begin *  $c_k$  est l'indice d'emplacement du produit à livrer le
          plus profond dans le casier k*

          <i, k,1> *choix de la couleur de franchissement*

          if  $X(k, 1) = 1$  then Livraison *livrer le produit
            correspondant*

            else convoyeur G;*Restocker le
              produit correspondant*

               $X(k,1) = X(k,1+j)$  ; *Passer à
                l'élément suivant de la matrice
                X*

          end;

        end;

      end;

    End;
  
```

3. 8. Conclusion

Dans ce chapitre nous avons montré les différents niveaux de décisions d'un système hiérarchisé. Nous nous sommes intéressés à la problématique de modélisation et d'ordonnancement des opérations de déstockage dans l'AS/RS à convoyeur gravitationnel, qui se situe au niveau opérationnel. Pour cela, nous avons développé un système de pilotage en boucle fermée utilisant un modèle RdPC comme observateur d'état et un optimiseur permettant de sélectionner les emplacements de produits à déstocker dans le rack. Deux modes de fonctionnements du système de contrôle sont traités dont le choix est conditionné par l'intensité de la demande. Nous avons ensuite proposé un algorithme d'ordonnancement exploitant la matrice de décision issue de l'optimiseur. Nous allons étudier dans le chapitre suivant le problème d'optimisation du nombre de cycles de déstockage et ainsi la génération de la matrice de décision.

Chapitre 4

Optimisation des cycles de déstockage

4. 1. Introduction

Dans ce chapitre nous traitons la méthode de génération de la matrice de décision issue du bloc optimiseur introduit dans le chapitre précédent. En effet, il s'agit du problème d'optimisation du nombre de cycles de déstockage à effectuer par la machine de déstockage vis-à-vis d'une commande client à satisfaire. Nous formulerons le problème en termes de programmation en nombres entiers. Pour cela, nous commençons tout d'abord, par une présentation de la programmation entière et ses applications pour la modélisation des problèmes de décision en recherche opérationnelle. Nous mettons ensuite l'accent sur la méthode de résolution par séparation et évaluation successives, qui est très utilisée dans la littérature pour résoudre les programmes en nombres entiers. Enfin, des exemples de cas concrets seront traités afin de montrer l'efficacité de la méthode proposée. Les résultats présentés dans ce chapitre ont fait l'objet d'une communication dans une conférence (Hachemi et Alla, 2007b). L'article a été sélectionné et accepté après, pour une publication au *Journal of Control Engineering and Applied Informatics* (Ceai, 2008).

4. 2. Positon du problème d'optimisation

Le système de stockage/déstockage à convoyeur gravitationnel est composé d'un rack et de deux machines, l'une pour le stockage et l'autre pour le déstockage. Ce rack est un ensemble de casiers identiques. Un casier contient des emplacements avec des types de produits dedans. Les casiers sont numérotés de 1 à m et les emplacements de 1 à q , comme indiqué sur la figure 3.3.

L'objectif à optimiser est : minimiser le nombre de cycles de déstockage tout en satisfaisant une demande de clients. Il est clair que, les variables représentant le nombre de cycles de déstockage dans chaque casier doivent être toutes entières. Par conséquent, nous

nous sommes intéressé à modéliser ce problème d'optimisation en utilisant la programmation en nombres entiers.

4. 3. Programmation en nombres entiers

La programmation linéaire en nombres entiers (PLNE) est une extension de la programmation linéaire qui englobe un grand nombre d'applications pour lesquels des variables sont contraintes à n'avoir que des valeurs entières. En effet, dans de nombreuses situations pratiques seules des valeurs entières sont admissibles pour les variables. Par exemple, dans un problème de planification manufacturière, le nombre d'unités produites doit être un nombre entier.

Une large classe de problèmes pratiques peut être modélisée en utilisant des variables en nombres entiers et des contraintes linéaires. Parfois un tel modèle consiste seulement en variables en nombres entiers. C'est un modèle pur de programmation en nombres entiers (*Pure Integer Programming*). S'il y'a en plus des variables continues, c'est un problème mixte (*Mixed Integer Programming*), (Avriel et Golany, 1996).

4. 3. 1. Formulation mathématique d'un programme en nombres entiers

Un problème de programmation linéaire en nombres entiers est un problème de programmation linéaire standard avec des contraintes d'intégrité, c'est-à-dire qu'on impose aux variables d'être entières. Autrement dit, si n désigne le nombre de variables, c'est un problème de la forme :

$$\begin{array}{l} \text{Maximiser } c.x \\ \text{Avec les contraintes} \end{array} \quad \left\{ \begin{array}{l} Ax \leq b \\ x \geq 0 \\ x \in \mathbb{N}^n \end{array} \right.$$

Alors qu'il existe des algorithmes polynomiaux pour résoudre les problèmes de programmation linéaire en variables réelles (l'algorithme du simplexe), le problème de programmation linéaire en variables entières est considéré comme NP-complet. Les

problèmes en variables entières sont plus difficiles à résoudre que les problèmes linéaires en variables réelles, on peut citer les raisons suivantes :

1. On ne connaît pas d'algorithme polynomial pour résoudre ce problème.
2. Avoir prouvé qu'un problème est NP-complet a des conséquences sur la façon dont on cherche à le résoudre.
 - Lorsque l'on doit traiter des problèmes de grande dimension, on peut s'attendre à ce que les algorithmes exacts ne réussissent pas à donner la solution optimale, faute de temps. On applique alors des méthodes appelées heuristiques, qui sont censées donner, en un temps raisonnable, une approximation de la solution (sans que l'on puisse parfois dire beaucoup de la façon dont elles approchent l'optimum).
 - Pour les algorithmes exacts, on fera appel à des algorithmes comme la programmation dynamique ou comme les méthodes par séparation et évaluation, appelées aussi méthodes arborescentes ou encore branch and bound.

La solution optimale des programmes entiers n'est pas toujours l'entier le plus proche de la solution optimale si celle-ci est fractionnaire. L'exemple suivant montre cela :

Exemple :

$$\begin{aligned} & \text{maximiser } 10x_1 + 11x_2 \\ \text{sous la contrainte } & 10x_1 + 12x_2 \leq 59 \\ & x_1, x_2 \geq 0 \text{ à valeurs entières} \end{aligned}$$

La solution optimale x est donnée par $x(x_1, x_2) = (5.9, 0)$, dans le cas d'un programme linéaire (sans contraintes d'intégrité). En effet, $x_1 \leq 5.9 - 1.2x_2$, et on obtient la solution optimal en posant $x_1 = 5.9, x_2 = 0$. Mais la solution $x' = (6, 0)$ n'est pas réalisable et la solution $x'' = (5, 0)$ n'est pas optimale parmi les solutions à valeurs entières.

En revanche la solution optimale à valeurs entières est donnée par $\tilde{x} = (1, 4)$.

4. 3. 2. Résolution graphique d'un programme en nombres entiers

Sur la figure 4.1 représentant un exemple général (pas l'exemple précédent), les points indiquent les couples de valeurs (x_1, x_2) dont les deux composantes sont entières. Les points encadrés représentent les valeurs admissibles compte tenu des contraintes $C1$ et $C2$. Le point M représente l'optimum obtenu en programmation linéaire en nombres réels. Le point N est l'optimum en nombres entiers. On s'aperçoit ici que l'optimum en nombres entiers n'est pas « proche » de l'optimum en nombres réels.

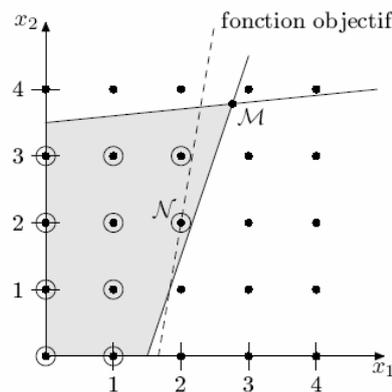


Figure 4.1. Représentation graphique d'un problème en nombre entier

4. 3. 3. Programme en nombres entiers à variables binaires

Dans de nombreuses situations pratiques on peut être amené à prendre une décision de type « tout ou rien ». Par exemple choisir parmi un ensemble de projets qui ne peuvent pas être fractionnés. Le problème de programmation fera alors intervenir des variables binaires $x_j \in \{0, 1\}$. La modélisation par des variables binaires peut être illustrée avec la décision de construire ou pas une usine ou un entrepôt sur un site j . Les variables x_j correspondent à la décision. Quand $x_j = 1$, la décision est de construire dans le site j . Quand $x_j = 0$, la décision est de ne pas construire dans le site j .

Exemple : Choix de projets

Une compagnie a établi une liste de projets d'expansion, donnée dans le tableau de la figure 4.2. Chaque projet est caractérisé par une valeur actuelle nette ainsi que par une immobilisation de capital sur chacune des 5 prochaines années. La disponibilité totale en

capital est évaluée pour chacune des 5 années suivant les montants indiqués en dernière ligne du tableau de la figure 4.2.

Projet	Immobilisation du capital par année					Valeur actuelle nette
	1	2	3	4	5	
1	10	5	20	10	0	40
2	30	20	10	10	10	70
3	10	20	27	20	10	80
4	20	10	40	20	20	100
Disponibilité en capital	50	45	70	40	30	

Figure 4.2. Données d'investissement de projets

Le problème du choix optimal de projets se pose alors ainsi :

$$\text{maximiser } 40x_1 + 70x_2 + 80x_3 + 100x_4$$

sous les contraintes

$$10x_1 + 30x_2 + 10x_3 + 20x_4 \leq 50$$

$$5x_1 + 20x_2 + 20x_3 + 10x_4 \leq 45$$

$$20x_1 + 10x_2 + 27x_3 + 40x_4 \leq 70$$

$$10x_1 + 10x_2 + 20x_3 + 20x_4 \leq 40$$

$$10x_2 + 10x_3 + 20x_4 \leq 30$$

$$x_j \in \{0, 1\}, j = 1, \dots, 4.$$

Si $x_j = 1$ le projet numéro j est sélectionné, si $x_j = 0$ il ne l'est pas. La solution, en nombres binaires, à ce problème est donnée par :

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0.$$

4. 3. 4. Méthode de résolution par séparations et évaluations successives

Dans cette section nous présentons la méthode de séparations et d'évaluations successives (*Branch and bound, en anglais*). Ce type de méthodes s'applique à la résolution de problèmes d'optimisation combinatoire avec un grand nombre de solutions envisageables et, en particulier, à la résolution de problèmes de programmation linéaire en nombres entiers où l'on peut définir la notion de solution réalisable, c'est-à-dire de solution satisfaisant les contraintes. Essentiellement, il s'agit de diviser pour conquérir, c'est-à-dire, séparer l'ensemble de toutes les solutions réalisables en sous ensembles plus petits (régions réalisables plus restreintes) et mutuellement exclusives. C'est la phase de *séparation* (ou "branching"). Ce processus de séparation élimine de l'ensemble des solutions réalisables du programme linéaire (c'est-à-dire le programme sans contraintes d'intégrité), certaines parties qui ne contiennent pas de solutions entières réalisables. On sépare ainsi le problème initial en sous problèmes mutuellement exclusifs. Ensuite, divers critères sont utilisés pour :

- Identifier les sous ensembles qui sont les plus vraisemblables de contenir la solution optimale.
- Identifier les sous ensembles qui ne doivent pas être explorés plus à fond car ils ne peuvent éventuellement pas contenir la solution optimale.

4. 3. 4. 1. Etapes de la méthode de résolution par séparation et évaluation successives

Considérons un programme linéaire en nombres entiers (PLNE). La relaxation linéaire du PLNE consiste à supprimer les contraintes d'intégrité. On obtient ainsi, un programme linéaire (PL) sous sa forme standard. Nous résumons ci-après les étapes de la méthode de résolution par séparations et évaluations successives d'un PLNE. Nous donnons ici la démarche pour un problème de maximisation, (Baillargeon, 1996).

1. Résolution du PL avec un algorithme du simplex

- Résoudre le PL correspondant (sans contraintes de variables entières). La valeur de l'objectif Z du tableau optimal donne une borne supérieure au PLNE. Appelons cette valeur Z_{BS} .
- Arrondir cette solution aux valeurs entières les plus proches (et réalisables) et calculer à nouveau la valeur de la fonction objectif. Cette valeur donne une borne inférieure pour la fonction objectif du PLNE. Appelons cette valeur Z_{BI} . Ainsi :

$$Z_{BI} \leq Z_{OPT} \text{ du PLNE} \leq Z_{BS}$$

2. Séparation

- Choisir une variable non entière pour générer deux sous problèmes. Soit x_k cette variable devant être entière dont la valeur optimale x_k^* est fractionnaire. Représentons par $[x_k^*]$ la partie entière de x_k^* .
- Pour éliminer la solution non entière x_k , on crée deux branches et deux sous problèmes. On obtient une branche avec $x_k \leq [x_k^*]$ et l'autre avec $x_k \geq [x_k^*] + 1$. Les deux sous problèmes sont obtenus en ajoutant au programme linéaire (PL) précédent (celui d'où vient la séparation), la contrainte $x_k \leq [x_k^*]$ pour un des sous problèmes et la contrainte $x_k \geq [x_k^*] + 1$ pour l'autre sous problème. Toutes les solutions entières réalisables sont maintenant contenues dans l'un ou l'autre des sous problèmes.

3. Résolution des sous problèmes et détermination de nouvelles bornes pour Z

- Il s'agit de résoudre chaque sous problème (en ne tenant pas compte des contraintes de nombres entiers). Puisque ces deux sous problèmes comportent toutes les solutions au problème à résoudre, nous pouvons également en déduire de nouvelles bornes pour Z . Notons par $MBSD$, la meilleure borne supérieure disponible. Elle correspond à la valeur maximale de la fonction objectif des deux sous problèmes que nous venons de résoudre.
- Nous révisons également la borne inférieure de Z . Notons cette valeur par $MBID$, la meilleure borne inférieure disponible. Elle correspond à la valeur maximale de la fonction objectif de toutes les solutions entières obtenues jusqu'à présent (pour un PLNE). Si dans la révision de $MBSD$, la valeur maximale Z^* qu'on obtient pour la

fonction objectif est moindre que MBID, alors la solution correspondante (celle qui donne Z^*) est moins favorable que celle qu'on obtient avec MBID. Cette solution est écartée et on pose $MBSD = MBID$.

4. Critère d'arrêt de l'exploration d'une branche

- L'exploration d'une branche est terminée lorsque l'une des conditions suivantes est satisfaite :

1. Le sous problème de la branche considérée n'admet pas de solution réalisable.
2. La solution entière obtenue pour le sous problème de la branche est réalisable mais la valeur Z^* obtenue est plus petite ou égale à MBID ($Z^* \leq MBID$). Ainsi, une branche est terminée aussitôt qu'on obtient une solution entière.
3. Le sous problème de la branche considérée admet une solution optimale non entière mais sa valeur Z^* est moindre (ou égale) que celle d'un autre sous problème à solution entière.

5. Critère d'optimalité

- Toutes les branches ont été explorées. Le processus de séparation est terminé.

La solution optimale au PLNE est la solution entière qui donne la meilleure borne inférieure pour Z (MBID). A l'optimum, on aura $MBID = MBSD = Z_{OPT}$ au PLNE.

6. Poursuite du processus de séparation (retour à 2)

- Si la solution optimale qu'on obtient à l'un ou l'autre des sous problèmes de chaque branche contient au moins une variable (devant être entière) ayant une valeur fractionnaire, on poursuit l'étape de séparation à partir du sous problème ayant la valeur la plus élevée pour Z^* (c'est-à-dire à partir du sous problème dont on a obtenu MBSD).

