

Chargement de véhicules à l'aide d'un convoyeur

BAPTISTE PIERRE¹, REBAINE DJAMAL², BRIKA ZAYNEB¹

¹ École Polytechnique de Montréal
Département de Mathématiques et Génie Industriel
C.P. 6079, succ. Centre-ville, Montréal, Québec, Canada H3C3A7.
pierre.baptiste@polymtl.ca

² Université du Québec à Chicoutimi
Département d'Informatique et de Mathématique
Université du Québec à Chicoutimi
555, boul. de l'Université, Chicoutimi, Québec, Canada G7H 2B1.
djamal_rebaine@uqac.ca

Résumé - Cet article s'intéresse à l'optimisation de la prise de palettes sur un convoyeur à gravité lorsque chaque palette de la séquence à une destination précise et que l'équipement de transport peut retirer du flux au plus K palettes consécutives si elles ont toutes la même destination. La minimisation du nombre de rotations du transporteur s'apparente à une version en une dimension du Clickomania, jeu informatique connu pour être NP-complet dans sa version générale. En fait, le problème est simultanément une restriction du Clickomania à une seule dimension et une extension avec deux contraintes supplémentaires, la largeur d'une fenêtre de prise et la limite du nombre d'objets simultanément pris. Des bornes de performance sont proposées, tenant compte de la largeur de la séquence accessible à la prise et de la valeur du nombre de palettes. Des conditions de dominance sont données. On propose également un modèle mathématique permettant d'optimiser le nombre de rotations.

Mots clés – Picking, optimisation, chargement de véhicules, Samegame, Clickomania.

Abstract – This article is concerned with the optimization of pallets picking on a gravity conveyor in which each pallet has a specific destination and the picking facility can extract simultaneously as much as K consecutive pallets of the sequence, provided they have the same destination. For a given sequence of pallets, and the corresponding set of destinations, the objective function is to minimize the number of movements that completely unloads the conveyor to the trucks. This problem is related to the classical Clickomania Game, which is known to be NP-Hard. In fact, the former is a restriction of the latter in its one dimension version, however, with two additional constraints: a limit on the number of extracted pallets, and an accessibility window. We propose several bounds of the optimal solution, dominance properties and a mathematical model.

Keywords - Picking, optimization, truck loading, Samegame, Clickomania.

1 INTRODUCTION

Le problème étudié dans cet article se rencontre à l'interface entre un système de production de gros objets (ou de palettes) et des quais d'expéditions, lorsque le système de production alimente un convoyeur d'expédition en produits suffisamment volumineux et de taille standard devant partir dans des directions variées (donc être déplacés vers des quais différents) et que le système de manutention permet de retirer du convoyeur simultanément plusieurs produits consécutifs, allant tous dans la même direction. Souvent, ces équipements sont conçus de telle sorte qu'ils peuvent saisir indifféremment entre 1 et K produits consécutifs. Évidemment, plus ils en transportent simultanément, moins ils font de rotations entre le flux et la remorque.

La Figure 1 représente un tel système, avec un convoyeur transportant des produits de trois textures différentes dans trois directions différentes (les quais 1, 2 et 3) et un chariot pouvant prendre trois produits consécutifs.

Idéalement, si le système de transport permet de déplacer K produits consécutifs en une seule prise, le système de production devrait être ordonnancé pour produire des lots de produits allant dans la même direction d'une taille multiple de K . Ainsi, le système d'expédition est utilisé au maximum de sa productivité. Malheureusement, il arrive qu'il ne soit pas possible d'ordonnancer le système de production, ou que des problèmes de contrôle ou de qualité effectuent des retraits inopinés dans la séquence, ou encore que les critères d'ordonnancement de la production et les critères d'optimisation du chargement soient antagonistes. Il faudrait alors ordonnancer au mieux les mouvements des chariots de chargement pour minimiser le nombre de rotations.

