



**HAL**  
open science

## Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel

Khalid Hachemi, Hassane Alla

► **To cite this version:**

Khalid Hachemi, Hassane Alla. Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel. *Journal Européen des Systèmes Automatisés (JESA)*, 2008, 42 (5/2008), pp.487-508. hal-00333689

**HAL Id: hal-00333689**

**<https://hal.science/hal-00333689>**

Submitted on 23 Oct 2008

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

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

---

# Pilotage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel

**Khalid Hachemi\* \*\* – Hassane Alla\***

*\* Laboratoire d'Automatique de Grenoble*

*ENSIEG – BP 46*

*F-38402 St Martin d'Hères Cedex*

*{khalid.hachemi, hassane.alla}@lag.ensieg.inpg.fr*

*\*\* IMSI, Université d'Oran et Laboratoire d'Automatique de Tlemcen*

---

*RÉSUMÉ. Dans cet article, nous proposons une structure de pilotage d'un système automatisé de stockage/déstockage (AS/RS) à convoyeur gravitationnel basé sur un modèle réseau de Petri Coloré (RdPC). Le modèle RdPC de l'AS/RS est utilisé comme observateur d'état pour fournir l'état du rack dans un système de contrôle en boucle fermée. Nous introduisons dans le système de contrôle un bloc d'optimisation comme processus de décision qui effectue périodiquement, une optimisation temps réel. L'objectif de la méthode d'optimisation proposée est de trouver les meilleurs emplacements à déstocker dans le rack pour un lot de demande de clients et ceci en minimisant le nombre de cycles de déstockage et par conséquent d'améliorer le débit de sortie du système.*

*ABSTRACT. In this paper we propose a control scheme based on Coloured Petri net (CPN) for a flow-rack automated storage and retrieval system. The AS/RS is modelled using Coloured Petri net. The developed model is used to capture and provide the rack state. We introduce in the control system an optimization module as decision process which performs a real-time optimization working on discrete events time scale. The objective is to find bin locations for the retrieval requests by minimizing the total number of retrieval cycles for a batch of requests and thereby increase the system throughput. By solving the optimization model, the proposed method gives according to customers request and the rack state, the best bin locations for retrieval, i.e. allowing at the same time to satisfy the customers request and carrying out the minimum of retrieval cycles.*

*MOTS-CLÉS: AS/RS à convoyeur gravitationnel, Réseaux de Petri colorés, débit, optimisation.*

*KEYWORDS: Flow-rack automated storage and retrieval systems, Coloured Petri nets, throughput, optimization.*

---

## 1. Introduction

Un système automatisé de stockage/déstockage (Automated Storage/Retrieval system, 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 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 des autres. 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. Pour atteindre cet objectif, les chercheurs ont développé plusieurs approches. Dans (Bozer et al., 1984), les auteurs développent des expressions pour les temps moyens pour le simple et le 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 de point de repos de la machine S/D. 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. précédent, (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. 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 interchangeable 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'ordonnement des opérations de stockage/déstockage en utilisant l'approche d'ordonnement 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 quelque intérêt. 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.

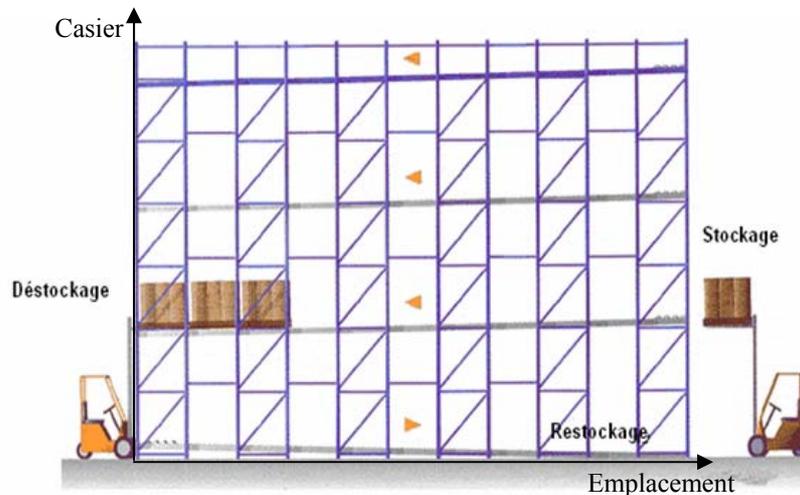
Pour ce faire, nous commencerons d'abord par une modélisation du système par les réseaux de Pétri colorés (RdPC), ensuite nous proposerons un schéma de commande qui exploite le modèle RdPC comme un modèle de référence (observateur d'état). Nous verrons que le système étudié se comporte comme une file FIFO. Les règles de décisions seront générées par un module d'optimisation du nombre de cycles de déstockage.