4.3.4. 2. Illustration de la méthode sur un exemple

Supposons que l'on veut résoudre le PLNE suivant :

$$\max Z = 2x_1 + x_2$$

sous les contraintes

$$x_1 + 2x_2 \leq 12$$

$$2x_1 - x_2 \leq 8$$

$$4x_1 - 3x_2 \geq 4$$

$$x_1 \geq 0, x_2 \geq 0$$

$$x_1, x_2 \text{ entiers.}$$

En résolvant le programme linéaire correspondant par la méthode du simplexe, on obtient la solution au PL suivante :

$$x_1^* = 5.6, x_2^* = 3.2, Z^* = 14.4. \text{ (Solution } S_0)$$

La solution arrondie est : $x_1^* = 5, x_2^* = 3, Z^* = 10 + 3 = 13$.

On peut écrire que : $Z_{BS} = 14.4$

$$Z_{BI} = 13$$

$$13 \leq Z_{OPT} \text{ du PLNE} \leq 14.4$$

Puisque la solution S_0 au PL n'est pas entière, il faut appliquer le processus de séparation. Choisissons la variable x_1 pour séparer le programme linéaire initial en deux sous problèmes. Puisque $x_1^* = 5.6$, alors $[x_1^*] = 5$. Nous formons donc les deux sous problèmes à l'aide des deux contraintes suivantes :

$$x_1 \leq 5 \text{ pour le sous problème 1} \quad (\text{SP-1})$$

$$x_1 \geq 6 \text{ pour le sous problème 2} \quad (\text{SP-2})$$

Nous avons alors les deux sous problèmes suivants à résoudre (en ignorant les contraintes de solution entière) :

<p>SP-1</p> $\max Z = 2x_1 + x_2$ <p>sous les contraintes</p> $x_1 + 2x_2 \leq 12$ $2x_1 - x_2 \leq 8$ $4x_1 - 3x_2 \geq 4$ $x_1 \leq 5$ $x_1 \geq 0, x_2 \geq 0$

<p>SP-2</p> $\max Z = 2x_1 + x_2$ <p>sous les contraintes</p> $x_1 + 2x_2 \leq 12$ $2x_1 - x_2 \leq 8$ $4x_1 - 3x_2 \geq 4$ $x_1 \geq 6$ $x_1 \geq 0, x_2 \geq 0$

La résolution des deux sous problème donne :

SP-1

$$x_1^* = 5, x_2^* = 3.5, Z^* = 13.5$$

SP-2

Pas de solution réalisable.

Le sous problème SP-2 est terminal par suite de la condition 1 du critère d'arrêt (solution non réalisable) ; il est inutile d'explorer plus loin cette branche.

Nous poursuivons le processus de séparation à partir de SP-1. Nous allons le séparer à l'aide de la valeur fractionnaire de x_2 . Puisque $x_2^* = 3.5$, alors $[x_2^*] = 3$. Nous devons former deux autres sous problèmes à l'aide des contraintes suivantes :

$$x_2 \leq 3 \text{ pour le sous problème 3} \quad (\text{pour SP-3})$$

$$x_2 \geq 4 \text{ pour le sous problème 4} \quad (\text{pour SP-4})$$

Nous ajoutons donc la contrainte $x_2 \leq 3$ au sous problème SP-1 pour obtenir le sous problème SP-3. On fait de même pour la contrainte $x_2 \geq 4$ pour obtenir le sous problème SP-4.

Nous éliminons alors de SP-1, toutes les valeurs de x_2 comprises dans l'intervalle $3 < x_2 < 4$.

SP-3

$$\max Z = 2x_1 + x_2$$

sous les contraintes

$$x_1 + 2x_2 \leq 12$$

$$2x_1 - x_2 \leq 8$$

$$4x_1 - 3x_2 \geq 4$$

$$x_1 \leq 5$$

$$x_2 \leq 3$$

$$x_1 \geq 0, x_2 \geq 0$$

SP-4

$$\max Z = 2x_1 + x_2$$

sous les contraintes

$$x_1 + 2x_2 \leq 12$$

$$2x_1 - x_2 \leq 8$$

$$4x_1 - 3x_2 \geq 4$$

$$x_1 \leq 5$$

$$x_2 \geq 4$$

$$x_1 \geq 0, x_2 \geq 0$$

La résolution de ces deux sous problèmes donne :

SP-3

$$x_1^* = 5, x_2^* = 3, Z^* = 13$$

SP-4

$$x_1^* = 4, x_2^* = 4, Z^* = 12$$

La figure 4.3 donne l'arbre d'exploration. Nous constatons que la branche du sous problème SP-3 est terminale selon la condition d'arrêt 2 : $Z^* = 13 = \text{MBID}$.

Toutes les branches ont été explorées. La solution optimale au PLNE est la solution entière réalisable qui donne MBID soit :

$$x_1 = 5, x_2 = 3, Z_{\max} = 13.$$

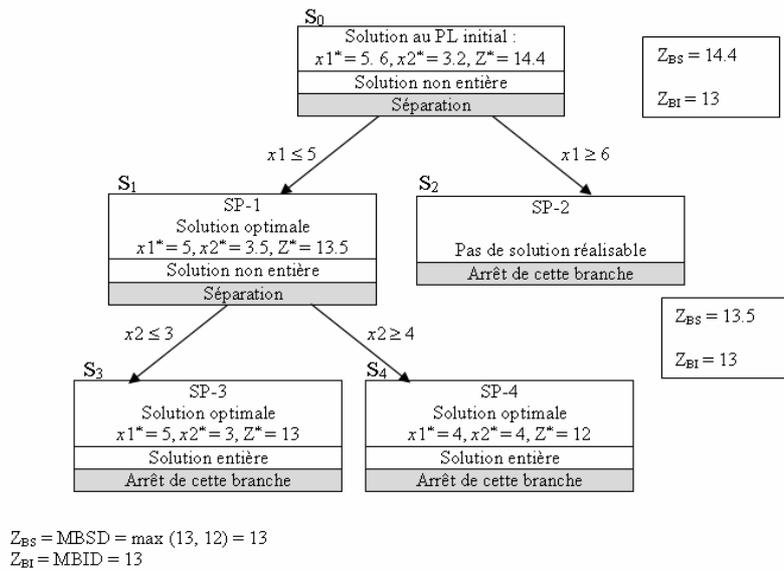


Figure 4.3. Arbre d'exploration des solutions

4. 4. Formulation du problème d'optimisation des cycles de déstockage

Le problème consiste à satisfaire la demande de clients (traitée par lot) en réalisant le minimum de cycles de déstockage. En effet, une opération de restockage d'un produit est très coûteuse car elle oblige à le renvoyer pour être restocké de nouveau dans le rack. Ceci étant, la différence entre les temps d'accès aux différents casiers du rack n'est pas pénalisante en termes de nombre de cycles de restockage et par conséquent, on considère toute la pénalité sur la profondeur du rack. Notre objectif est de minimiser le nombre de cycles.

Nous considérons de plus une politique de stockage aléatoire, c'est-à-dire que les produits peuvent être stockés dans n'importe quel emplacement du rack. Ce choix permet de montrer l'efficacité de la méthode d'optimisation qui reste cependant valable pour toute autre politique de stockage.

La solution consiste à rechercher les emplacements dans les casiers, contenant les produits demandés, et permettant d'employer le minimum de cycles de déstockage. Il est évident qu'un minimum de cycles de déstockage induit forcément un minimum de cycles de restockage. De plus, la considération du nombre de cycles de déstockage diffère selon que les produits à déstocker se trouvent dans le même casier ou dans des casiers différents.

Pour illustrer le problème de déstockage, considérons l'exemple de la figure 4.4 représentant sur un plan x-y, un rack à six casiers de quatre emplacements chacun. Le rack contient trois types de produits symbolisés par ●, ■ et ▲. Considérons une demande de clients composée de quatre produits de type ● et de deux produits de type ▲. Une solution (trajectoire) consiste à déterminer une combinaison d'emplacements qui satisfait la demande (Chapitre 3). Deux solutions S_1 et S_2 particulières sont indiquées dans cette figure (S_1 indiqué en trait plein et S_2 en pointillés).

Pour calculer le coût global de déstockage pour une trajectoire, on calcule d'abord le coût par produit et ensuite on fait la somme des coûts pour tous les produits. Le coût du produit est égal à sa profondeur dans le casier. Dans le cas de présence de plusieurs produits de la trajectoire dans le même casier, le coût de tous ces produits est égal à celui du produit le plus profond. Dans la suite on ne parlera plus que du coût du casier donné par le produit le plus profond. Ainsi, la trajectoire S_1 a un coût global de : $1 + 2 + 3 + 3 + 4 + 4 = 17$ alors que celui de S_2 est de : $3 + 2 + 3 + 3 + 2 = 13$ (seuls 5 termes apparaissent).

A noter que, pour la première solution, le nombre de produits à restocker est égal à $17 - 6 = 11$ produits alors que seulement $13 - 6 = 7$ produits sont à restocker pour la deuxième solution.

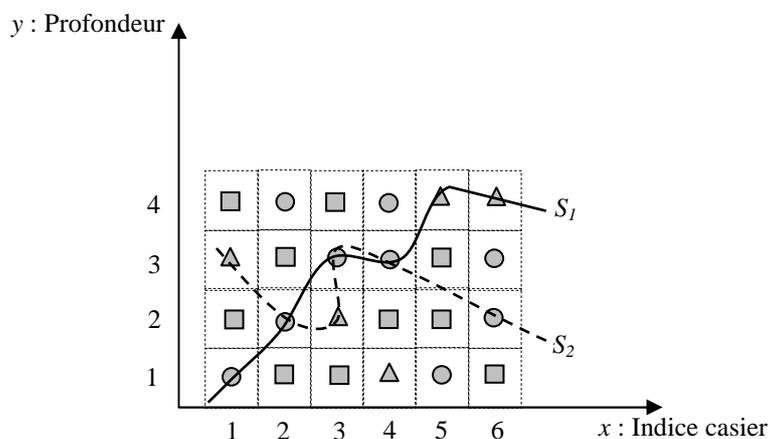


Figure 4.4. Coût de déstockage en fonction des emplacements choisis

D'une façon générale, les données sont :

- m casiers ayant chacun q emplacements.
- Les produits de types différents contenus dans les emplacements.
- Les quantités demandées de chaque type de produit.

Problème : Déterminer une trajectoire satisfaisant une demande (contenant les produits demandés) de coût global minimal.

Critère: Minimiser la somme des coûts des casiers. Ceci est formalisé par les relations 1 à 6, qui seront présentées dans la suite.

Le rack est présenté comme une matrice où les numéros de lignes indiquent les indices de casiers, les numéros de colonnes indiquent les indices des emplacements et où les valeurs correspondent aux types des produits. La demande de clients (lot de la demande) est représentée par un vecteur contenant les quantités demandées par type de produit. Les paramètres utilisés sont :

k : indice (numéro) de casier, $k \in [1, m]$.

j : indice (numéro) de l'emplacement, $j \in [1, q]$.

i : type de produit, $i \in [1, n]$.

S_{kj} : matrice d'état du rack, cette matrice est déduite du marquage Z du modèle RdPC, ce dernier est réactualisé suivant l'évolution du système.

$S_{kj} = 0$; veut dire qu'aucun produit n'est contenu dans cet emplacement.

$$Z = \begin{bmatrix} m(V) \\ m(W) \\ m(R) \\ m(D) \end{bmatrix}$$

$$S : F \times E \rightarrow P$$

$$(fk, ej) \mapsto S(fk, ej) = pi$$

$$\forall \langle pi, fk, ej \rangle \in m(W) : S(fk, ej) = pi$$

$$S(fk, ej) = \phi \text{ ailleurs}$$

En utilisant juste les indices :

$$S_{kj} = \begin{cases} i, & \text{si } \langle pi, fk, ej \rangle \in m(W) \\ 0 & \text{sinon} \end{cases}$$

Q : le vecteur du lot de demande des clients de dimension n

M_{kj} : matrice binaire définie comme suit :

$M_{kj} \in \{0,1\}$, cette matrice donne le coût de chaque casier.

X_{kj} : matrice de décision binaire définie comme suit :

$X_{kj} \in \{0,1\}$ tel que :

$$X_{kj} = \begin{cases} 1 & \text{si l'emplacement } j \text{ du casier } k \text{ est sélectionné pour un déstockage} \\ 0 & \text{sinon} \end{cases}$$

Le problème mathématique peut être représenté comme un problème d'optimisation en nombres entiers. Ainsi l'objectif peut être formulé comme suit :

$$\text{Objectif : } \min \sum_{k=1}^m \sum_{j=1}^q M_{kj} * j \quad [1]$$

Sous

$$\sum_{j=1}^q M_{kj} \leq 1 \quad \forall k \in [1, m] \quad [2]$$

$$M_{kj} = 0 \text{ si } S_{kj} = 0 \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [3]$$

$$\sum_{k=1}^m \sum_{j=1}^q X_{kj} = Q_i \text{ si } S_{kj} = i \quad \forall i \in [1, n] \quad [4]$$

$$X_{kj} = 0 \text{ si } S_{kj} = 0 \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [5]$$

$$X_{kj} \leq \sum_{l=j}^q M_{kl} \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [6]$$

L'objectif [1] consiste à minimiser les cycles de déstockage. Il minimise la profondeur totale des produits dans le rack. La contrainte [2] permet de prendre en compte le fait que, si plusieurs produits à déstocker sont dans le même casier, la profondeur exprimant le nombre de cycles de déstockage à effectuer dans ce casier est égale au maximum des indices d'emplacements. La contrainte [3] permet d'éliminer du choix les emplacements vides. La contrainte [4] assure une satisfaction complète de la demande de clients, en fonction de la disponibilité des produits dans le rack. Similairement à la contrainte [3], la contrainte [5] assure que les emplacements vides seront éliminés dans la matrice de décision X. La contrainte [6] réajuste les éléments de la matrice M, afin de permettre le calcul du nombre de cycles de déstockage de l'objectif. En effet, la valeur pour laquelle M_{kj} est non nul est exactement la valeur de X_{kj} non nul le plus profond. La solution de ce modèle donne les

emplacements des produits à déstocker et le nombre minimum de cycles de déstockage de la machine de déstockage MD.

4. 5. Etude de cas

4. 5. 1. Exemple 1

Considérons un rack avec 6 casiers de 7 emplacements chacun. Supposons que 10 types de produits sont disponibles dans le rack. Ainsi :

$m = 6$; le nombre de casiers du rack,

$q = 7$; le nombre d'emplacements de chaque casier,

$n = 10$; le nombre de types de produits dans le rack.

Etant donné que nous nous intéressons à l'optimisation du nombre de cycles de déstockage et par conséquent du nombre de cycles de restockage et pour des raisons de clarté, nous présenterons le rack avec des casiers disposés verticalement. C'est une représentation abstraite équivalente à la représentation physique.

La matrice S représente l'état du rack, avec $S_{kj} = i$ où k est l'indice de casier, j l'indice de l'emplacement du casier et i est le type de produit s'y trouvant. La figure 4.5 représente le rack en fonction de la matrice S considérée dans cet exemple, les nombres à l'intérieur des emplacements sont les types de produits.