Dans l'exemple de la Figure 1, la fenêtre de prise est de 12 objets, il y a 3 destinations et le chariot peut prendre de 1 à 3 objets consécutifs.

Ce problème industriel a été rencontré dans une entreprise de fabrication de produits électroménagers de grands gabarits. En

fait, ce produit était inextricablement couplé à un problème d'ordonnement de production (création de la séquence) et d'ordonnement des camions (en sortie).

Nous avons dans un premier temps proposé une solution heuristique à ce problème [Baptiste *et al.*, 2008], puis formalisé les doubles problèmes d'ordonnement par un problème de *cross-docking* [Maknoon et Baptiste, 2009], [Rim *et al.*, 2011]. En ce temps, les problèmes de *cross-docking* engendraient les plus grosses difficultés. Nous formalisons ici le problème de *picking*.

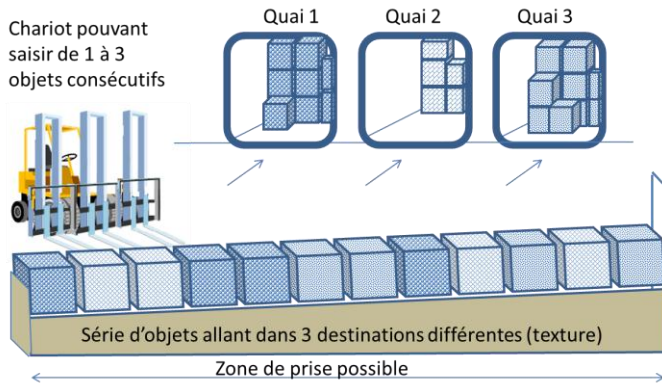


Figure 1 : schéma de l'installation étudiée

Dans un tel problème, le but n'est pas de minimiser les mouvements avec un algorithme complexe d'optimisation. On est réellement au niveau opérationnel et le cariste ne dispose pas d'un ordinateur de bord. De plus, la séquence de cubes est très rarement connue à l'avance et elle est sujette à de très nombreuses perturbations.

Clairement, le cariste utilise plusieurs heuristiques simple : la première est de prendre systématiquement les K produits consécutifs s'il peut en prendre K . Dans cet article, nous montrons que l'ensemble des solutions générées est dominant. Ensuite, il peut soit favoriser les prises maximales, soit favoriser les prises créant des sous-séquences de K éléments.

Notre objectif est de savoir quelle heuristique est la plus performante et sous quelles conditions. Pour tester la qualité de ces différentes heuristiques, il faut être capable de mieux comprendre le problème, et de créer des benchmarks dont on connaît la solution optimale. C'est l'objectif du travail présenté dans cet article.

2 REVUE DE LA LITTÉRATURE

Le problème général du chargement des camions est un problème classique que de nombreux auteurs ont étudié. Les auteurs regardent soit la disposition dans le camion [Morabito *et al.*, 2000], soit la disposition de l'entrepôt alimentant les quais [Chana et Chan, 2011], [Yüceer et Özakça, 2010]. En fait, la prise des produits elle-même ne semble pas avoir fait l'objet d'étude particulière. Le problème étudié relève plutôt du domaine des jeux. Il s'apparente à une réduction à une dimension du jeu *Clickomania*, étudié par plusieurs auteurs [Biedl *et al.*, 2002], [Schaad *et al.*, 2008], [Kendall *et al.*, 2008].

Ce jeu consiste en un tableau de N lignes et M colonnes initialement rempli $M \times N$ de cubes de couleurs différentes. Deux cubes sont dits voisins s'ils sont de la même couleur et ont un côté adjacent. La fermeture transitive de la relation de

voisinage définie des « groupes » connexes de cubes de couleurs identiques.