La suite de l'article est organisée comme suit. Le fonctionnement de l'AS/RS à convoyeur gravitationnel est décrit dans la section 2. La section 3 expose le modèle RdPC et le couplage avec le module d'optimisation. Dans la section 4, nous traitons le modèle mathématique du problème d'optimisation du nombre de cycles de

déstockage ainsi qu'un exemple d'application. Enfin, les conclusions sont présentées dans la section 5.

## 2. Système automatisé de stockage/déstockage à convoyeur gravitationnel

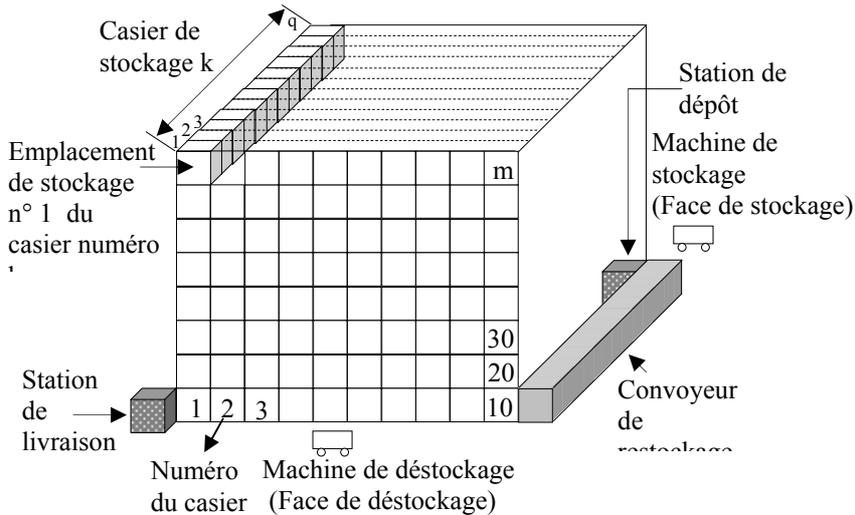
L'AS/RS à convoyeur gravitationnel est composé d'un rack profond regroupant un ensemble de casiers pouvant contenir plusieurs produits placés les uns à la suite des autres. Chaque casier est muni d'un convoyeur gravitationnel à base de rouleaux ou roues libres inclinées de façon à permettre le glissement 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 1 donne la structure d'un tel AS/RS industriel.



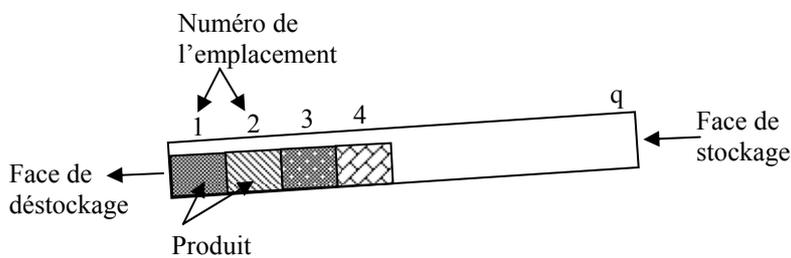
**Figure 1.** *Vue de profil d'un AS/RS à convoyeur gravitationnel*

La figure 2 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 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 restockage pour être réintroduit dans le rack. Le produit demandé sera déposé dans la station de livraison.



**Figure 2.** Constitution typique d'un rack à convoyeur gravitationnel



**Figure 3.** 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 représenté 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 :

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

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

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 on ne choisissant pas forcément les produits qui ont l'indice d'emplacement le plus faible. 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 (Section 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. 3.