D'où :

$$S = \begin{bmatrix} 1 & 2 & 9 & 8 & 4 & 6 & 2 \\ 8 & 10 & 4 & 3 & 4 & 6 & 10 \\ 8 & 6 & 10 & 3 & 10 & 10 & 7 \\ 1 & 4 & 10 & 7 & 7 & 9 & 10 \\ 10 & 2 & 3 & 4 & 10 & 7 & 3 \\ 3 & 1 & 5 & 3 & 7 & 8 & 8 \end{bmatrix}$$

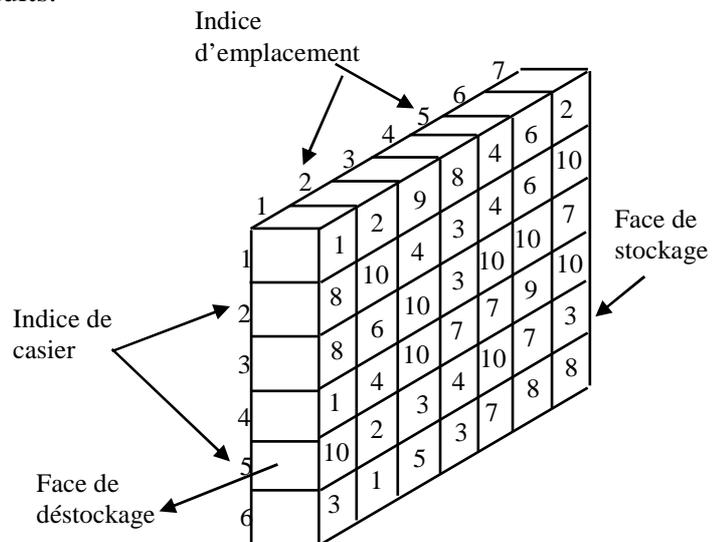


Figure 4.5. Etat du rack avec des types de produits dedans

La demande des clients (lot) est donnée comme suit :

$$Q = [3 \ 3 \ 0 \ 5 \ 0 \ 0 \ 0 \ 0 \ 5]^T$$

Les éléments de ce vecteur représentent les quantités demandées pour chaque type de produit. Cette demande consiste en 3 produits de type 1, 3 de type 2, 5 de type 4 et 5 de type 10.

Pour programmer ce problème d'optimisation, nous avons utilisé le langage AMPL, qui est un langage très puissant pour modéliser des problèmes linéaires, non linéaires et en nombres entiers. Ce modèle est ensuite résolu par CPLEX ; un solveur pour les programmes linéaires, entiers et réseaux.

Le modèle est exécuté sur un PC avec une vitesse de processeur de 1,73 GHz et 256 Mo de mémoire vive (RAM).

Pour évaluer les performances de la méthode d'optimisation, nous avons exécuté le programme sur la plateforme logicielle d'optimisation qui est disponible dans (Neos, 2007). Cette plateforme fournit un listing des différentes opérations exécutées, dont on peut résumer les principaux résultats, pour l'exemple du rack traité ici, comme suit :

- Le nombre de variables, toutes binaires, est égal à 84.
- Le nombre de contraintes, toutes linéaires, est de 142.
- Le temps de résolution est égal à 0.06 secondes, avec une solution optimale trouvée.

Après la résolution du programme en utilisant le système AMPL/CPLEX, nous obtenons les résultats suivants :

Objectif = 24; est le nombre de cycles de déstockage à effectuer par la machine de déstockage.

La matrice de décision X sélectionnant les emplacements des produits à livrer est égale à :

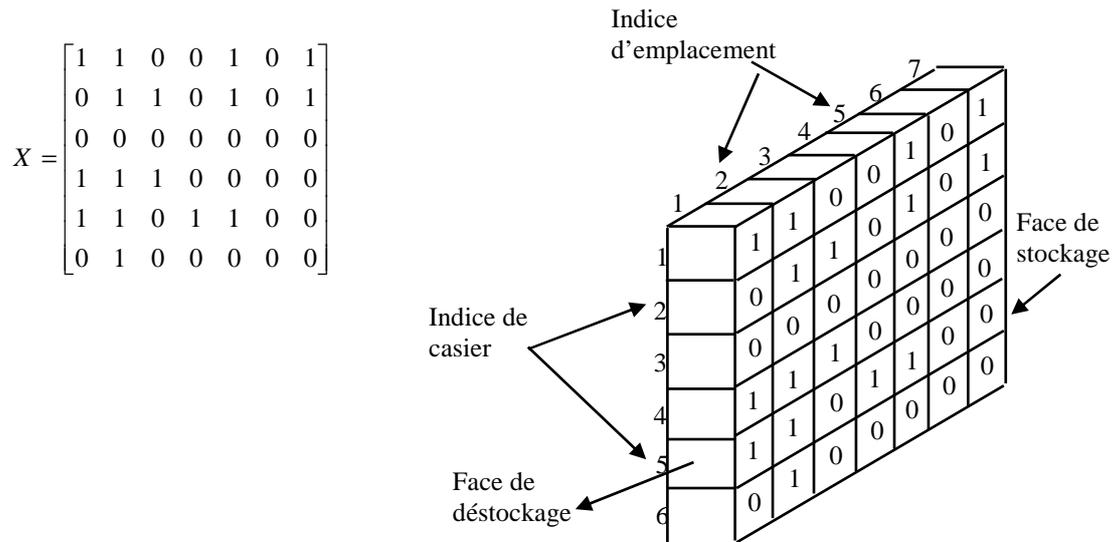


Figure 4.6. Emplacements des produits à déstocker

La matrice X , où les lignes représentent les indices de casiers et les colonnes les indices d'emplacements, est une matrice binaire avec des éléments pouvant prendre soit la valeur 0 soit 1 :

0: Le produit contenu dans l'emplacement et le casier correspondant n'est pas pris.

1: Le produit contenu dans l'emplacement et le casier correspondant est pris.

La figure 4.6 montre les emplacements des produits à déstocker (contenant la valeur 1) en fonction de la demande de clients. Ainsi, la commande de la machine est traitée casier par casier selon le tableau de la figure 4.7.

Casier (f_k)	Emplacements des produits à déstocker (e_j)	Nombre de cycles de déstockage (incluant les produits délivrés et restockés)
$f1$	$e1, e2, e5, e7$	7
$f2$	$e2, e3, e5, e7$	7
$f3$	<i>Aucun</i>	0
$f4$	$e1, e2, e3$	3
$f5$	$e1, e2, e4, e5$	5
$f6$	$e2$	2

Figure 4.7. Emplacements des produits à déstocker par casier

Notons qu'étant donné que la machine de déstockage atteint uniquement le premier emplacement de chaque casier, sa commande consiste à faire un nombre donné de cycles de déstockage. Le nombre de cycles de déstockage c_k (incluant les cycles de livraison et les cycles vers restockage), pour chaque casier est égal à :

$$c_k = \sum_{j=1}^q M_{kj} * j$$

Le tableau de la figure 4.8 donne les emplacements des produits à déstocker par type. Dans cet exemple, on remarque que le nombre total de cycles à effectuer par la machine de déstockage est égal à 24 cycles et la demande de clients est composée de 16 produits, cela veut dire que le nombre de produits à restocker est égal à : $24 - 16 = 8$ produits, à envoyer au convoyeur de restockage.

Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{16}{24} = 0.67$$

Cela implique que 67% des opérations de déstockage sont pour la livraison. Les opérations restantes représentent des opérations d'envoi vers le convoyeur de restockage. La solution du problème est optimale, ce taux est le meilleur taux qu'il est possible d'atteindre pour la demande de clients considérée.

Type de produit	Quantités demandées	Emplacements des produits à déstocker (casier, emplacement)
1	3	$(f1, e1), (f4, e1), (f6, e2)$
2	3	$(f1, e2), (f1, e7), (f5, e2)$
4	5	$(f1, e5), (f2, e3), (f2, e5),$ $(f4, e2), (f5, e4)$
10	5	$(f2, e2), (f2, e7), (f4, e3),$ $(f5, e1), (f5, e5)$

Figure 4.8. Emplacements des produits à déstocker par type

4. 5. 2. Exemple 2

Considérons un rack avec 10 casiers de 10 emplacements chacun, ce qui donne une capacité de stockage de 100 produits. Le nombre de types de produits disponibles dans le rack est de 10. Ainsi :

Casier (f_k)	Emplacements des produits à déstocker (e_j)	Nombre de cycles de déstockage (incluant les produits délivrés et restockés)
$f1$	<i>Aucun</i>	0
$f2$	$e2, e3$	3
$f3$	$e3, e4, e5, e6$	6
$f4$	$e1$	1
$f5$	$e1$	1
$f6$	$e1$	1
$f7$	<i>Aucun</i>	0
$f8$	$e1$	1
$f9$	$e1, e5, e6, e7, e9$	9
$f10$	$e1$	1

Figure 4.9. Emplacements des produits à déstocker par casier

Le tableau de la figure 4.9 indique les opérations de la machine de déstockage sur les différents casiers du rack.

Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{16}{23} = 0.7$$

Cela implique que 70% des opérations de déstockage sont pour la livraison. Par conséquent, dans ce cas, 30% des cycles de la machine de déstockage sont destinés vers le convoyeur de restockage.

Pour le modèle résolu de cet exemple on a :

- Le nombre de variables, toutes binaires, est égal à 200.
- Le nombre de contraintes, toutes linéaires, est de 320.
- Le temps de résolution est égal à 0.11 secondes, avec une solution optimale trouvée.

4. 5. 3. Exemple 3

Considérons un rack avec 25 casiers de 12 emplacements chacun, ce qui donne une capacité de stockage de 300 produits. Le nombre de types de produits disponibles dans le rack est de 10. Ainsi :

$m = 25$; le nombre de casiers du rack,

$q = 12$; le nombre d'emplacements de chaque casier,

$n = 10$; le nombre de types de produits dans le rack.

Le rack est composé des produits selon la matrice S.

$$S = \begin{bmatrix} 3 & 10 & 8 & 8 & 2 & 3 & 6 & 6 & 7 & 4 & 7 & 3 \\ 10 & 3 & 9 & 7 & 2 & 4 & 3 & 5 & 6 & 5 & 2 & 4 \\ 5 & 3 & 1 & 9 & 10 & 7 & 5 & 5 & 2 & 8 & 8 & 10 \\ 1 & 1 & 5 & 6 & 9 & 3 & 10 & 3 & 7 & 5 & 6 & 7 \\ 6 & 7 & 3 & 5 & 7 & 10 & 8 & 1 & 4 & 2 & 4 & 5 \\ 9 & 5 & 2 & 5 & 5 & 7 & 3 & 3 & 9 & 5 & 8 & 1 \\ 9 & 6 & 6 & 5 & 8 & 9 & 4 & 5 & 3 & 3 & 9 & 10 \\ 8 & 10 & 3 & 4 & 4 & 6 & 10 & 1 & 1 & 6 & 5 & 4 \\ 2 & 1 & 1 & 5 & 10 & 8 & 2 & 5 & 3 & 2 & 9 & 7 \\ 3 & 2 & 7 & 9 & 4 & 3 & 9 & 5 & 7 & 4 & 4 & 8 \\ 7 & 2 & 5 & 8 & 1 & 6 & 9 & 2 & 3 & 8 & 4 & 4 \\ 1 & 5 & 4 & 3 & 6 & 7 & 6 & 6 & 8 & 9 & 5 & 7 \\ 4 & 9 & 9 & 7 & 9 & 5 & 2 & 3 & 3 & 10 & 8 & 6 \\ 9 & 10 & 7 & 8 & 3 & 10 & 2 & 4 & 6 & 1 & 8 & 1 \\ 2 & 3 & 4 & 4 & 9 & 6 & 9 & 2 & 8 & 9 & 10 & 4 \\ 1 & 5 & 6 & 5 & 5 & 4 & 9 & 4 & 5 & 4 & 2 & 3 \\ 2 & 7 & 9 & 1 & 2 & 5 & 2 & 2 & 5 & 4 & 7 & 9 \\ 10 & 4 & 3 & 4 & 1 & 3 & 5 & 4 & 6 & 6 & 1 & 1 \\ 4 & 5 & 4 & 5 & 10 & 9 & 3 & 2 & 5 & 2 & 10 & 6 \\ 6 & 7 & 5 & 9 & 6 & 9 & 1 & 1 & 10 & 10 & 3 & 7 \\ 1 & 3 & 4 & 6 & 9 & 4 & 1 & 2 & 3 & 8 & 7 & 4 \\ 1 & 5 & 6 & 9 & 5 & 2 & 9 & 2 & 5 & 4 & 9 & 1 \\ 9 & 8 & 4 & 5 & 6 & 7 & 9 & 4 & 3 & 2 & 5 & 2 \\ 10 & 4 & 9 & 10 & 3 & 5 & 4 & 1 & 3 & 5 & 4 & 6 \\ 4 & 9 & 9 & 10 & 4 & 5 & 7 & 3 & 1 & 2 & 9 & 3 \end{bmatrix}$$

La demande des clients (lot) est donnée comme suit :

$$Q = [6 \ 5 \ 0 \ 4 \ 0 \ 5 \ 0 \ 0 \ 7 \ 3]^T$$

La résolution du programme en utilisant le système AMPL/CPLEX, donne les résultats suivants :

Objectif = 32; est le nombre de cycles de déstockage à effectuer par la machine de déstockage.

La matrice de décision X sélectionnant les emplacements des produits à livrer est égale à :

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{30}{32} = 0.94$$

Cela implique que 94% des opérations de déstockage sont pour la livraison. En effet, comme on peut le voir sur la matrice X il n'y a que deux produits à renvoyer vers le restockage, c'est le produit contenu dans l'emplacement 1 du casier 10 et le produit contenu dans l'emplacement 2 du casier 15 (éléments (10, 1) et (15, 2) de la matrice X).

Pour le modèle résolu de cet exemple on a :

- Le nombre de variables, toutes binaires, est égal à 600.
- Le nombre de contraintes, toutes linéaires, est de 930.
- Le temps de résolution est égal à 0.28 secondes, avec une solution optimale trouvée.

4. 6. Conclusion

Ce chapitre a été consacré au problème d'optimisation du nombre de cycles de déstockage à effectuer par la machine de déstockage tout en satisfaisant la demande des clients. Un nombre minimum de cycles de déstockage engendre un nombre minimum de cycles de restockage. Les cycles de restockage ne participent pas à la satisfaction de la demande et en plus ils sont coûteux en temps, d'où tout l'intérêt de les réduire au maximum. Ceci étant, le problème d'optimisation est formulé en termes de programme en nombres entiers. Les données de ce problème proviennent de la matrice d'état du rack, fourni par le modèle RdPC déjà exposé dans le chapitre précédent, et du vecteur de la demande de clients. La solution de ce problème donne les emplacements optimaux de produits à déstocker pour satisfaire la demande de clients, réalisant le minimum de cycles de déstockage de la machine de déstockage. Pour les exemples traités, les temps de calcul de la solution optimale étaient de quelques dixièmes de secondes. L'avantage de cette méthode d'optimisation est qu'elle donne le nombre minimum de cycles de la machine de déstockage permettant de satisfaire le lot de la demande des clients. Ainsi, le taux de livraison se trouve maximisé et le taux de restockage minimisé, ce qui permet à la machine de déstockage de fonctionner plus efficacement avec un meilleur débit du système.

Chapitre 5

Intégration des dates de péremption des produits dans le pilotage

5. 1. Introduction

Dans le chapitre précédent, nous avons présenté la méthode d'optimisation qui est à la base de la génération de la matrice de décision. Dans ce problème d'optimisation le produit stocké n'était caractérisé que par son type et son emplacement dans le rack. Dans ce chapitre, nous allons ajouter une nouvelle information concernant le produit, il s'agit de sa date de péremption. En effet, cela devient primordial quand on a affaire à des produits périssables qui ont une durée de vie limitée. Nous allons tout d'abord monter le contexte et les motivations qui nous ont poussés à introduire les dates de péremption des produits dans le pilotage du système de stockage. Ensuite, nous étendons le modèle RdPC du système déjà développé pour contenir également les dates de péremption. La prise en compte des dates de péremption engendre un changement dans le modèle de décision. Ainsi nous développerons une nouvelle méthode d'optimisation qui minimise le nombre de cycles de déstockage mais en même temps réduit le risque de péremption dans le rack. Pour ce faire, une formulation mathématique du nouveau problème est présentée. Enfin, nous traitons des exemples de cas concrets et nous exposons les performances des solutions trouvées.

5. 2. Contexte et enjeux

Avec des dépenses de 29 milliards d'euros en 2002, les médicaments représentent 40 % environ de la consommation de soins et de biens médicaux, pour les hôpitaux français, (Balsan et al, 2004). L'analyse du flux physique de médicaments dans un hôpital permet de distinguer trois grandes étapes :

- livraison des médicaments commandés à la pharmacie, jusqu'au stockage,
- dispensation des médicaments dans les services, avec préparation et transport,

- administration des médicaments aux patients (et retour des médicaments non administrés).