Le joueur doit retirer les $M \times N$ cubes. Pour retirer des cubes, il suffit de choisir un cube appartenant à un groupe d'au moins 2 éléments et tous les cubes de son groupe sont retirés. Les espaces ainsi créés sont comblés par gravité (dans chaque colonne les trous sont comblés en descendant les cubes restants). Si une colonne se vide, toutes les colonnes de droite sont décalées vers la gauche. Chaque retrait modifie donc les relations de voisinages et les cubes. La Figure 2 donne un exemple simple de déroulement du jeu gagné (plus de cubes).

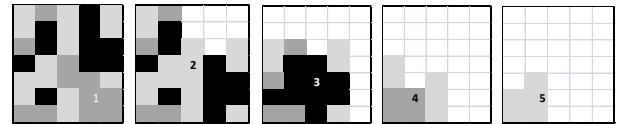


Figure 2 : exemple de jeu *Clickomania*

Dans ce jeu, il y a le problème d'existence (peut-on enlever tous les cubes), de minimisation du nombre de cube restant, d'optimisation d'un score (souvent la somme des carrés des nombres de cubes retirés à chaque étape au delà des 2 minimaux), de minimisation du nombre d'opérations. Dans l'exemple, le jeu est faisable, le nombre de coup est 5 et le score serait de :

$$(7-2)^2 + (9-2)^2 + (10-2)^2 + (4-2)^2 + (5-2)^2 = 151.$$

En ce qui concerne l'existence d'une solution, [Biedl *et al.*, 2002] ont montré que le problème à une colonne et deux couleurs peut se résoudre linéairement, le problème à 2 colonnes et 5 couleurs est NP-difficile ainsi que le problème à 5 colonnes 3 couleurs. Ils ont montré que si le problème n'est pas faisable, la minimisation du nombre de cubes restant peut aussi se résoudre en un temps polynomial. Finalement, les mêmes auteurs ont montré que pour un jeu donné, on peut décider en $O(kn^3)$ si un jeu à une colonne de n cubes et k couleurs est faisable. En revanche, les auteurs n'ont pas abordés le problème de la minimisation du nombre de coup si le jeu est faisable (ni sa complexité ni sa résolution).

Le problème général à une colonne est ouvert. Une variante de ce jeu, le *Samegame*, a été montrée NP-difficile [Schadd *et al.*, 2008].

En 2008, un état de l'art sur la complexité des puzzles [Kendall *et al.*, 2008] montre que le problème à une colonne et N couleurs reste un problème ouvert. Toutefois, contrairement aux problèmes étudiés, dans le jeu de *Clickomania* à une colonne et k couleurs, on ne peut retirer les cubes qu'avec des groupes d'au moins deux éléments et tous les cubes de même couleur partent simultanément. Donc il n'est pas possible d'étendre simplement les résultats connus.

Même si le problème étudié ne correspond pas exactement au jeu de *Clickomania*, il en est très proche et les démonstrations de complexité de ses versions les plus simples semblent conforter l'idée que notre problème de *picking* est un problème difficile. Cela explique aussi le peu de littérature dans le domaine du *picking*.

3 DESCRIPTION DU PROBLEME

On peut donc caractériser ce problème de chargement de la séquence S de N palettes dans D camions par :

- Un *Clickomania* d'une colonne avec D couleurs, où D est le nombre de destinations,
- Deux contraintes additionnelles qui sont :
 - o au maximum K cubes peuvent être retirés simultanément (problème d'existence de solutions identique, nombre de mouvements différents),
 - o seuls les M premiers cubes sont accessibles, mais on connaît la séquence complète des cubes supposée arriver.
- Une contrainte relaxée : il est possible de retirer un seul cube (ce qui fait que tous les problèmes ont une solution).

On parlera du problème noté par $C(S, N, D, K, M)$.

L'objectif n'est pas, comme dans le jeu de *Clickomania* classique, de maximiser un score (qui pousse à constituer les groupes les plus volumineux possibles, indépendamment du nombre de groupes), mais bien à minimiser le nombre de groupes.

Définition 1 : Une stratégie est un ensemble de retraits permettant de retirer les N cubes d'un *Clickomania* avec des retraits d'au plus K cubes consécutifs.