### 3. 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.

#### 3.1. Modèle Réseau de Petri coloré

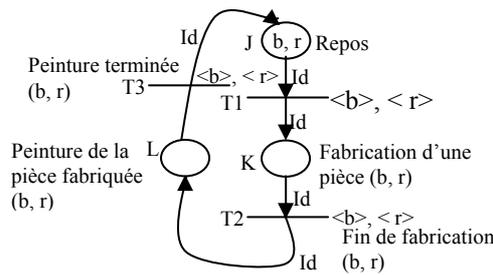
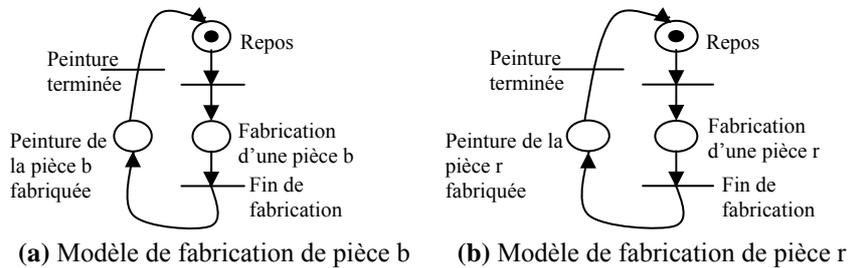
Les Réseaux de Petri Colorés (RdPC) sont une extension des réseaux de Petri qui permet de modéliser des systèmes complexes, ils constituent une abréviation des réseaux de Petri ordinaires. Dans un RdPC, les places sont associées à des identificateurs ou "couleurs". Chaque transition peut être franchie 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.

Un RdP Coloré N est un sextuplet  $N = \langle P, T, Pré, Post, M_0, C \rangle$ . 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. Pour plus de détails, voir (David et al., 2004).

Pour illustrer le franchissement d'une transition par rapport à une couleur, nous considérons l'exemple de la figure 4. La figure 4.a. donne le réseau de Petri modélisant la fabrication et la peinture d'une pièce du type b par une machine. Similairement, la figure 4.b. représente la fabrication et la peinture d'une pièce du type r par la même machine.