Le schéma de la figure 5.1 fournit un synoptique du circuit physique du médicament :

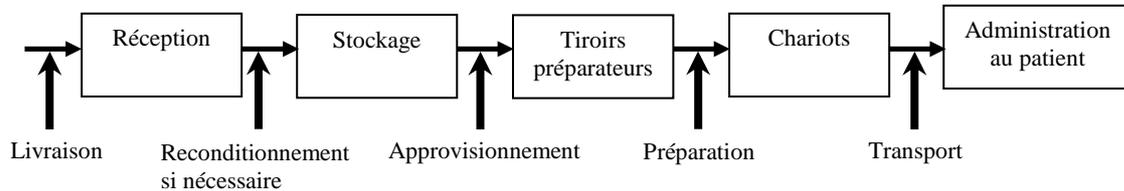


Figure 5.1. Le circuit du médicament

Le cadre d'étude dans ce chapitre se rapporte plus particulièrement à l'installation de systèmes automatisés de stockage dans des pharmacies d'hôpitaux. Le problème de l'optimisation du stockage des médicaments et des dispositifs médicaux stériles répond à des enjeux importants pour les hôpitaux ;

- gain de temps ou de la productivité par l'efficacité dans l'activité du traitement et de la manutention des produits stockés

- économie de surface au sol par la réduction des surfaces de stockage

- réduction des pertes de médicaments par une optimisation du contrôle de l'inventaire.

On voit se développer de plus en plus l'installation des systèmes automatisés et robotisés dans la distribution des médicaments dans les structures de santé. En France, les dispositifs de distribution automatisés sont graduellement installés sur quelques sites pilotes comme Aix-en-Provence (Woronoff et al, 2003) ou l'hôpital de Toulouse qui a installé le premier système robotisé de la distribution individuelle de médicaments « Homerus » (Hospimedia, 2002). Au Québec, l'intérêt pour les nouvelles technologies et les possibilités qu'elles offrent en ce qui concerne la distribution des médicaments, a fait l'objet de plusieurs recommandations dans un récent rapport sur les systèmes automatisés et robotisés utilisés pour la distribution des médicaments dans les établissements de santé (Dgpsegi, 2005). Mais la plupart des systèmes de contrôle ne peuvent pas tenir en compte d'un aspect majeur du problème pharmaceutique qui est l'amélioration de la qualité et de la diminution des risques par une meilleure gestion des péremptions des médicaments. La réduction des péremptions passe par la gestion des

dates de péremption de chaque lot, avec l'intégration de cette information dans le système de contrôle pendant les opérations d'entrées/sorties en stock ou pendant le retour des médicaments non administrés.

Vu la périssabilité des produits, garder les produits périmés induit un coût supplémentaire. Des pertes importantes peuvent survenir si le stock n'est pas efficacement géré. La surveillance et le contrôle des produits périssables, tels que les produits pharmaceutiques peuvent être facilité par l'utilisation d'un système automatisé de stockage/déstockage qui permet une collection de données en temps réel et un interfaçage efficace avec le système de gestion dans la chaîne logistique. Un autre avantage consiste dans le fait que pour superviser l'inventaire, le système à besoin d'identifier les produits une seule fois (juste avant leur introduction dans le rack) sans déplacement de produits, ni scanning ou intervention humaine à chaque fois.

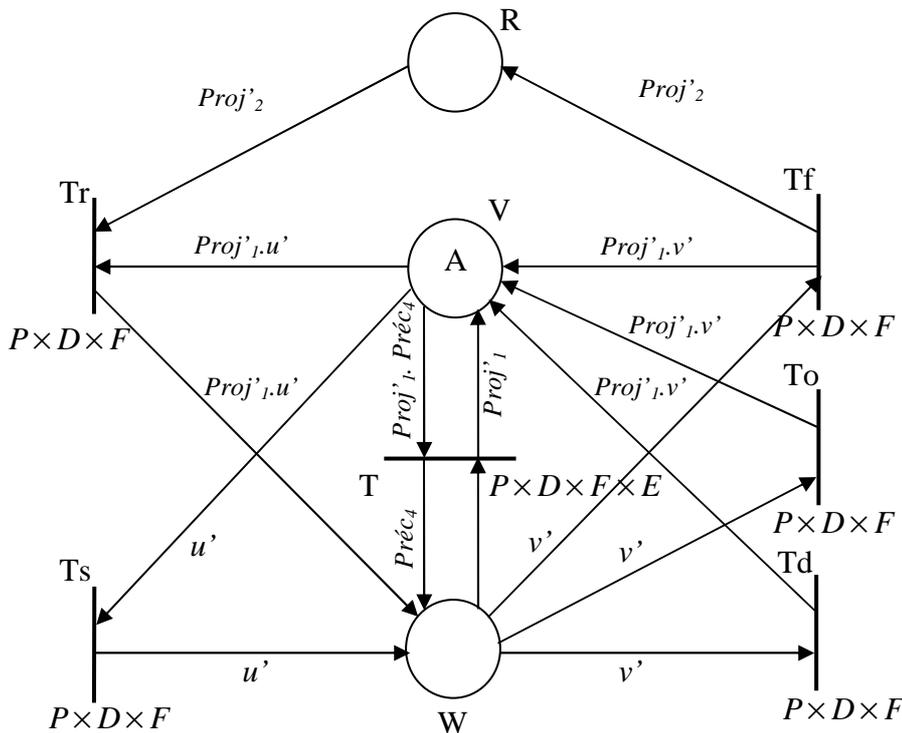
Nous avons constaté que les études traitant l'ordonancement (séquencement) des opérations dans les AS/RS ont considéré uniquement le type du produit stocké sans sa date de péremption. De ce fait, nous nous proposons de prendre en compte ce facteur (date de péremption) dans le modèle de contrôle. En effet, les produits périssables comme les produits pharmaceutiques exigent plus de contraintes dans la gestion des stocks, spécialement celles relatives aux dates de péremptions des produits. Chaque article a une date de péremption, c'est-à-dire qu'il a une durée de vie connue après quoi il n'est plus utile et doit être enlevé du rack. D'ailleurs, il est nécessaire de choisir les produits à déstocker, qui ont les dates de péremption les plus proches afin de réduire le risque de péremption dans le rack.

L'objectif de cette étude est d'intégrer les dates de péremption des produits dans le contrôle optimal du système automatisé de stockage/déstockage à convoyeur gravitationnel. Nous traitons le problème d'ordonancement (séquencement) des opérations de déstockage en introduisant en plus les dates de péremptions des produits. A notre connaissance ce problème n'a pas été traité dans la littérature relative au séquencement du déstockage dans les AS/RS. Le rôle principal du système de contrôle et d'ordonancement est de satisfaire la demande des clients avec les produits dont la date de péremption est la plus proche, en réalisant le minimum de cycles de déstockage. Ainsi, nous réduisons le risque d'expiration des produits.

5. 3. Intégration des dates de péremption des produits dans le modèle RdPC de l'AS/RS à convoyeur gravitationnel

Le modèle RdPC de l'AS/RS présenté dans le chapitre 3, modélisait la dynamique du système par rapport aux mouvements des produits à l'intérieur du rack. Dans ce modèle, les produits sont caractérisés par seulement leur type sans aucune information sur leur date d'expiration. Le modèle est valable pour les produits non périssable, tel que les pièces mécaniques. En revanche, pour des produits périssables tel que les médicaments et les produits de soins, la prise en compte des péremptions devient capitale et impérative, pour les raisons que nous avons cité plus haut. Dans cette section, nous nous proposons d'étendre ce modèle en intégrant en plus du type de produit, sa date de péremption. Par conséquent, le modèle doit représenter aussi l'évolution dans le temps, des dates de péremptions et si nécessaire déterminer les produits périmés pour les évacuer du rack.

La figure 5.2 montre le nouveau modèle RdPC incluant les dates de péremption comme information supplémentaire des produits.



$$A = M_0(V) = \sum_{k=1}^m \sum_{j=1}^q \langle f_k, e_j \rangle$$

Figure 5.2. Le modèle RdPC de l'AS/RS avec les dates de péremption

Similairement au modèle du chapitre 3, le modèle de la figure 5.2 représente le rack comme des files FIFO. Pour modéliser les dates de péremption des produits, nous avons ajouté un autre ensemble de couleur D , représentant les dates de péremptions selon des classes, pour une raison de simplicité. En effet, plus l'indice de classe est petit plus le produit a une date de péremption proche. Les ensembles de couleurs régissant ce modèle sont :

$$P = \{ \langle pi \rangle, i \in [1, n] \} : \text{ensemble des types de produits.}$$

$$D = \{ \langle dr \rangle, r \in [1, l] \} : \text{ensemble des dates de péremptions (par classe).}$$

$$F = \{ \langle fk \rangle, k \in [1, m] \} : \text{ensemble des casiers du rack.}$$

$$E = \{ \langle ej \rangle, j \in [1, q] \} : \text{ensemble des emplacements d'un casier.}$$

$$\text{Le marquage initial } A = M_0(V) = \sum_{k=1}^m \sum_{j=1}^q \langle fk, ej \rangle$$

Les arcs sont associés à des fonctions et des fonctions composées. Elles sont définies comme suit :

La fonction u' introduit un produit pi de date de péremption dr dans le dernier emplacement d'un casier fk :

$$\begin{aligned} u' : P \times D \times F &\rightarrow P \times D \times F \times \{eq\} \\ (pi, dr, fk) &\mapsto u(pi, dr, fk) = \langle pi, dr, fk, eq \rangle \end{aligned}$$

La fonction v' extrait un produit pi de date de péremption dr du premier emplacement d'un casier fk :

$$\begin{aligned} v' : P \times D \times F &\rightarrow P \times D \times F \times \{e1\} \\ (pi, dr, fk) &\mapsto v(pi, dr, fk) = \langle pi, dr, fk, e1 \rangle \end{aligned}$$

Dans ce qui suit, nous introduisons une nouvelle fonction $Proj'_i$ tel que $Proj'_i = Proj_i Proj_i$.

Dans le RdPC de la figure 5.2 la transition T_s est validée par rapport à la couleur $\langle pi, dr, fk \rangle$ car la place V contient la couleur $Proj'_1$. $u'(\langle pi, dr, fk \rangle) = Proj'_1(pi, dr, fk, eq) = \langle fk, eq \rangle$, l'emplacement q du casier est vide. Le franchissement de T_s dépose un jeton de couleur $u'(\langle pi, dr, fk \rangle) = \langle pi, dr, fk, eq \rangle$ dans la place W et retire un jeton $\langle fk, eq \rangle$ de la place V , ainsi l'emplacement q du casier fk contient un produit de type i de date de péremption dr . Ce produit sera transféré d'un emplacement à un autre par le franchissement de la transition T . Supposons que la place W contienne un jeton $\langle pi, dr, fk, ej \rangle$, la transition T sera franchie

selon cette couleur si la place V contient un jeton de couleur $Proj'_1$. $Préc4 (< pi, dr, fk, ej >)$ $= < fk, ej-1 >$, l'emplacement $(j-1)$ doit être vide. Le franchissement de T dépose un jeton $Proj'_1(<pi, dr, fk, ej >) = < fk, ej >$ dans la place V (l'emplacement j est libéré) et un jeton $Préc4 (< pi, dr, fk, ej >) = < pi, fk, ej-1 >$ dans la place W (l'emplacement $j-1$ est occupé). Ce transfert va continuer tant qu'il y a de la place dans le casier jusqu'à ce que le produit arrive à l'emplacement 1.

La transition Tr décrit un fonctionnement similaire à celui de la transition Ts mais pour le restockage d'un produit. Un produit peut être évacué d'un casier soit vers la station de livraison, soit vers le restockage (recyclage). Le franchissement de Tf dépose un jeton $Proj'_2 (pi, dr, fk) = pi$ dans la place R et un jeton $Proj'_1. v' (<pi, dr, fk>) = Proj'_1 (pi, dr, fk, e1) = <fk, e1>$ dans la place V . Il retire un jeton $v' (<pi, dr, fk>) = < pi, dr, fk, e1 >$ de la place W . Le franchissement de Td évacue un produit vers la station de livraison. Ce franchissement dépose un jeton $Proj'_2 (pi, dr, fk) = pi$ dans la place D et un jeton $Proj'_1. v' (<pi, dr, fk>) = Proj'_1(pi, dr, fk, e1) = <fk, e1>$ dans la place V . En même temps il retire un jeton $v' (<pi, dr, fk>) = < pi, dr, fk, e1 >$ de la place W .

La transition To représente l'évacuation des produits périmés en dehors du rack. Cette transition doit être temporisée/synchronisée pour permettre de déterminer les produits dont la date de péremption est atteinte, afin de les extraire du rack.

L'interprétation des places et des transitions du RdPC est résumée dans le tableau de la figure 5.3.

Places/Transition	Couleurs	Interprétation
V	$\langle fk, ej \rangle$	L'emplacement ej du casier fk est vide
W	$\langle pi, dr, fk, ej \rangle$	L'emplacement ej du casier fk contient un produit de type pi de date de péremption dr
R	$\langle pi \rangle$	Un produit de type pi est recyclé
Ts	$\langle pi, dr, fk \rangle$	Introduction d'un produit pi de date de péremption dr dans l'emplacement q du casier fk
Tr	$\langle pi, dr, fk \rangle$	Introduction d'un produit recyclé pi de date de péremption dr dans l'emplacement q du casier fk
Td	$\langle pi, dr, fk \rangle$	Evacuation d'un produit de type pi de date de péremption dr de l'emplacement 1 du casier fk vers la station de livraison
To	$\langle pi, dr, fk \rangle$	Evacuation d'un produit périmé pi de date de péremption dr en dehors du rack
Tf	$\langle pi, dr, fk \rangle$	Evacuation d'un produit de type pi de date de péremption dr de l'emplacement 1 du casier fk vers le convoyeur de restockage
T	$\langle pi, dr, fk, ej \rangle$	Déplacement d'un produit pi de date de péremption dr de l'emplacement j à l'emplacement $j-1$

Figure 5.3. Interprétation des places et transitions du modèle RdPC

5. 4. Problème d'optimisation intégrant les dates de péremption

5. 4. 1. Problème

Le nouveau problème d'optimisation consiste à faire un compromis entre la profondeur des produits dans le rack (définissant ainsi le nombre de cycles de déstockage) et le choix des dates de péremptions de ces produits. La solution que nous recherchons est de trouver une combinaison d'emplacements de produits à déstocker satisfaisant la demande, contenant le minimum de cycles de déstockage tout en tenant en compte les contraintes de péremption des produits. En effet, en plus du problème d'optimisation du nombre de cycles de déstockage, il s'agit aussi de minimiser le risque de péremption. Minimiser le risque de péremption passe par le choix de produits dont la date de péremption est la plus proche.

Notons que la structure de pilotage en boucle fermé reste la même, seuls les modèles RdPC de l'observateur et le bloc d'optimisation changent.

5. 4. 2. Formulation mathématique du problème d'optimisation

Le rack est présenté comme une matrice où les numéros de lignes indiquent les indices de casiers, les numéros de colonnes indiquent les indices des emplacements et où les valeurs correspondent aux types des produits. Une autre matrice représente les dates de péremption des produits dans le rack. La demande de clients (lot de la demande) est représentée par un vecteur contenant les quantités demandées par type de produit. L'objectif consiste à :

Objectif : Satisfaire la demande des clients avec les produits qui ont la date de péremption la plus proche et en même temps réalisant le minimum de cycles de déstockage.

Les paramètres utilisés sont :

k : indice (numéro) de casier, $k \in [1, m]$.

j : indice (numéro) de l'emplacement, $j \in [1, q]$.

i : type de produit, $i \in [1, n]$.

r : date de péremption du produit, $r \in [1, l]$

S_{kj} : matrice d'état du rack, avec $k = 1 \dots m$, $j = 1 \dots q$. Les éléments de cette matrice représentent les types de produits stockés dans leurs respectifs emplacements.