Remarque 1 : On peut représenter l'ensemble des stratégies par l'ensemble des expressions parenthésées définies sur la séquence S telle que chaque parenthèse contienne entre 1 et K éléments unitaires (donc hors parenthèse), tous de même couleur.

Exemple 1 : Soit le problème $C(ababacbcabc, 12, 3, 2, 5)$. Les expressions suivantes :

- $(a)(b(a)b)(a)(c)(b)(c(b(a)b)c)$
- $(a(b(a(b)a)(c(b)c)b)a)(b)(c)$

sont deux stratégies de résolution pour ce problème.

Corollaire : Le nombre de stratégies sur une séquence de n éléments est donc strictement inférieur au nombre de Catalan C_n .

Rappelons que le nombre de Catalan C_n correspond par définition entre autres au nombre d'expressions valides sur un ensemble de n éléments. En fait, ce nombre est une borne supérieure car de nombreuses expressions ne sont pas valides, du fait, d'une part, les objets de premier niveau dans la parenthèse doivent être similaires (même destination) et, d'autre part, leur nombre est borné. Par exemple, sur la séquence de l'exemple, l'expression $(ab)(aba)(c(b(cba)b)c)$ est une expression valide, mais ne constitue pas une stratégie car les groupes ne sont pas homogènes.

Mais la recherche d'une valeur plus précise du nombre de stratégies semble difficile à trouver. Aucune des extensions du nombre de Catalan ne semble correspondre à ce type d'énumération.

Définition 2 : Une solution est une séquence totalement ordonnée des retraits constituant une stratégie.

Remarque 2 : Chaque parenthèse (donc chaque opération de retrait) est un intervalle de la séquence S . Par construction (parenthèses valides), deux intervalles quelconques sont soit disjoints, soit l'un est inclus dans l'autre. On peut donc définir sur l'ensemble des opérations de retraits un ordre partiel au sens de l'inclusion des parties d'un ensemble.

Exemple 2 : Pour les deux stratégies, les solutions, représentées sous forme d'ordres partiels, sont illustrées par la Figure 3 et la Figure 5 (les chiffres indiquent la numérotation des parenthèses).

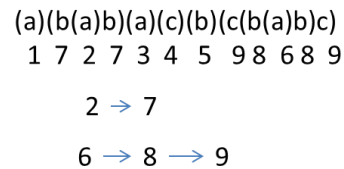


Figure 3 : solution avec 9 retraits

Pour mieux comprendre les mouvements, on peut numérotter les différents objets.

Tableau 1 : correspondance produit-rang

Objet	a	b	a	b	a	c	b	c	b	a	b	c
rang	1	2	3	4	5	6	7	8	9	10	11	12
Opération	1	7	2	7	3	4	5	9	8	6	8	9

Dans l'exemple de la Figure 3, les opérations de retraits sont donc des parties de $\{1,2,3,4,5,6,7,8,9,10,11,12\}$ et donc des segments de $[1,12]$ soit disjoints, soit inclus. La Figure 4 décrit la correspondance entre opération et partie.

opération	retiré	Partie
1	{1}	[1]
2	{3}	[3]
3	{5}	[5]
4	{6}	[6]
5	{7}	[7]
6	{10}	[10]
7	{2,4}	[2,4]
8	{9,11}	[9, 11]
9	{8,12}	[8, 12]

Figure 4 : correspondance opération-partie

Sur le même exemple, on peut créer une autre stratégie comme sur la Figure 5.

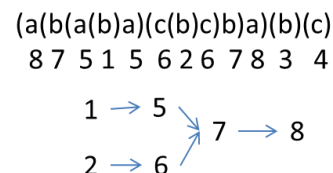


Figure 5 : solution avec seulement 8 mouvements

Dans cet exemple, le mouvement 8 encadre les mouvements 1, 2, 5 et 6 par la séquence : (8-7-5-1-5-6-2-6-7-8). Donc 8 est supérieur au sens de l'inclusion à ces mouvements.