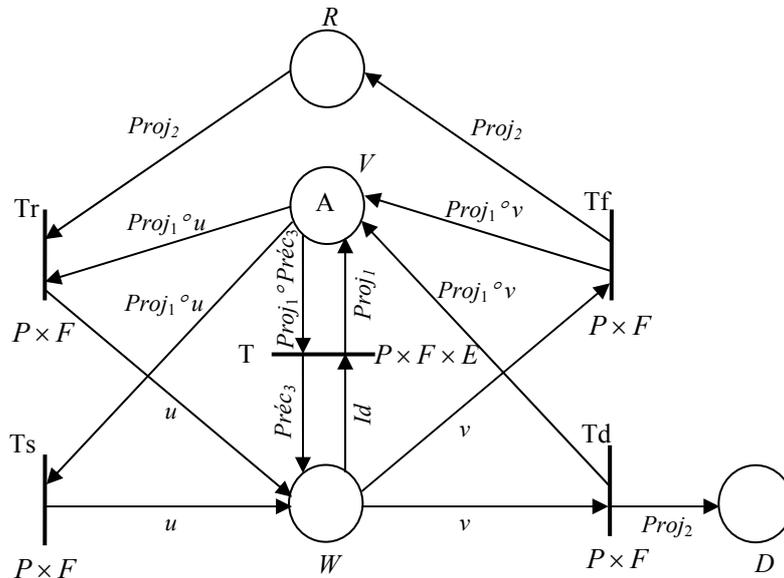


(c) Modèle coloré de fabrication des pièces b et r

**Figure 4.** Réseaux de Petri de la fabrication et peinture d'une pièce

La figure 4.c. représente le réseau de Petri coloré obtenu par pliage des deux précédents. Le franchissement de la transition T1 du réseau de Petri coloré peut se faire selon deux couleurs ;  $\langle b \rangle$  et  $\langle r \rangle$ . Dans ce RdPC la transition T1 est validée par rapport aux deux couleurs  $\langle b \rangle$  et  $\langle r \rangle$ , car la place  $J$  contient la couleur  $\text{Id}(\langle b \rangle) = \langle b \rangle$  ainsi que la couleur  $\text{Id}(\langle r \rangle) = \langle r \rangle$  (la fonction identité ne produit aucun changement de couleur). Le tir de T1 par rapport à la couleur  $\langle b \rangle$  consiste à déposer un jeton de couleur  $\text{Id}(\langle b \rangle) = \langle b \rangle$  dans la place  $K$  et retirer un jeton  $\langle b \rangle$  de la place  $J$ , ainsi la machine entame la fabrication d'une pièce du type  $b$ . Similairement, le tir de T1 par rapport à la couleur  $\langle r \rangle$  consiste à déposer un jeton de couleur  $\text{Id}(\langle r \rangle) = \langle r \rangle$  dans la place  $K$  et retirer un jeton  $\langle r \rangle$  de la place  $J$ . Notons que dans cet exemple, l'ordre de fabrication des pièces  $b$  et  $r$  n'est pas spécifié.

Afin de représenter la dynamique de l'AS/RS, 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 5. L'état du système est donné à tout instant par le marquage du RdPC.



**Figure 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 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 :

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

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

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

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

Les arcs sont associés à des fonctions et 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  $fk$  :

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

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

Le tableau 1 résume les fonctions utilisées, portées sur les arcs (Alla, 1987).

Ensemble de départ	Ensemble d'arrivée	Fonction	Définition	Commentaire
C	C	Id	$\text{Id}(\langle c_i \rangle) = \langle c_i \rangle$	Identité
$C^m$	$C^m$	$\text{Préc}_j$	$\text{Préc}_j(\langle c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im} \rangle)$ $= \langle c_{i1}, c_{i2}, \dots, c_{i(i-1)}, \dots, c_{im} \rangle$	Précédent pour un m-uplet
$C^m$	$C^{m-1}$	$\text{Proj}_j$	$\text{Proj}_j(\langle c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im} \rangle)$ $= \langle c_{i1}, \dots, c_{i(j-1)}, c_{i(j+1)}, \dots, c_{im} \rangle$	Projection sur la j <sup>ème</sup> 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				

**Tableau 1.** Définitions des fonctions utilisées

Dans le RdPC de la figure 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 2.

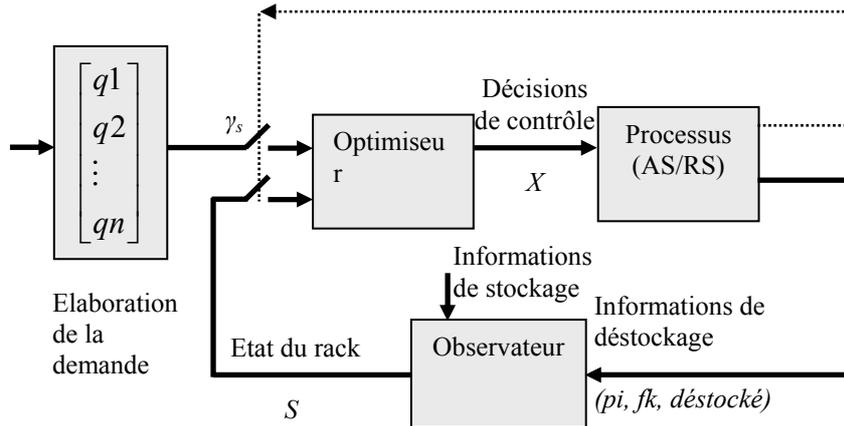
Place	Couleur	Interprétation
$V$	$\langle fk, ej \rangle$	L'emplacement $ej$ du casier $fk$ est vide
$W$	$\langle pi, fk, ej \rangle$	L'emplacement $ej$ du casier $fk$ contient un produit de type $pi$
$D$	$\langle pi \rangle$	La station de livraison contient un produit de type $pi$
$R$	$\langle pi \rangle$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

**Tableau 2.** Interprétation des places et transitions du modèle RdPC

### 3.2. Pilotage en boucle fermée

La figure 6 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 6.** 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 6) est éventuellement utilisée pour adapter la période de traitement des demandes  $\gamma_s$ .

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.