T_{kj} : matrice des dates de péremptions, $k = 1 \dots m$, $j = 1 \dots q$. Les éléments de cette matrice représentent les dates de péremptions des produits stockés dans leurs respectifs emplacements.

En fait ces matrices peuvent être déduites du marquage G du modèle RdPC. Ce marquage est actualisé en fonction de l'évolution du système.

$$G = \begin{bmatrix} m(V) \\ m(W) \\ m(R) \\ m(D) \\ m(O) \end{bmatrix}$$

$$S : F \times E \rightarrow P$$

$$(fk, ej) \mapsto S(fk, ej) = pi$$

$$\forall \langle pi, fk, ej \rangle \in m(W) : S(fk, ej) = pi$$

$$S(fk, ej) = \emptyset \text{ ailleurs}$$

$$\forall \prec pi, dr, fk, ej \succ \in m(W): S(fk, ej) = pi$$

$$S(fk, ej) = \phi \text{ ailleurs}$$

En utilisant juste les indices :

$$S_{kj} = \begin{cases} i, & \text{si } \prec pi, dr, fk, ej \succ \in m(W) \\ 0 & \text{sinon} \end{cases}$$

$S_{kj} = 0$; veut dire qu'aucun produit n'est contenu dans cet emplacement.

Similairement, la matrice T_{kj} est déduite comme suit :

$$T_{kj} = \begin{cases} r, & \text{si } \prec pi, dr, fk, ej \succ \in m(W) \\ 0 & \text{sinon} \end{cases}$$

$T_{kj} = 0$; cela veut dire qu'aucun produit n'est contenu dans cet emplacement.

Q : le vecteur du lot de demande des clients de dimension n

M_{kj} : matrice binaire définie comme suit :

$M_{kj} \in \{0,1\}$, cette matrice donne le coût de chaque casier.

X_{kj} : matrice de décision binaire définie comme suit :

$X_{kj} \in \{0,1\}$ tel que :

$$X_{kj} = \begin{cases} 1 & \text{si l'emplacement } j \text{ du casier } k \text{ est sélectionné pour un déstockage} \\ 0 & \text{sinon} \end{cases}$$

L'approche que nous utilisons consiste à résoudre le problème d'optimisation en deux phases. Dans la première phase, il est nécessaire de sélectionner tous les produits candidats pour le déstockage et ceci en déterminant une limite supérieure de la date de péremption pour chaque type de produit demandé. En d'autres termes, il s'agit de déterminer pour chaque type de produit demandé, l'intervalle le plus court des dates de péremption, dans lequel se trouve un nombre suffisant de produits, permettant la satisfaction de la quantité demandé.

Pour la simplicité, nous considérons les dates de péremptions selon des classes, pour lesquelles un produit avec la classe de date de péremption r possède une date d'expiration plus proche que celle d'un produit de classe $r + 1$.

Pour illustrer le choix de la limite supérieur de la date de péremption, prenons comme exemple, une demande de sept produits d'un type donné de produit. Les quantités disponibles dans le rack en fonction des classes de péremption sont données dans le tableau de la figure 5.4. Alors l'intervalle le plus court des classes de péremption suffisant pour satisfaire la quantité demandée (7 produits) est [1 3], car la quantité totale résultante des trois premières classes est égale dix ce qui est suffisant. La classe quatre n'est pas utile dans ce cas.

Classe de péremption pour un type de produit	Quantité dans le rack
1	2
2	3
3	5
4	3

Figure 5.4. Quantités en fonction des classes de péremption

La seconde phase consiste à sélectionner les emplacements des produits à déstocker, parmi les produits dont la classe de péremption appartient à l'intervalle de classes de péremption du produit concerné. Ceci en réalisant le minimum de cycles de déstockage. Par conséquent à chaque phase il y a un objectif.

Pour la première phase, nous définissons la matrice C_{ir} , avec $i=1,\dots,n, r=1,\dots,l$. Les éléments de cette matrice représentent le nombre d'un produit donné i avec la classe de péremption r . Cette matrice est calculée comme suit :

$$C_{ir} = \sum_{k=1}^m \sum_{j=1}^q Z_{kj} \quad \forall i \in [1, n], \forall r \in [1, l]$$

$$\text{Tel que : } Z_{kj} = \begin{cases} 1, & \text{if } (S_{kj} = i) \text{ and } (T_{kj} = r) \\ 0 & \text{sinon} \end{cases}$$

H_{ri} : est une matrice binaire définie comme suit :

$$H_{ri} \in \{0,1\}, \quad \forall r \in [1, l], \forall i \in [1, n]$$

H'_{ri} : est une matrice binaire de décision définie comme suit :

$H'_{ri} \in \{0,1\}$ tel que :

$$H'_{ri} = \begin{cases} 1 & \text{si la classe de péremption } r \text{ du type de produit } i \text{ est sélectionné pour le déstockage} \\ 0 & \text{sinon} \end{cases}$$

H''_{ri} : est une matrice binaire de décision définie comme suit :

$$H''_{ri} \in \{0,1\}$$

Phase 1:

$$\text{Objectif} : \min \sum_{i=1}^n \sum_{r=1}^l C_{ir} * H_{ri} + r * H''_{ri} \quad [1]$$

Sous

$$\sum_{r=1}^l C_{ir} * H_{ri} \geq Q_i \quad \forall i \in [1, n] \quad [2]$$

$$H_{ri} \geq H_{(r+1)i} \quad \forall r \in [1, l-1] \text{ et } \forall i \in [1, n] \quad [3]$$

$$H'_{ri} = H_{ri} \quad \text{si } C_{ir} > 0 \quad \forall r \in [1, l] \text{ et } \forall i \in [1, n] \quad [4]$$

$$H'_{ri} = 0 \quad \text{si } C_{ir} = 0 \quad \forall r \in [1, l] \text{ et } \forall i \in [1, n] \quad [5]$$

$$H'_{ri} \leq \sum_{k=r}^l H''_{ki} \quad \forall r \in [1, l] \text{ et } \forall i \in [1, n] \quad [6]$$

L'objectif [1] minimise simultanément, le nombre total des produits choisis en fonction de leurs classes de péremptions, ainsi que la somme des plus grandes classes de péremption utilisées pour chaque type de produit demandé. Il donne par le biais de la matrice H' les classes de péremption concernées par le déstockage pour chaque type de produit, c'est à dire que les produits qui seront déstockés seront parmi ceux dont la classe de péremption est sélectionnée dans cette matrice.

Phase 2:

$$\text{Objectif} : \min \sum_{k=1}^m \sum_{j=1}^q M_{kj} * j \quad [7]$$

Sous

$$\sum_{j=1}^q M_{kj} \leq 1 \quad \forall k \in [1, m] \quad [8]$$

$$M_{kj} = 0 \quad \text{si } S_{kj} = 0 \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [9]$$

$$\sum_{k=1}^m \sum_{j=1}^q X_{kj} = C_{ir} \quad \text{si } (S_{kj} = i) \text{ et } (T_{kj} = r) \text{ et } ((H' - H'')_{ri} = 1) \quad \forall i \in [1, n] \text{ et } \forall r \in [1, l] \quad [10]$$

$$\sum_{k=1}^m \sum_{j=1}^q X_{kj} = 0 \quad \text{si } (S_{kj} = i) \text{ et } (H'_{di} = 0) \quad \forall i \in [1, n] \quad [11]$$

$$\sum_{k=1}^m \sum_{j=1}^q X_{kj} = Q_i \quad \text{si } S_{kj} = i \quad \forall i \in [1, n] \quad [12]$$

$$X_{kj} = 0 \quad \text{si } S_{kj} = 0 \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [13]$$

$$X_{kj} \leq \sum_{l=j}^q M_{kl} \quad \forall k \in [1, m] \text{ et } \forall j \in [1, q] \quad [14]$$

L'objectif [7] consiste à minimiser le nombre des cycles de déstockage. Il minimise la profondeur totale des produits dans le rack. Les contraintes [10] et [11] permettent de réduire la recherche aux produits déjà sélectionnés dans la phase 1. En effet, le lien entre les deux phases se fait par le bais des matrices H' et H'' . La contrainte [12] assure une satisfaction complète de la demande de clients, en fonction de la disponibilité des produits dans le rack. Les autres contraintes sont identiques à celles du problème d'optimisation des cycles de déstockage du chapitre précédent.

5. 5. Etude de cas

5. 5. 1. Exemple 1

Considérons un rack avec 5 casiers de 10 emplacements chacun. Le rack contient 10 types de produits avec 5 classes de péremption. Ainsi :

$m = 5$; le nombre de casiers du rack,

$q = 10$; le nombre d'emplacements de chaque casier,

$n = 10$; le nombre de types de produits dans le rack,

$l = 5$; la classe de péremption la plus élevée.

Le rack contient des produits selon la matrice S représentant l'état du rack l'état du rack, avec $S_{kj} = i$ où k est l'indice de casier, j l'indice de l'emplacement du casier et i est le type de produit s'y trouvant. Pour cet exemple, elle est donnée comme suit :

$$S = \begin{bmatrix} 1 & 2 & 9 & 8 & 4 & 1 & 2 & 9 & 6 & 1 \\ 8 & 10 & 4 & 3 & 4 & 6 & 10 & 5 & 3 & 10 \\ 8 & 6 & 10 & 3 & 10 & 10 & 1 & 10 & 2 & 4 \\ 1 & 4 & 10 & 7 & 7 & 9 & 10 & 4 & 5 & 7 \\ 10 & 2 & 3 & 4 & 10 & 7 & 3 & 2 & 4 & 9 \end{bmatrix}$$

La matrice T (éléments T_{kj}) représentant les classes de péremption des produits stockés dans leurs respectifs emplacements est aussi donnée.

La figure 5.5 représente le rack selon les matrices S et T considérées dans cet exemple. Le premier nombre des couples à l'intérieur du rack représente le type de produit et le second représente sa classe de péremption.

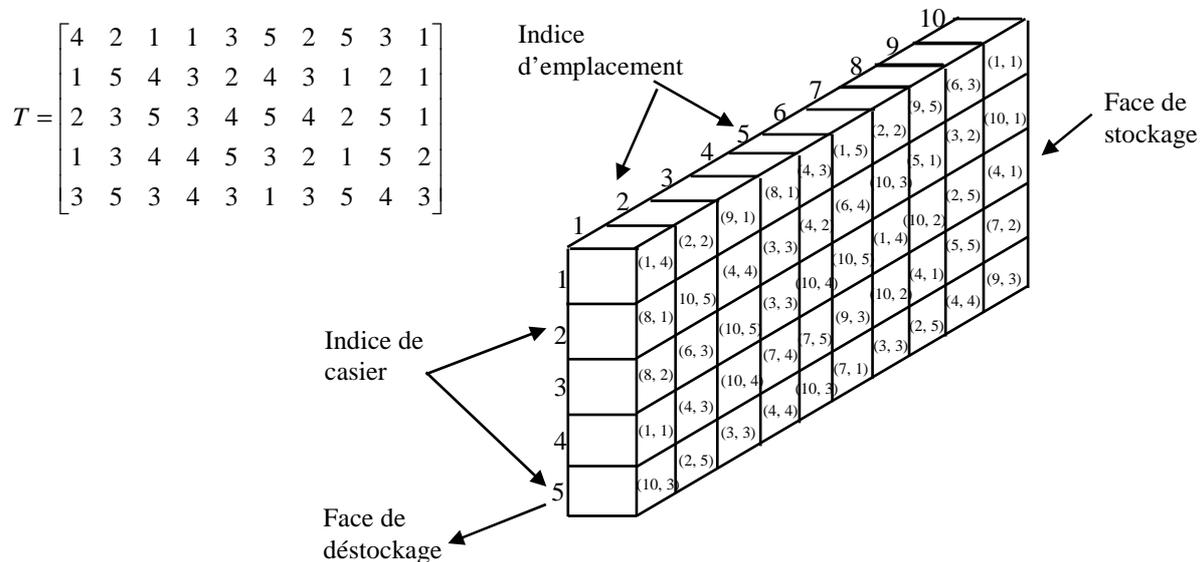


Figure 5.5. Le rack suivant les types et classes de péremption des produits

La demande des clients (lot) est donnée comme suit :

$$Q = [3 \ 3 \ 0 \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 5]^T$$

Rappelons que les éléments de ce vecteur représentent les quantités demandées pour chaque type de produit. Cette demande consiste en 3 produits de type 1, 3 de type 2, 5 de type 4 et 5 de type 10.

Pour programmer ce problème d'optimisation, nous avons utilisé le système langage AMPL. Ce modèle est ensuite résolu par le solveur CPLEX.

Pour le modèle résolu de la phase 1, de cet exemple on a :

- Le nombre de variables, toutes binaires, est égal à 240.
- Le nombre de contraintes, toutes linéaires, est de 230.
- Le temps de résolution est égal à 0.09 secondes, avec une solution optimale trouvée.

Pour le modèle résolu de la phase 2, on a :

- Le nombre de variables, toutes binaires, est égal à 100.
- Le nombre de contraintes, toutes linéaires, est de 255.
- Le temps de résolution est égal à 0.07 secondes, avec une solution optimale trouvée.

Après la résolution séquentielle des deux problèmes, correspondants aux deux phases, on obtient les résultats suivants :

Objectif = 39; est le nombre de cycles de déstockage à effectuer par la machine de déstockage.

La matrice de décision X sélectionnant les emplacements des produits à livrer est égale à :

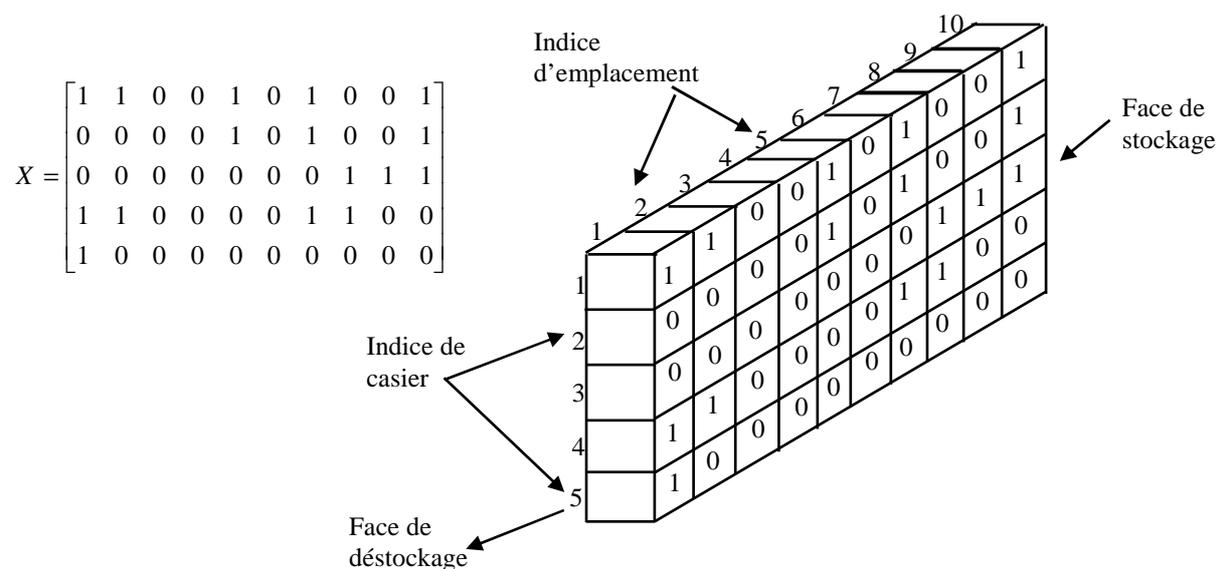


Figure 5.6. Emplacements des produits à déstocker

La figure 5.6 montre les emplacements des produits à déstocker (contenant la valeur 1) en fonction de la demande de clients. Ainsi, la commande de la machine est traitée casier par casier selon le tableau de la figure 5.7.