Pour ce second exemple, la Figure 6 décrit la liste des parties et des retraits.

opération	retiré	Partie
1	{4}	[4]
2	{7}	[7]
3	{11}	[11]
4	{12}	[12]
5	{3,5}	[3,5]
6	{6,8}	[6,8]
7	{2,9}	[2,9]
8	{1,10}	[1,10]

Figure 6: correspondance partie-retrait du second exemple

Le dénombrement des solutions compatibles avec une stratégie est donc équivalent au dénombrement des ordres totaux supérieurs à un ordre partiel, qui reste un problème ouvert dans le cas général.

Remarque 3 : Dans les exemples simples considérés, les opérations représentées par des parenthèses { } et les segments [] se ressemblent. Cependant, dans le cas général, on peut avoir une opération avec K produits initialement non consécutifs comme { 1, 3, 8 } qui correspond au segment [1,8].

Définition 3 : Une solution est admissible pour un $C(N, D, K, M)$ si tous les retraits se font dans la fenêtre de largeur K .

Par convention, on peut décider que la partie droite de la séquence est visible. Pour une stratégie donnée, il peut se faire qu'aucune solution ne soit admissible si la fenêtre n'est pas assez large.

Exemple 3 : La stratégie suivante produit plusieurs solutions, mais aucune n'est faisable avec une fenêtre inférieure ou égale à 4 : (a)(b)a(b(a(c)(b(c)b)a)b)(c). En effet, on peut commencer par retirer le dernier (c) mais ensuite, le premier objet possible à dégager pour respecter la stratégie est le c en 4^{ème} position.

Proposition 1 : Pour un problème $C(\alpha * K + 1, D, K, M)$, $D \geq 3$, la largeur minimale pour garantir qu'une solution de la stratégie optimale soit admissible est $M > \alpha * (K - 1)$.

Preuve. Notons que le paramètre α est ici pour donner un nombre d'objets impairs. Construisons la séquence $(a, b, \dots, a, b, c, b_{k-1}, a_{k-1}, \dots, b_{k-1}, a_{k-1})$, les extrémités étant des a ou b selon la parité de α . La solution optimale est évidemment de faire en premier le retrait de « c » puis de faire α retraits consécutifs de K produits, en partant du centre. Il n'y a qu'une stratégie optimale, et pour qu'elle soit admissible, il faut que le « c » soit atteignable, d'où la proposition.

Par conséquent, pour avoir une garantie d'optimalité, la fenêtre de prise doit être très large.

Proposition 2 : Si à un instant t , une sous séquence de X produits consécutifs, $X \geq K$, ont la même destination, alors toutes les solutions consistant à retirer K éléments de ces X produits sont équivalentes.

Preuve : Cette propriété est évidente. Puisque les X objets sont consécutifs, tout retrait de K produits induira une séquence de $(X-K)$ produits consécutifs allant tous dans la même direction. Donc la solution finale est indépendante des produits retirés dans la prise de K produits.

Proposition 3 : Les solutions admissibles consistant à retirer dès que possible les sous séquences consécutives de K objets sont dominantes.

Preuve : Soit une stratégie telle qu'une sous séquence X de K objets consécutifs ne soient pas pris consécutivement.

$$K = x_1, X', x_k$$

Avec x_1, x_k respectivement les premiers et derniers éléments de la séquence. Les objets x_1, x_k vont être retirés dans des sous séquences que nous appellerons Y et Z , telles que :

$$Y = Y' - x_1, \text{ avec } \text{card}(Y) \leq K.$$

$$Z = x_k - Z', \text{ avec } \text{card}(Z) \leq K.$$

Par construction, et puisque la séquence X est consécutive, à un instant donné la séquence $Y - X - Z$ apparaîtra consécutive. Or, d'après la Proposition 2, les objets de cette séquence d'objets consécutifs de même destination peuvent être retirés dans n'importe quel ordre, en particulier en retirant X immédiatement. Par conséquent, les stratégies consistant à retirer toutes les séquences de K produits consécutifs sont dominantes.