### 3.3. 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.

#### 3.3. 1. Fonctionnement à période d'échantillonnage fixe

Dans ce mode de fonctionnement, la période de traitement des demandes (période d'échantillonnage)  $\gamma_s$  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  $\gamma_s \geq qg \cdot (tm + tlu) + tc$ .

$tm = \max\left(\frac{x_m}{s_h}, \frac{y_m}{s_v}\right)$ , 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 ( $\text{ms}^{-1}$ ),

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

$q_g$  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  $\gamma_s = q_g \cdot (t_m + t_{lu}) + t_c$ .

### **3.3. 2. Fonctionnement à période d'échantillonnage adaptative**

Dans ce cas la période d'échantillonnage  $\gamma_s$  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  $\gamma_s$  du fait que  $\gamma_s$  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  $\gamma_s$  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 6 est utilisé pour activer une nouvelle acquisition d'un lot de demande sans attendre la fin de la période  $\gamma_s$  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.

## **4. Bloc d'optimisation**

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.

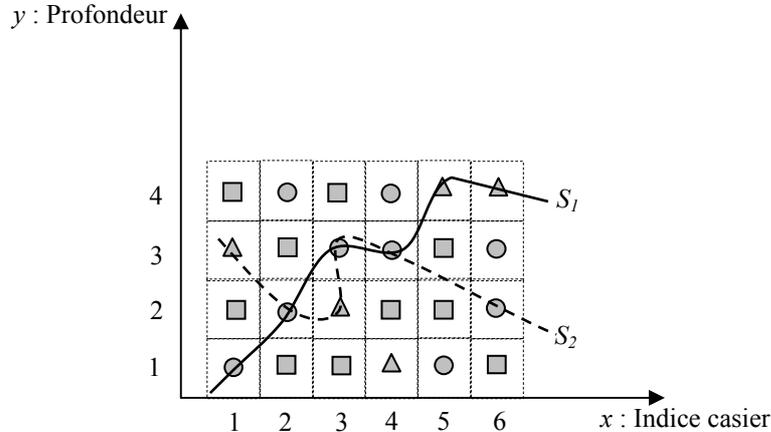
La solution consiste à rechercher les emplacements dans les casiers, contenant les produits demandés, et permettant d'employer le minimum de cycles de déstockage. Il est évident qu'un minimum de cycles de déstockage induit forcément un minimum de cycles de restockage. De plus, la considération du nombre 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 7 représentant sur un plan x-y, un rack à six casiers de quatre emplacements chacun. Le rack contient trois types de produits symbolisé 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 (Section 2). Deux solutions 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 7.** 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 8.

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 du  $X_{kj}$  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 MD.

#### 4.1. Etude de cas

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,

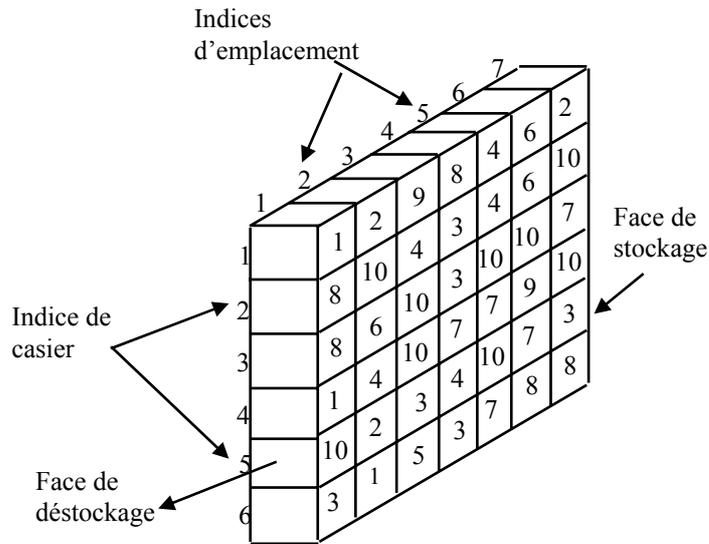
$q = 7$  ; le nombre d'emplacements de chaque casier,

$n = 10$  ; 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 8 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.



**Figure 8.** Etat du rack avec des types de produits dedans

D'où :

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

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).