Casier (f_k)	Emplacements des produits à déstocker (e_j)	Nombre de cycles de déstockage (incluant les produits délivrés et restockés)
$f1$	$e1, e2, e5, e7, e10$	10
$f2$	$e5, e7, e10$	10
$f3$	$e8, e9, e10$	10
$f4$	$e1, e2, e7, e8$	8
$f5$	$e1$	1

Figure 5.7. Emplacements des produits à déstocker par casier

Notons que le nombre total de cycles à effectuer par la machine de déstockage est égal à 39 cycles. Par ailleurs, la demande de clients est composé de 16 produits, ce qui veut dire que le nombre de produits à restocker, donc à envoyer vers le convoyeur de restockage, est égal à : $39 - 16 = 23$ produits. Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{16}{39} = 0.41$$

Cela implique que 41% des opérations de déstockage sont pour la livraison. Par conséquent, dans ce cas, presque 60% des cycles de la machine de déstockage sont destinés vers le convoyeur de restockage. Notons que si nous n'avons pas tenu en compte des péremptions, le nombre optimal des cycles de déstockage serait seulement de 22 cycles. Cela permettrait une économie de 17 cycles avec un taux de la livraison meilleur et égal à 72%. Cependant, le risque de péremption des produits à l'intérieur du rack sera plus grand.

5. 5. 2. Exemple 2

Considérons un rack avec 15 casiers de 10 emplacements chacun. Le rack contient 10 types de produits avec 8 classes de péremption. Ainsi :

$m = 15$; le nombre de casiers du rack,

$q = 10$; le nombre d'emplacements de chaque casier,

$n = 10$; le nombre de types de produits dans le rack,

$l = 8$; la classe de péremption la plus élevée.

Le rack est composé des produits selon la matrice S :

$$S = \begin{bmatrix} 6 & 5 & 9 & 1 & 4 & 3 & 2 & 10 & 3 & 7 \\ 8 & 10 & 4 & 3 & 4 & 6 & 10 & 5 & 3 & 10 \\ 8 & 6 & 10 & 3 & 10 & 10 & 1 & 10 & 2 & 4 \\ 1 & 4 & 10 & 7 & 7 & 9 & 10 & 4 & 5 & 7 \\ 10 & 2 & 3 & 4 & 10 & 7 & 3 & 2 & 4 & 9 \\ 8 & 6 & 10 & 3 & 10 & 10 & 1 & 10 & 2 & 4 \\ 4 & 10 & 7 & 1 & 7 & 9 & 10 & 4 & 5 & 7 \\ 2 & 2 & 5 & 4 & 8 & 4 & 9 & 7 & 3 & 2 \\ 5 & 8 & 10 & 3 & 7 & 4 & 1 & 5 & 4 & 1 \\ 2 & 5 & 1 & 6 & 4 & 2 & 5 & 4 & 10 & 8 \\ 9 & 3 & 4 & 2 & 10 & 8 & 9 & 7 & 1 & 3 \\ 10 & 1 & 4 & 7 & 2 & 1 & 9 & 3 & 2 & 6 \\ 5 & 2 & 1 & 3 & 5 & 6 & 9 & 10 & 3 & 1 \\ 7 & 2 & 3 & 10 & 9 & 7 & 4 & 2 & 2 & 7 \\ 10 & 5 & 4 & 4 & 6 & 1 & 3 & 6 & 4 & 5 \end{bmatrix}$$

Les classes de péremption des produits sont données par la matrice T comme suit :

$$T = \begin{bmatrix} 4 & 5 & 2 & 2 & 3 & 4 & 7 & 7 & 8 & 8 \\ 3 & 2 & 1 & 5 & 4 & 6 & 2 & 4 & 3 & 1 \\ 2 & 1 & 2 & 5 & 3 & 4 & 4 & 8 & 7 & 6 \\ 1 & 3 & 7 & 4 & 5 & 3 & 2 & 1 & 6 & 2 \\ 3 & 5 & 3 & 4 & 3 & 1 & 3 & 5 & 4 & 3 \\ 5 & 1 & 3 & 2 & 7 & 6 & 6 & 1 & 8 & 4 \\ 3 & 5 & 5 & 1 & 7 & 2 & 8 & 4 & 5 & 7 \\ 5 & 2 & 2 & 3 & 1 & 4 & 1 & 6 & 7 & 8 \\ 7 & 2 & 1 & 1 & 2 & 3 & 3 & 5 & 4 & 5 \\ 1 & 4 & 2 & 6 & 3 & 1 & 4 & 2 & 6 & 6 \\ 1 & 8 & 4 & 8 & 1 & 2 & 1 & 5 & 4 & 2 \\ 8 & 7 & 1 & 1 & 4 & 5 & 6 & 4 & 7 & 5 \\ 6 & 2 & 1 & 3 & 5 & 6 & 2 & 3 & 3 & 1 \\ 1 & 2 & 6 & 8 & 4 & 6 & 2 & 7 & 7 & 8 \\ 4 & 4 & 6 & 5 & 6 & 7 & 1 & 6 & 4 & 5 \end{bmatrix}$$

La demande des clients (lot) est donnée comme suit :

$$Q = [5 \ 0 \ 7 \ 3 \ 0 \ 8 \ 0 \ 2 \ 3 \ 2]^T$$

Cette demande consiste en 5 produits de type 1, 7 de type 3, 3 de type 4, 8 de type 6, 2 de type 8, 3 de type 9 et 2 de type 10, ce qui donne une demande de 30 produits.

Pour programmer ce problème d'optimisation, nous avons utilisé le système langage AMPL. Ce modèle est ensuite résolu par le solveur CPLEX.

Pour le modèle résolu de la phase 1, de cet exemple on a :

- Le nombre de variables, toutes binaires, est égal à 240.
- Le nombre de contraintes, toutes linéaires, est de 230.
- Le temps de résolution est égal à 0.11 secondes, avec une solution optimale trouvée.

Pour le modèle résolu de la phase 2, on a :

- Le nombre de variables, toutes binaires, est égal à 300.
- Le nombre de contraintes, toutes linéaires, est de 565.
- Le temps de résolution est égal à 0.11 secondes, avec une solution optimale trouvée.

Après la résolution séquentielle des deux problèmes, correspondants aux deux phases, on obtient les résultats suivants :

Objectif = 78; est le nombre de cycles de déstockage à effectuer par la machine de déstockage.

La matrice de décision X sélectionnant les emplacements des produits à livrer est égale à :

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

La commande de la machine de déstockage est traitée casier par casier selon le tableau de la figure 5.8.

Casier (f_k)	Emplacements des produits à déstocker (e_j)	Nombre de cycles de déstockage (incluant les produits délivrés et restockés)
$f1$	$e1$	1
$f2$	$e3$	3
$f3$	$e1, e2$	2
$f4$	$e1, e8$	8
$f5$	$e3$	3
$f6$	$e2, e4$	4
$f7$	$e4$	4
$f8$	$e5, e7$	7
$f9$	$e3, e4$	4
$f10$	$e3, e4$	4
$f11$	$e1, e5, e7, e10$	10
$f12$	$e3, e10$	10
$f13$	$e3, e4, e6, e9, e10$	10
$f14$	Aucun	0
$f15$	$e5, e7, e8$	8

Figure 5.8. Emplacements des produits à déstocker par casier

Notons que le nombre total de cycles à effectuer par la machine de déstockage est égal à 78 cycles. Par ailleurs, la demande de clients est composé de 30 produits, ce qui veut dire que le nombre de produits à restocker, donc à envoyer vers le convoyeur de restockage, est égal à : $78 - 30 = 48$ produits. Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{30}{78} = 0.38$$

Cela implique que 38% des opérations de déstockage sont pour la livraison. Par conséquent, dans ce cas, 62% des cycles de la machine de déstockage sont destinés vers le convoyeur de restockage. Notons que si nous n'avons pas tenu en compte des péremptions, le nombre optimal des cycles de déstockage serait seulement de 42 cycles. Cela permettrait une économie de 36 cycles avec un taux de la livraison meilleur et égal à 71%. Cependant, le risque de péremption des produits à l'intérieur du rack sera plus grand.

Pour tenter de réduire cet écart induit par la prise en compte de la péremption, on peut penser à choisir une politique de stockage, s'arrangeant au maximum à stocker les produits selon leur classe de péremption.

5. 5. 3. Exemple 3

Afin de réduire l'écart dû à la prise en compte des péremptions des produits, on peut choisir une politique de stockage qui consiste à stocker, *dans la mesure du possible*, les produits selon leur classe de péremption. En effet, malgré ce choix, la disposition des produits selon leur péremption peut être perturbée par les opérations de restockage, dans lesquels des produits non demandés peuvent se retrouver dans le fond du rack.

Dans cet exemple, nous allons voir qu'un stockage basé sur les classes de péremption, dans lequel, on s'arrange à disposer les produits d'un type donné, en ordre graduel de péremption, permet d'améliorer le taux de livraison et ainsi réduire l'écart dû à la contrainte de péremption. Ce choix revient à placer les produits, d'un type donné, qui ont la classe de péremption la plus proche, le plus près de la face de déstockage.

Considérons un rack avec 20 casiers de 10 emplacements chacun. Le rack contient 10 types de produits avec 8 classes de péremption. Ainsi :

$m = 20$; le nombre de casiers du rack,

$q = 10$; le nombre d'emplacements de chaque casier,

$n = 10$; le nombre de types de produits dans le rack,

$l = 8$; la classe de péremption la plus élevée.

Le rack est composé des produits selon la matrice S :

$$S = \begin{bmatrix} 6 & 5 & 9 & 1 & 4 & 3 & 2 & 10 & 3 & 7 \\ 8 & 10 & 4 & 3 & 4 & 6 & 10 & 5 & 3 & 10 \\ 8 & 6 & 10 & 3 & 10 & 10 & 1 & 10 & 2 & 4 \\ 1 & 4 & 10 & 7 & 7 & 9 & 10 & 4 & 5 & 7 \\ 10 & 2 & 3 & 4 & 10 & 7 & 3 & 2 & 4 & 9 \\ 8 & 6 & 10 & 3 & 10 & 10 & 1 & 10 & 2 & 4 \\ 4 & 10 & 7 & 1 & 7 & 9 & 10 & 4 & 5 & 7 \\ 2 & 2 & 5 & 4 & 8 & 4 & 9 & 7 & 3 & 2 \\ 5 & 8 & 10 & 3 & 7 & 4 & 1 & 5 & 4 & 1 \\ 2 & 5 & 1 & 6 & 4 & 2 & 5 & 4 & 10 & 8 \\ 9 & 3 & 4 & 2 & 10 & 8 & 9 & 7 & 1 & 3 \\ 10 & 1 & 4 & 7 & 2 & 1 & 9 & 3 & 2 & 6 \\ 5 & 2 & 1 & 3 & 5 & 6 & 9 & 10 & 3 & 1 \\ 7 & 2 & 3 & 10 & 9 & 7 & 4 & 2 & 2 & 7 \\ 10 & 5 & 4 & 4 & 6 & 1 & 3 & 6 & 4 & 5 \\ 4 & 8 & 9 & 1 & 5 & 7 & 2 & 3 & 10 & 4 \\ 3 & 7 & 6 & 3 & 2 & 10 & 8 & 1 & 2 & 9 \\ 8 & 1 & 5 & 7 & 3 & 3 & 9 & 2 & 2 & 10 \\ 2 & 7 & 3 & 10 & 5 & 2 & 1 & 3 & 5 & 6 \\ 9 & 3 & 4 & 2 & 10 & 10 & 6 & 9 & 2 & 4 \end{bmatrix}$$

Les classes de péremption des produits sont données par la matrice T comme suit :

$$T = \begin{bmatrix} 4 & 2 & 2 & 1 & 3 & 4 & 7 & 7 & 8 & 5 \\ 3 & 2 & 1 & 5 & 4 & 6 & 2 & 4 & 3 & 1 \\ 2 & 1 & 2 & 5 & 3 & 4 & 3 & 8 & 7 & 6 \\ 1 & 3 & 1 & 1 & 3 & 3 & 2 & 1 & 6 & 5 \\ 3 & 5 & 3 & 4 & 3 & 4 & 3 & 5 & 4 & 3 \\ 5 & 1 & 3 & 2 & 7 & 6 & 8 & 1 & 8 & 4 \\ 3 & 5 & 1 & 1 & 3 & 2 & 8 & 4 & 4 & 5 \\ 5 & 2 & 2 & 3 & 1 & 4 & 1 & 5 & 7 & 8 \\ 2 & 2 & 1 & 1 & 3 & 3 & 8 & 4 & 4 & 5 \\ 1 & 1 & 1 & 6 & 3 & 1 & 2 & 2 & 6 & 6 \\ 1 & 8 & 4 & 8 & 1 & 2 & 1 & 5 & 8 & 2 \\ 8 & 1 & 1 & 1 & 4 & 3 & 6 & 4 & 7 & 5 \\ 1 & 2 & 3 & 3 & 4 & 6 & 2 & 3 & 3 & 8 \\ 1 & 2 & 6 & 8 & 4 & 3 & 2 & 7 & 7 & 5 \\ 4 & 1 & 6 & 5 & 6 & 3 & 1 & 6 & 4 & 4 \\ 1 & 8 & 4 & 3 & 2 & 2 & 1 & 5 & 4 & 2 \\ 8 & 1 & 1 & 1 & 4 & 5 & 6 & 8 & 7 & 5 \\ 6 & 3 & 1 & 3 & 5 & 6 & 2 & 3 & 3 & 1 \\ 1 & 1 & 6 & 8 & 2 & 6 & 8 & 7 & 4 & 8 \\ 4 & 4 & 6 & 5 & 6 & 7 & 1 & 6 & 4 & 5 \end{bmatrix}$$

La demande des clients (lot) est donnée comme suit :

$$Q = [5 \ 0 \ 0 \ 0 \ 5 \ 0 \ 10 \ 0 \ 0 \ 0]^T$$

Cette demande consiste en 5 produits de type 1, 5 de type 5 et 10 de type 7, ce qui donne une demande de 20 produits.

Pour programmer ce problème d'optimisation, nous avons utilisé le système langage AMPL. Ce modèle est ensuite résolu par le solveur CPLEX.

Pour le modèle résolu de la phase 1, de cet exemple on a :

- Le nombre de variables, toutes binaires, est égal à 240.
- Le nombre de contraintes, toutes linéaires, est de 230.
- Le temps de résolution est égal à 0.07 secondes, avec une solution optimale trouvée.

Pour le modèle résolu de la phase 2, on a :

- Le nombre de variables, toutes binaires, est égal à 400.
- Le nombre de contraintes, toutes linéaires, est de 720.

- Le temps de résolution est égal à 0.07 secondes, avec une solution optimale trouvée.

Après la résolution séquentielle des deux problèmes, correspondants aux deux phases, on obtient les résultats suivants :

Objectif = 39; est le nombre de cycles de déstockage à effectuer par la machine de déstockage.

La matrice de décision X sélectionnant les emplacements des produits à livrer est égale à :

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Notons que le nombre total de cycles à effectuer par la machine de déstockage est égal à 39 cycles. Par ailleurs, la demande de clients est composé de 20 produits, ce qui veut dire que le nombre de produits à restocker, donc à envoyer vers le convoyeur de restockage, est égal à : $39 - 20 = 19$ produits. Le taux de livraison sera donc égal à :

$$\tau_d = \frac{\text{nombre de produits délivrés}}{\text{nombre de produits déstockés}} = \frac{20}{39} = 0.51$$

Cela implique que 51% des opérations de déstockage sont pour la livraison. Par conséquent, dans ce cas, 49% des cycles de la machine de déstockage sont destinés vers le

convoyeur de restockage. Pour évaluer l'écart entre cette solution et celle correspondante à des produits sans considération de leur classe de péremption, nous comparons les deux solutions.