Maintenant, parmi les solutions admissibles compatibles avec une stratégie donnée, il est clair que si une séquence de K produits consécutifs peut être retirée simultanément, le retrait au plus tôt est dominant car il augmente la visibilité sur les produits de la suite de la séquence. Autrement dit, les solutions consistant à retirer au plus tôt les séquences de K produits consécutifs pour la même direction sont dominantes.

Exemple 4 : Sur un exemple plus complet, il est possible de mieux comprendre l'importance des paramètres respectifs du modèle. Soit donc un problème avec 20 cubes, $(S, 20, 3, 3, 6)$ où $S = (A, B, B, C, A, C, A, B, B, C, C, A, C, B, B, A, C, B, B, A)$. La séquence se lisant de droite à gauche. La fenêtre de prise étant à droite, en gris sur la Figure 7 (donc au début, la séquence visible est (B, A, C, B, B, A)).

1	A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	B	B	A
2			A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	A
3				A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	A
4					A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	B
5						A	B	B	C	A	C	A	B	B	C	C	C	C	B	B
6							A	B	B	C	A	C	A	B	B	B	B	B	B	B
7											A	B	B	C	A	C	A	B	B	B
8													A	B	B	C	A	C	A	B
9														A	B	B	C	C	B	B
10																A	B	B	B	B
11																				A

Figure 7 : solution avec 11 mouvements au $(S, 20, 3, 3, 6)$

La solution de la Figure 7 correspond à l'expression :

$$(A),(B,B),(C,(A),C),(A),B),(B),(C,C,(A),C),B,B),(A,(C),(B,B),A).$$

Dans cette solution, la partie en jaune correspond à la zone de prise visible de l'opérateur et la zone en rouge correspond aux produits consécutifs saisis.

1	A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	B	B	A
2			A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	A
3				A	B	B	C	A	C	A	B	B	C	C	A	C	A	C	A	A
4					A	B	B	C	A	C	A	B	B	C	C	C	A	C	A	A
5						A	B	B	C	A	C	A	B	B	A	C	A	C	A	A
6									A	B	B	C	A	C	A	C	A	C	A	A
7												A	B	B	C	A	C	C	A	A
8													A	B	B	C	C	C	A	A
9																A	B	B	A	A
10																			A	A

Figure 8 : Solution à 10 mouvement pour (S, 20, 3, 3, 6)

La solution présentée à la Figure 8 donne une solution à 10 mouvements seulement pour le même problème. Elle correspond à l'expression suivante :

$$(A,(B,B),(C,(A),C,(A,(B,B),(C,C,(A),C),(B,B),A),C),(B,B),A).$$

Remarque 4 : Dans cette solution optimale, le premier et le dernier produit sont pris ensemble.

Dans cet exemple, si on considère le problème (S, 20, 3, 2, 6), donc avec une prise maximale de 2 produits consécutifs, il existe une solution avec seulement 11 mouvements (voir Figure 99).

1	A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	B	B	A	
2			A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	A	
3				A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	A	
4					A	B	B	C	A	C	A	B	B	C	C	A	C	A	C	B	
5						A	B	B	C	A	C	A	B	B	A	C	A	C	B	B	
6									A	B	B	C	A	C	A	A	C	B	B	B	
7												A	B	B	C	A	C	C	B	B	
8													A	B	B	C	A	B	B	B	
9																A	B	B	C	A	
10																			A	C	A
11																				A	A