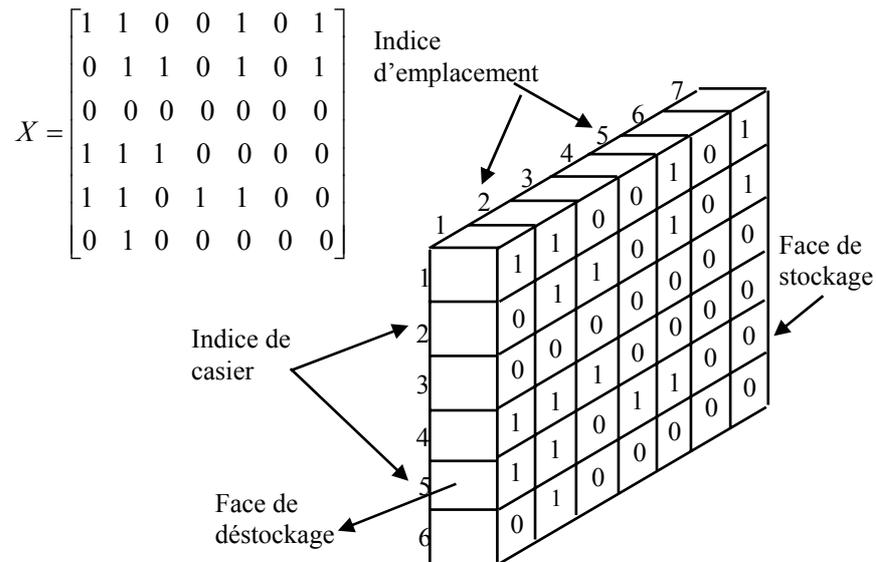
Pour évaluer les performances de la méthode d'optimisation, nous avons exécuté le programme sur la plateforme logicielle d'optimisation qui est disponible dans (Neos, 2007). Cette plateforme fournit un listing des différentes opérations exécutées, dont on peut résumer les principaux résultats, pour l'exemple 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 9.** *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 9 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 3.

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

**Tableau 3.** *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 4 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	$(f2, e2), (f2, e7), (f4, e3), (f5, e1), (f5, e5)$

**Tableau 4.** *Emplacements des produits à déstocker par type*

## 5. Conclusions

Dans cet article, nous avons présenté une structure de contrôle d'un AS/RS à convoyeur gravitationnel basée sur un modèle réseau de Petri Coloré. Le modèle RdPC permet de représenter la dynamique du système et par conséquent d'appréhender sa complexité. Comme performance poursuivie, nous nous sommes attachés à minimiser le nombre total de cycles de déstockage et par conséquent le nombre de cycles de restockages qui sont des cycles très coûteux en temps. Pour ce faire, nous avons intégré dans une boucle de contrôle un bloc d'optimisation comme solutionneur de conflits, afin de sélectionner les emplacements de produits à

déstocker dans le rack. Le problème d'optimisation est formulé en termes de programme en nombres entiers. Ce bloc exploite le modèle RdPC, par le biais de son marquage, pour lire l'état du rack. La solution de ce problème donne les emplacements optimaux de produits à déstocker pour satisfaire la demande de clients, ainsi que le nombre de cycles de déstockage de la machine de déstockage. 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. Comme perspective à ce travail, nous envisageons d'intégrer les dates de péremptions des produits dans le problème de déstockage où en plus de la minimisation du nombre de cycles de déstockage, il faudra minimiser aussi le risque de péremption des produits dans le rack.

## 6. 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.
- Bozer Y. A., White J.A., « Travel time models for automated storage/retrieval system ». *IIE Transactions*, 16, 1984, p. 329-338.
- 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.
- David R., Alla H., *Discrete, Continuous and Hybrid Petri Nets*, Springer, Heidelberg Germany, October 2004.
- 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.
- 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.
- 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.
- 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

- 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.
- Neos., serveur d'optimisation pour problèmes au format AMPL, disponible au site <http://neos.mcs.anl.gov/neos/solvers/milp:scip/AMPL.html>, 2007.
- 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.
- 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
- Van den berg J. P., Gademant A.J.R.M., « Optimal routing in an automated storage/retrieval system with dedicated storage ». *IIE transactions*, 31, 1999, p. 407-415.