Le nombre optimal trouvé des cycles de déstockage sans prise en considération de la péremption est de 33 cycles, avec un taux de livraison de 60%. Ainsi, la prise en considération des péremptions n'induit que $39 - 33 = 6$ cycles en plus. On constate alors, que la contrainte de péremption dans ce cas est moins pénalisante que dans les exemples précédents.

5. 6. Conclusion

Dans ce chapitre, nous avons présenté le contexte qui nous a incité à intégrer les dates de péremption des produits dans le pilotage dynamique du système de stockage/déstockage. Dès lors qu'il s'agit de produits périssables, une bonne gestion du stock, passe nécessairement par la réduction des péremptions. Par conséquent, le système de pilotage doit être en mesure de surveiller l'évolution des péremptions des produits dans le rack pour empêcher que des produits dépassent leur date de validité. Nous nous sommes proposé alors, de traiter un nouveau problème d'optimisation qui consiste à minimiser le nombre de cycles de déstockage tout en minimisant, par un choix judicieux des produits à déstocker, le risque de péremption dans le rack. Cela nous a conduit à formuler le problème en deux phases. Dans la première phase, l'objectif est de déterminer les produits candidats au déstockage au vu de leur date de péremption. L'objectif de la deuxième phase utilise les résultats du premier, pour déterminer les emplacements des produits à déstocker dans le rack. La résolution séquentielle des deux problèmes correspondant aux deux phases, donne les emplacements de produits à déstocker et le nombre de cycles de déstockage à effectuer. Notons que les solutions étaient trouvées rapidement avec des temps de résolution de l'ordre du dixième de seconde.

Conclusion générale

Dans ce travail, nous nous sommes intéressés au pilotage dynamique d'un système automatisé de stockage/déstockage (AS/RS) à convoyeur gravitationnel. La contribution des travaux que nous avons réalisés est double. La première porte sur le développement d'un modèle réseau de Pétri coloré (RdPC) pour la modélisation de la dynamique de l'AS/RS à convoyeur gravitationnel, c'est un modèle qui est capable de donner à chaque instant l'état du système. La deuxième réside dans la proposition d'une méthode d'optimisation du nombre de cycles à effectuer par la machine de déstockage pour satisfaire une demande de clients. Nous avons constaté que bien que l'AS/RS à convoyeur gravitationnel est utilisé dans l'industrie, peu d'études lui ont été consacrés. Ce travail fait suite à celui réalisé dans (Sari, 2003) concernant la modélisation et l'évaluation de performances de l'AS/RS à convoyeur gravitationnel.

Le modèle RdPC que nous avons développé pour l'AS/RS a été motivé par deux objectifs :

1. Le premier consiste dans le fait que les RdPC permettent bien de représenter la dynamique complexe d'un système, en offrant une meilleure puissance de représentation dû à la notion de couleur.
2. Le deuxième consiste dans le fait que le formalisme des réseaux de Pétri permet de vérifier et valider le modèle (vivacité, réversibilité, atteignabilité par exemple). Ce deuxième objectif nous a semblé très important et c'est le besoin de pouvoir établir des preuves formelles de propriétés qui nous a conduit au choix des réseaux de Pétri. D'autant plus que la validation des modèles FIFO existe abondamment dans la littérature. Nous garantissons ainsi le fonctionnement correct du modèle en dehors de toute simulation.

Fort d'un tel formalisme de modélisation, nous avons ensuite proposé un système de pilotage en boucle fermée utilisant ce modèle comme observateur d'état et un optimiseur permettant de sélectionner les emplacements de produits à déstocker dans le rack pour satisfaire une demande. Le modèle RdPC fournit l'état du rack à l'optimiseur par le biais de son marquage.

Le rôle principal de l'optimiseur est la génération de la matrice de décision, déterminant une séquence spécifique des emplacements de casiers à déstocker. Ceci en tenant compte de la demande des clients et de l'état du rack. En effet, l'optimiseur reçoit en entrée, la demande périodique et l'état du rack fourni par le modèle RdPC, et génère les décisions de contrôle en sortie.

Basée sur une formulation en programme en nombres entiers, la démarche d'optimisation que nous avons développée donne après résolution du modèle deux résultats. Elle donne les emplacements optimaux de produits à déstocker pour satisfaire la demande de clients, réalisant le minimum de cycles de déstockage. En même temps elle donne le nombre de cycles de déstockage à effectuer. L'avantage de cette méthode d'optimisation est qu'elle donne le nombre minimum de cycles de la machine de déstockage permettant de satisfaire le lot de la demande des clients. Par conséquent, le taux de livraison se trouve maximisé et le taux de restockage minimisé, ce qui permet à la machine de déstockage de fonctionner plus efficacement avec un meilleur débit de sortie du système.

Un autre point fort de cette méthode se caractérise par les temps de calcul qui sont de l'ordre du dixième de seconde pour les cas concrets que nous avons traités. Ce qui la place très bien dans le contexte du pilotage temps réel.

Afin de prévoir toute variabilité dans la demande et offrir plus de réactivité, nous avons envisagé deux modes de fonctionnements du système de contrôle dont le choix est conditionné par l'intensité de la demande. Nous avons ensuite proposé un algorithme d'ordonnancement exploitant la matrice de décision issue de l'optimiseur.

Les résultats que nous avons obtenus ont été approuvés par leur publication dans (Hachemi et Alla, 2007).

Dans un premier temps nous nous sommes intéressés au type de produit comme donnée principale de celui-ci. Cependant, un autre enjeu est apparu quand on considérait le caractère périssable de certains produits comme les médicaments par exemple. Il s'en est suivi dans un second temps qu'il est indispensable de tenir compte, en plus du type de produit, de l'information de sa date de péremption. Nous avons alors procédé à l'intégration de la date de péremption du produit dans le modèle RdPC et de modifier en conséquence le bloc d'optimisation dans le système de pilotage. Nous nous sommes proposé donc, de traiter un nouveau problème d'optimisation qui consiste à minimiser le nombre de cycles de déstockage

tout en minimisant, par un choix judicieux des produits à déstocker, le risque de péremption dans le rack. Cela nous a conduit à formuler le problème en deux phases. Dans la première phase, l'objectif est de déterminer les produits candidats au déstockage au vu de leur date de péremption. L'objectif de la deuxième phase utilise les résultats du premier, pour déterminer les emplacements des produits à déstocker dans le rack. La résolution séquentielle des deux problèmes correspondant aux deux phases, donne les emplacements de produits à déstocker et le nombre de cycles de déstockage à effectuer. Les temps de calcul de cette deuxième méthode n'ont pas augmenté significativement et restent toujours de l'ordre du dixième de seconde.

Au terme de ces travaux de recherche, plusieurs perspectives peuvent se dégager :

- Etant donné que nous avons considéré que toute la pénalité se portait sur le nombre de cycles de déstockage et par conséquent le nombre de cycles de restockage, il sera envisageable d'ajouter une autre contrainte qui permet éventuellement de choisir les casiers les plus proches de la station de livraison lorsque cela n'induit pas plus de cycles pour la machine de déstockage.
- Nous avons constaté dans le cas de produits périssables, que parfois le taux de livraison est significativement pénalisé par rapport au taux de livraison sans prise en compte de la péremption. Par conséquent, il faudra penser à une politique de stockage basée sur la règle FIFO en fonction des dates de péremption. Cela n'est pas toujours évident du fait que les cycles de restockage peuvent perturber la disposition des produits dans le rack.

Par ailleurs, et dans une vision plus générale, le contexte des applications dans le milieu hospitalier semble prometteur à l'utilisation des systèmes automatisés de stockage/déstockage. De ce fait, il sera intéressant d'envisager un système automatisé d'approvisionnement en médicaments et dispositifs médicaux pour les différents plateaux techniques.

Bibliographie

- Alla H., Réseaux de Petri Colorés et réseaux de Petri Continus : Application à l'Etude des Systèmes à Evénements Discrets. Thèse de doctorat, Institut National polytechnique de Grenoble, France, 1987.
- Amato F., Basile F., Carbonne C., « An approach to control automated warehouse systems », *Control Engineering Practice*, 13, 2005, p. 1223-1241.
- Anthony, R., Planning and control systems: a framework for analysis. *Harvard University Press*, Cambridge, 1965.
- Aravindan N. J. P., Ponnambalam S. G., "Optimal Random Storage Allocation for an AS/RS in an FMS", *International Journal of Advanced Manufacturing Technology*, 1998, 14, p. 116-132.
- Ashayeri J., Heuts R. M., Valkenburg M. W. T., Veraart H. C., Wilhelm M. R., « A geometrical approach to computing expected cycle times for zone-based layouts in AS/RS ». *International Journal of Production Research*, 40, 2002, p. 4467-4483.
- Askin R. G., Standridge C. R., *Modeling and Analysis of Manufacturing Systems*, John Wiley & Sons, Inc. USA, 1993.
- Avriel M., Golanya B., *Mathematical Programming for Industrial Engineers*. Edition Marcel Dekker. USA, 1996.
- Baillargeon G., *Programmation linéaire appliquée, outils d'optimisation et d'aide à la décision*. Les éditions SMG, Canada, 1996.
- Balsan D., Laine M., Legendre C., Évolution de la part de la pharmacie dans les budgets hospitaliers publics 1997-2001, Document de travail, Ministère des affaires sociales, du travail et de la solidarité, France, 2004.
- Bozer Y. A., White J.A., « Travel time models for automated storage/retrieval system ». *IIE Transactions*, 16, 1984, p. 329-338.

- Bozer Y.A., White J.A., "Design and performance models for end-of-aisle order picking systems", *Management Science* 36(7), 1990, p. 852-866.
- Bozer Y.A., White, J.A., "A generalized design and performance analysis models for end-of-aisle order-picking systems", *IIE Transactions* 28(4), 1996, p. 271-280.
- Caron F., Marchet G., Perego A., "Optimal layout in low-level picker-to-part systems", *International Journal of Production Research*, 2000, 38(1), p.101-117.
- Ceai, Journal of Control Engineering and Applied Informatics, site: <http://ceaijournal.ics.pub.ro>, 2008.
- Chang D. T., Wen U. P., "The impact on rack configuration on the speed profile of the storage and retrieval machine", *IIE Transactions*, 1997, 29, p. 525- 531.
- Chang N., Kwon W. H., Park J., "Hardware implementation of real-time Petri-net-based controllers", *Control Engineering Practice*, 1998, 6, p. 889-895.
- Chang S H., Egbelu P. J., « Relative pre-positioning of storage/retrieval machines in automated storage/retrieval systems to minimize maximum system response time », *IIE Transactions*, 29, 1997, p. 303-312.
- Chang S. H., Egbelu P. J., "Multi Relative pre-positioning of storage/retrieval machines in automated storage/retrieval system to minimize expected system response time", *IIE Transactions*, 1997, 29, p. 313- 322.
- Chang T. H., Fu H. P., Hu K. Y., "Innovative application of an integrated multi-level conveying device to a mobile storage system", *International Journal of Advanced Manufacturing Technology*, 2006, 29, p. 962–968.
- Chen J., Chen F. F., "Performance Modelling and Evaluation of Dynamic Tool Allocation in Flexible Manufacturing Systems using Coloured Petri Nets: An Object-Oriented Approach". *International Journal of Advanced Manufacturing Technology*, 2003, 21, p. 98–109.
- Chetty O. V. K., Reddy M. S., « Genetic algorithms for studies on AS/RS integrated with machines », *International Journal of Advanced Manufacturing Technology*, 22 (12), 2003, p. 932-940.
- David R., Alla H., *Discrete, Continuous and Hybrid Petri Nets*, Springer, Heidelberg Germany, October 2004.
- DGPSEGI, Les systèmes automatisés et robotisés utilisés pour la distribution des médicaments dans les établissements de santé au Québec, Rapport et recommandations du groupe de travail. Direction générale de la planification stratégique, de l'évaluation et de la gestion de l'information, direction générale du personnel réseau et ministériel, avril 2005.
- Dotoli M., Fanti M. P., « A coloured Petri net model for automated storage and retrieval systems serviced by rail-guided vehicles: a control perspective ». *International Journal of Computer Manufacturing*, vol. 18, n°2-3, 2005, p. 122-136.

- Duta, L., Contribution à l'étude de la conduite des systèmes de désassemblage, Thèse de Doctorat, Université Franche-Comté, France, 2006.
- Egbelu P.J., « Framework for dynamic positioning of storage/retrieval machines in an automated storage/retrieval system ». *International Journal of Production Research*, 31, 1991, p. 17-37.
- Eldemir F., Graves R. J., Malmborg C. J., "New cycle time and space estimation models for automated storage and retrieval system conceptualization", *International Journal of Production Research*, 42(22), 2004, p. 4767-4783.
- Elsayed E.A., Lee M. K., "Order processing in automated storage/retrieval systems with due dates", *IIE Transactions* 28, 1996, p. 567-577.
- Elsayed E.A., Lee M.-K., Kim S., Scherer E., "Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals", *International Journal of Production Research* 31(3), 1993, p. 727-738.
- Eynan A., Rosenblatt M.J., "An interleaving policy in automated storage/retrieval systems", *International Journal of Production Research* 31(1), 1993, p. 1-18.
- Feldmann K., Schnur C., Colombo W., "Modularised, Distributed Real-time Control of Flexible Production Cells, using Petri Nets", *Control Eng. Practice*, 4(8), 1996, p. 1067-1078.
- Feldmann K., Schnur C., Colombo W., "Monitoring of flexible production systems using high-level Petri net specifications", *Control Engineering Practice*, 1999, 7, p. 1449-1466.
- Foley R. D., Frazelle E. H., Park B. C., "Throughput bounds for miniload automated storage/retrieval systems", *IIE Transactions*, 2002, 34, p. 915- 920.
- Galasso F., Aide à la planification dans les chaînes logistiques en présence de demande flexible. Thèse de Doctorat en Systèmes industriels, Institut National Polytechnique de Toulouse, France, 2007.
- Ganeshan R., Harrison T., An introduction to supply chain management, *Penn State University, department of Management Science and Information Systems*, 1995.
- Ganeshan R., Tayur S., Magazine M., "A taxonomic review of supply chain management", dans "Quantitative models for supply chain management modelling", *Operations Research Management Science*, 1999, p. 838-879.
- Ghoul H. R., Modélisation et Conduite des Systèmes de Production par les réseaux de Petri. Thèse de doctorat d'Etat, Université Badji Mokhtar, Annaba, Algérie, 2003.
- Giard V., Gestion de la production et des flux, Economica, 2003.
- Graves S.C., Hausman W.H., Schwarz L.B., « Storage/Retrieval Interleaving in Automatic Warehouse Systems ». *Management Science*, 23, 1977, p. 935-945.

- Hachemi K., Alla H., 2007a, « Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel », accepté au *Journal Européen des Systèmes Automatisés*, décembre 2007.
- Hachemi, K., Alla, H., 2007b, "Dynamic control of a flow-rack Automated Storage and Retrieval system", The 16th International Conference on Control Systems and Computer Science CSCS16, 22 may-26 may 2007, Bucharest, Romania.
- Han M. H., McGinnis L. F., Shieh J. S., White, J. A., « On Sequencing Retrievals in an Automated Storage/Retrieval System », *IIE Transactions*, 19(1), 1987, p. 56-66.
- Hausman W.H., Schwarz L.B., Graves S.C., « Optimal Storage Assignment in Automatic Warehouse Systems ». *Management Science*, 22(6), 1976, p. 629-638.
- Henry S., Synthèse de lois de Commande pour la Configuration et la reconfiguration des Systèmes Industriels Complexes. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 2005.
- HOSPIMEDIA, Dispensation automatisée de médicaments - L'hôpital Rangueil de Toulouse teste le robot Homerus. Page Web, <http://www.hospimedia.fr/r/dispensation-automatisee-de-medicaments-l-hopital-rangueil-de-toulouse-teste-le-robot-omerus/1530.html>, 2002.
- Hsieh S., Tsai K. C., "A BOM Oriented Class-Based Storage Assignment in an Automated Storage/retrieval System", *International Journal of Advanced Manufacturing Technology*, 2001, 17, p. 683-691.