Figure 9 : solution au (S, 20, 3, 2, 6) avec 11 mouvements

L'expression correspondante est donc comme suit :

$$(A,(B,B),(C,(A),C,(A,(B,B),(C,C,(A),C),(B,B),(A,(C),(B,B),A).$$

1	A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	B	B	A	
2			A	B	B	C	A	C	A	B	B	C	C	A	C	B	B	A	C	A	
3				A	B	B	C	A	C	A	B	B	C	C	A	C	A	C	A	A	
4					A	B	B	C	A	C	A	B	B	C	B	A	C	A	C	A	
5						A	B	B	C	A	C	A	B	B	A	C	A	C	A	A	
6									A	B	B	C	A	C	A	C	A	C	A	A	
7												A	B	B	C	A	A	A	C	A	
8																A	B	B	C	A	
9																	A	B	B	A	
10																				A	A

Figure 10 : solution à 10 mouvements au (S, 20, 3, 2, 7)

Si maintenant on étudiait le problème (S, 20, 3, 2, 7) avec une fenêtre de prise très légèrement plus large, il existe une solution avec 10 mouvements seulement, comme illustrée par la Figure 10. Par conséquent, les paramètres comme la largeur de la fenêtre et le nombre de prises simultanées ont une importance significative sur les résultats.

4 MODELE MATHEMATIQUE

Avant de procéder, introduisons d'abord les variables de décision suivantes :

- S une séquence de produits, $i=1, \dots, N$
- d_i destination du $i^{\text{ème}}$ produit dans la séquence
- M largeur de la fenêtre de prise
- K nombre maximum de produits chargés ensemble
- Maxj borne du nombre de transports (au pire, N)
- $x_{i,j}$ = 1 si le produit i part avec le transport j
- $A_{i,j}$ = 1 si un produit $< i$ est expédié dans le transport j

- $B_{i,j}$ = 1 si un produit $> i$ est expédié dans le transport j
- y_j = 1 si le transport j est utilisé

La formulation mathématique est alors comme suit. Notons que les premières inégalités constituent les contraintes du modèle alors que la dernière contrainte est la fonction objectif.

$$\forall j, \quad x_{i,j} \leq y_j \quad (1)$$

$$\forall j, k < i \quad x_{k,j} \leq A_{i,j} \quad (2)$$

$$\forall j, k > i \quad x_{k,j} \leq B_{i,j} \quad (3)$$

$$\forall i, \quad \sum_j x_{i,j} = 1 \quad (4)$$

$$\forall j, \quad \sum_i x_{i,j} \leq K \quad (5)$$

$$\forall j, i_1, i_2 / d_{i_1} \neq d_{i_2}, \quad x_{i_1,j} + x_{i_2,j} \leq 1 \quad (6)$$

$$\forall i, l \quad A_{i,l} + B_{i,l} \leq 2 - \sum_{j=l+1}^n x_{i,j} \quad (7)$$

$$\forall i, j, \quad \left(i - \sum_{a \leq i} \sum_{l \leq j} x_{a,l} \right) \leq M + (i - M) * (1 - x_{i,j}) \quad (8)$$

$$\forall j < l \quad y_l \leq y_j \quad (9)$$

$$\text{Min} \left(\sum_j y_j \right) \quad (10)$$

Les contraintes (1), (2) et (3) définissent seulement les variables indicées de A, B et y par rapport aux variables indicées de type x.

La contrainte (4) exprime que chaque produit doit partir une et une seule fois et la contrainte (5) exprime le maximum d'objets pris en un transport.

La contrainte (6) dit que deux produits n'allant pas dans la même destination ne peuvent pas être déplacés lors du même mouvement.

La contrainte (7) exprime que si un produit est expédié par le transport j , alors il n'existe aucun transport précédent j qui ait pris simultanément un produit précédent i et un produit succédant i .

La contrainte (9) est une équation permettant de vérifier qu'il n'y a jamais plus de M produits en attente de déplacement. En effet, le second terme est égal à M (donc la taille maximale) si le produit i est expédié dans le transport j . Sinon, de toute manière, la contrainte est toujours respectée. Le premier terme,

lui, correspond à tous les produits classés avant i qui n'ont pas encore été expédiés quand le transport j est effectué.