- Huang G. Q., Lau, J. S .K., Mak K. L., "The impacts of sharing production information on supply chain dynamics: a review of the literature". *International Journal of Production Research*, 41(7), 2003, p. 1483—1517.
- Hwang H., Hahn K. H., "An optimal procurement policy for items with an inventory level-dependent demand rate and fixed lifetime", *European Journal of Operational Research*, 127, 2000, p. 537-545.
- Hwang H., Lee S.B., "Travel-time models considering the operating characteristics of the storage/retrieval system", *International Journal of Production Research* 28(10), 1990, p. 1779-1789.
- Hwang H., Lim J. M., « Deriving an Optimal Dwell Point of the Storage/Retrieval Machine in an Automated Storage/Retrieval System », *International Journal of Production Research*, 31(11), 1993, p. 2591-2602.
- Hwang H., Song J.Y., "Sequencing picking operations and travel time models for man-on-board storage retrieval system", *International Journal of Production Economics* 29, 1993, p. 75-88.
- Keserla A., Peters B., "An analysis of dual shuttle automated storage/retrieval systems", *Journal of Manufacturing Systems*, 13(6), 1994, p. 424-434.
- Kim B. I., Heragu S. S., Graves R.J., Onge A. ST., « Realization of short cycle time in warehouse replenishment and order picking ». *International Journal of Production Research*, 41, 2003, p. 349-364.

- Kirsch S. E., « Automated Storage and Retrieval - The Next Generation: How Northridge's Success is Spurring a Revolution in Library Storage and Circulation, », *ACRL 1999 National Conference Papers* (1999), <http://www.ala.org/acrl/kirsch.pdf>.
- Kouvelis P., Papanicolaou V., « Expected travel time and optimal boundary formulas for a two-class-based automated storage/retrieval system ». *International Journal of Production Research*, 33(10), 1995, p. 2889-2905.
- Kulturel S., Ozdemirel N. E., Sepil C., Bozkurt Z., “Experimental investigation of shared storage assignment policies in automated storage/retrieval systems”, *IIE Transactions*, 1999, 31, p. 739- 749.
- Labadi K., Contribution à la Modélisation et à l'Analyse de Performances des Systèmes Logistiques à l'aide d'un nouveau modèle de Réseaux de Petri Stochastique. Thèse de doctorat, Université de Technologie de Troyes, France, 2005.
- Lee H. F., Schaefer S. K., « Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings ». *International Journal of Production Research*, 34, 1996, p. 2943-2962.
- Lee H. F., Schaefer S. K., « Sequencing methods for Automated Storage and Retrieval systems with dedicated storage ». *Computers and Industrial Engineering*, 32(2), 1997, p. 351-362.
- Lerher T., Sraml M., Kramberger J., Potrc I., Borovinsek M., Zmazek B., “Analytical travel time models for multi aisle automated storage and retrieval systems”, *International Journal of Advanced Manufacturing Technology*, 30, 2006, p. 340-356.
- Lin C. W. R., Tsao Y. Z., “Dynamic availability-oriented control of the automated storage/retrieval system. A computer integrated manufacturing perspective”, *International Journal of Advanced Manufacturing Technology*, 2006, 29, p. 948–961.
- Lin J. T., Lee C. C., “A Petri net-based integrated control and scheduling scheme for flexible manufacturing cells”, *Computer Integrated Manufacturing Systems*, 10(2), 1997, p. 109-122.
- Linn R. J., Wysk R. A., (1990a), « An expert system framework for automated storage/retrieval system » *Computers & Industrial Engineering*, 18(1), 1990, p. 37-48.
- Linn R. J., Wysk R. A., (1990b), « An expert system based controller for an automated storage/retrieval system » *International Journal of Production Research*, 28(4), 1990, p. 735-756.
- Lopez P., Roubellat F., *Ordonnancement de la production*, Paris, Editions Hermes, 2001.
- Mahjan S., Rao B. V., Peters B. A., “A retrieval sequencing heuristic for miniload end-of-aisle automated storage/retrieval systems”, *International Journal of Production Research*, 36(6), 1998, p. 1715-1731.
- Malmberg C.J., "Optimization of cube-per-order index warehouse layouts with zoning constraints", *International Journal of Production Research*, 1995, 33(2), p. 465-482.

- Malmberg C.J., "Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design", *International Journal of Production Research*, 2001, 39(3), p. 511-527.
- Malmberg C.J., "Storage assignment policy tradeoffs", *International Journal of Production Research*, 1996, 34(2), p. 363-378.
- Manzini R., Gamberi M., Regattieri A., "Design and control of an AS/RS", *International Journal of Advanced Manufacturing Technology*, 2006, 28, p. 766-774.
- Meller R. D., Mungwattana A., "Multi-shuttle automated storage/retrieval systems", *IIE Transactions*, 1997, 29, p. 925- 938.
- Meller R. D., Mungwattana A., « AS/RS dwell-point strategy selection at high system utilization: a simulation study to investigate the magnitude of the benefit ». *International Journal of Production Research*, vol. 43, n°24, 2005, p. 5217-5227.
- Meller, R., Klote, J., « A throughput model for carousel/VLM pods », *IIE Transactions*, 36(8), 2004, p. 725.
- Miller T., *Hierarchical operations and supply chain planning*, Springer, 2001.
- Moon G., Kim, G., « Effects of relocation to AS/RS storage location policy with production quantity variation ». *Computers & Industrial Engineering*, 40, 2001, p. 1-13.
- Müller M., The Use of Information Technologies in Supply Chains – A Transaction Cost Analysis, in *Strategy and Organization in Supply Chains*, (Eds : Seuring, Müller, Goldbach, Schneidewind), Physica-Verlag, Heidelberg New York, 2003.
- Muralidharan B., Linn R.J., Pandit, R., « Shuffling heuristics for the storage location assignment in an AS/RS ». *International Journal of Production Research*, 33(6), 1995, p. 1661-1672.
- Neos., Serveur d'optimisation pour problèmes au format AMPL, disponible au site <http://neos.mcs.anl.gov/neos/solvers/milp:scip/AMPL.html>, 2007.
- Neos., Serveur d'optimisation pour problèmes au format AMPL, disponible au site <http://neos.mcs.anl.gov/neos/solvers/milp:scip/AMPL.html>, 2007.
- Pan C.-H., Wang C. H., "A framework for the dual command cycle model in automated warehousing systems", *International Journal of Production Research* 34(8), 1996, p. 2099-2117.
- Park B. C., "An optimal dwell point policy for automated storage/ retrieval systems with uniformly distributed, rectangular racks", *International Journal of Production Research*, 39(7), 2001, p. 1469-1480.
- Park B. C., "Performance of automated storage/retrieval systems with non-square-in-time racks and two-class storage", *International Journal of Production Research*, 44(6), 2006, p. 1107-1123.

- Peters B. A., Smith J. S., Hale T.S., « Closed form models for determining the optimal dwell point location in automated storage/retrieval systems ». *International Journal of Production Research*, 34, 1996, p. 1757-1771.
- Petersen C.G., "An evaluation of order picking routing policies", *International Journal of Operations & Production Management*, 1997, 17, p.1098-1110.
- Petersen C.G., "The impact of routing and storage policies on warehouse efficiency", *International Journal of Operations & Production Management*, 1999, 19(10), p.1053-1064.
- Petersen C.G., Aase G.R., Heiser D.R., "Improving order picking performance through the implementation of class-based storage", *International Journal of Physical Distribution & Logistics Management*, 2004, 34(7), p.534-544.
- Proth J. M., Xie X., *Les réseaux de Petri pour la conception et la gestion des systèmes de production*, Editions Masson, France, 1995.
- Ramirez A. A. C., Contribution à la Modélisation et à la Gestion des Interactions Produit-Processus dans la Chaîne Logistique par l'Approche Produits Communicants. Thèse de Doctorat, Université Henri Poincaré, Nancy-I, France, 2006.
- Randhawa S. U., Shroff R., « Simulation-based design evaluation of unit load automated storage/retrieval systems ». *Computer & industrial Engineering*, 28 (1), 1995, p.77-79.
- Randhawa S.U., Mcdowell E.D., Wang W.T., « Evaluation of Scheduling rules for single and dual-dock automated storage/retrieval systems ». *Computers & Industrial Engineering*, 20(4), 1991, p. 401-410.
- Rosenblatt M.J., Eynan A., « Deriving the Optimal Boundaries for Class-based Automatic Storage/Retrieval Systems ». *Management Science*, 35(12), 1989, p. 1519-1524.
- Rosenblatt M.J., Roll Y., Zyser V., "A combined optimization and simulation approach for designing automated storage/retrieval systems", *IIE Transactions* 25(1), 1993, p. 40-50.
- Sari Z., Modélisation, Analyse et Evaluation des Performances d'un AS/RS à Convoyeur Gravitationnel. Thèse de doctorat d'Etat, Université Abou Bekr Belkaïd, Tlemcen, Algérie, 2003.
- Sari Z., Saygin C., Ghouali N., « Travel-time models for flow-rack automated storage and retrieval systems ». *International Journal of Advanced Manufacturing Technology*, 25, 2005, p. 979-987.
- Sarker B.R., Babu P.S., "Travel time models in automated storage/retrieval systems: A critical review", *International Journal of Production Economics* 40, 1995, p. 173-184.
- Seidmann A., "Intelligent control schemes for automated storage and retrieval systems", *International Journal of Production Research* 26(5), 1988, p. 931-952.
- Shapiro J., "On the connections among activity based costing, optimization models for strategic decision support and the resource-based view of the firm". *European Journal of Operational Research*, 118 (2), 1999, p. 295-314.

- Singh N., *Systems approach to Computer-Integrated Design and Manufacturing*, John Wiley & Sons, Inc. USA, 1996.
- Tricas F., Martoanez J., “Distributed control systems simulation using high level Petri nets”, *Mathematics and Computers in Simulation*, 1998, 46, p. 47- 55.
- Van den berg J. P., Gademan A.J.R.M., « Optimal routing in an automated storage/retrieval system with dedicated storage ». *IIE transactions*, 31, 1999, p. 407-415.
- Van den berg J. P., Gademan A.J.R.M., « Simulation study of an automated storage/retrieval system ». *International Journal of Production Research*, 38(6), 2000, p. 1339-1356.
- Venkatesh K., Ilyas M., “Real-time Petri nets for modelling, controlling, and simulation of local area networks in flexible manufacturing systems”. *Computers and Industrial Engineering*, 28(1), 1995, p. 147-162.
- Wen U. P., Chang, D. T., « The impact of acceleration/deceleration on travel-time models in class-based automated S/R systems », *IIE Transactions*, 33(7), 2001, p. 599-608.
- Woronoff-Lemsi M. C., Grall J. Y., Monier B., Bastianelli J. P., *Le médicament à l’hôpital*, rapport, France, mai 2003.
- Yin L., Rau H., “Dynamic selection of sequencing rules for a class-based unit-load automated storage and retrieval system”, *International Journal of Advanced Manufacturing Technology*, 2006, 29, 1259-1266.
- Zhang W., Freiheit T., Yang H., “Dynamic scheduling in flexible assembly system based on timed Petri nets model”, *Robotics and Computer-Integrated Manufacturing*, 21, 2005, p. 550-558.
- Zhou M., Dicesare F., *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, USA, 1993.

Table des Matières

Introduction générale.....	1
Chapitre 1 Les systèmes automatisés de stockage/déstockage.....	5
1. 1. Introduction	5
1. 2. Configuration d'un AS/RS	6
1. 2. 1. Composants et terminologie utilisée	6
1. 2. 2. Fonctionnement d'un AS/RS	7
1. 3. Types d'AS/RS.....	8
1. 3. 1. L'AS/RS à charge unitaire	8
1. 3. 2. L'AS/RS à mini-charge	9
1. 3. 3. L'AS/RS à carrousel.....	10
1. 3. 4. L'AS/RS à convoyeur gravitationnel	10
1. 4. Avantages des AS/RS.....	11
1. 5. Politiques de stockage	13
1. 5. 1. Stockage dédié.....	13
1. 5. 2. Stockage aléatoire ouvert	14
1. 5. 3. Stockage aléatoire par classe.....	14
1. 6. Modèles de temps de transport dans les AS/RS	15
1. 6. 1. Distribution des temps de transport	15
1. 6. 2. Temps moyen de transport	18
1. 7. Etat de l'art	20
1. 8. Conclusion.....	22

Chapitre 2 Notions de base sur les réseaux de Petri 25

2. 1. Introduction	25
2. 2. Les notions de base des réseaux de Petri.....	25
2. 2. 1. Marquage.....	27
2. 2. 2. Franchissement des transitions.....	28
2. 3. Quelques définitions particulières	30
2. 3. 1. Réseaux de Petri autonomes et non autonomes	30
2. 3. 2. Transition source et transition puits	31
2. 3. 3. Circuits élémentaires et boucles	31
2. 3. 4. Réseaux de Petri à capacité finie.....	32
2. 3. 5. Verrou et trappe.....	33
2. 4. Arbre des marquages, arbre de recouvrement et graphe de recouvrement	33
2. 5. Matrice d'incidence et équation d'état	34
2. 6. Propriétés comportementales et structurelles	35
2. 6. 1. Propriétés comportementales	35
2. 6. 2. Propriétés structurelles	37
2. 7. Exemples de modélisation.....	39
2. 7. 1. Convoyeurs.....	39
2. 7. 2. Systèmes de stockage	40
2. 8. Abréviations et extensions des réseaux de Petri.....	42
2. 9. Réseaux de Petri colorés	42
2. 9. 1. Présentation intuitive et notion de couleur.....	43
2. 9. 2. Définition des réseaux de petri colorés	44
2. 9. 3. Evolution du marquage dans un réseau de petri coloré.....	45

2. 10. Conclusion.....	50
Chapitre 3 Modélisation et pilotage de l'AS/RS à convoyeur gravitationnel	51
3. 1. Introduction	51
3. 2. Réactivité des systèmes de stockage/déstockage	51
3. 3. Système de décision	52
3. 3. 1. Niveaux décisionnels.....	53
3. 4. Ordonnancement des opérations de déstockage.....	57
3. 4. 1. Les éléments d'ordonnancement	57
3. 4. 2. Les étapes d'ordonnancement	59
3. 5. AS/RS à convoyeur gravitationnel.....	60
3. 6. Structure du système de pilotage.....	63
3. 6. 1. Modèle réseaux de Petri coloré	64
3. 6. 2. Pilotage en boucle fermée	68
3. 6. 3. Mode de traitement des demandes	69
3. 7. Algorithme d'ordonnancement	70
3. 8. Conclusion.....	71
Chapitre 4 Optimisation des cycles de déstockage.....	73
4. 1. Introduction	73
4. 2. Position du problème d'optimisation	73
4. 3. Programmation en nombres entiers.....	74
4. 3. 1. Formulation mathématique d'un programme en nombres entiers.....	74
4. 3. 2. Résolution graphique d'un programme en nombres entiers.....	76
4. 3. 3. Programme en nombres entiers à variables binaires	76

4. 3. 4. Méthode de résolution par séparations et évaluations successives	78
4. 4. Formulation du problème d'optimisation des cycles de déstockage.....	84
4. 5. Etude de cas.....	88
4. 5. 1. Exemple 1.....	88
4. 5. 2. Exemple 2.....	92
4. 5. 3. Exemple 3.....	95
4. 6. Conclusion.....	98
Chapitre 5 Intégration des dates de péremption des produits dans le pilotage	99
5. 1. Introduction	99
5. 2. Contexte et enjeux.....	99
5. 3. Intégration des dates de péremption dans le modèle RdPC de l'AS/RS à CG.....	102
5. 4. Problème d'optimisation intégrant les dates de péremption	105
5. 4. 1. Problème.....	105
5. 4. 2. Formulation mathématique du problème d'optimisation	106
5. 5. Etude de cas.....	110
5. 5. 1. Exemple 1.....	110
5. 5. 2. Exemple 2.....	113
5. 5. 3. Exemple 3.....	117
5. 6. Conclusion.....	121
Conclusion générale	123
Bibliographie.....	127