Pour terminer, la contrainte (9) permet de n'utiliser que les premiers mouvements et la fonction objectif (10) consiste à minimiser le nombre total de transports utilisés.

5 RESULTATS PRELIMINAIRES

Nous passons dans cette section à l'étude de la performance du modèle mathématique que nous venons de proposer ci-dessus. Pour ce faire, nous avons implémenté ce modèle sous *Cplex*, version 12.2. Nous avons exécuté ce programme sur un Intel Pentium (R), muni d'un processeur de 3.2GHz et d'une mémoire vive de 3.2Go.

Nous avons testé ce programme sur deux séquences de 20 et 30 produits, respectivement. Nous n'avons pas pu aller au-delà car les temps d'exécution devenaient prohibitifs, comme on peut s'en apercevoir sur les Figures 11, 12 et 13. Notons que le symbole **x** dans ces tableaux signifie que le temps d'exécution dépasse la limite de 6 heures qu'on s'est fixée. Les différentes instances sur lesquelles cette formulation a été testée ont été générées d'une manière aléatoire à partir d'une distribution uniforme. Les paramètres importants caractérisant une instance sont la taille de la séquence, le nombre de chaque produit, le nombre de destinations, et la largeur de la fenêtre de prise. Nous avons résumé les résultats préliminaires obtenus dans les tableaux suivants. Notons les abréviations suivantes :

- **fen** : largeur de la fenêtre de prise.
- **comb** : combinaison des produits dans la séquence. Par exemple, la combinaison 7-7-6 signifie qu'il y a 7 produits pour la destination 1, 7 autres produits pour la destination 2, et 6 produits pour la destination 3.
- **exple** : une instance correspondant à la combinaison traitée; notons que 3 instances sont générées pour chaque combinaison.
- **temps** : temps d'exécution (en secondes) mis pour obtenir la solution optimale.
- **obj** : solution optimale produite pour l'instance correspondante.

Fen	comb	exple	temps	obj
6	10-10	1	5.49s	8
		2	2.84s	8
		3	3.45s	8
	15-15	1	13.03s	9
		2	2.9s	8
		3	14.82s	8
8	10-10	1	16.64s	8
		2	7.69s	8
		3	9.4s	8
	15-15	1	5.71s	8
		2	4.43s	8
		3	19.49s	8
10	10-10	1	8.07s	8
		2	18.65s	8
		3	22.33s	8
	15-15	1	12s	8
		2	8.07s	8
		3	5.48s	8

Figure 11 : simulation avec 20 produits et 2 destinations

fen	comb	exple	Tps	obj
6	7-7-6	1	39.96s	10
		2	13.6s	9
		3	8.31s	9
	10-5-5	1	9.31s	9
		2	5.36s	9
		3	12.73s	9
	8-9-3	1	53.87s	10
		2	7.7s	9
		3	3.25s	8
8	7-7-6	1	294.84s	10
		2	63.96s	9
		3	27.04s	9
	10-5-5	1	35.29s	9
		2	61.9s	9
		3	18.05s	9
	8-9-3	1	337.57s	9
		2	9.41s	8
		3	5.5s	8
10	7-7-6	1	1326.84s	10
		2	930.6s	9
		3	198.45s	9
	10-5-5	1	247.69s	9
		2	135.85s	9
		3	139.83s	9
	8-9-3	1	227.68s	9
		2	15.67s	8
		3	16.18s	8

Figure 12 : simulation avec 20 produits et 3 destinations

fen	comb	exple	temps	obj
6	2-6-6-6	1	17.78s	10
		2	61.19s	11
		3	26.78s	10
	5-5-5-5	1	28.93s	10
		2	125.43s	11
		3	170.7s	11
	8-4-4-4	1	45.38s	10
		2	91.47s	10
		3	798.17s	12
8	2-6-6-6	1	501.03s	10
		2	31.96s	10
		3	113.48s	10
	5-5-5-5	1	176.93s	10
		2	1214.15s	11
		3	190.22s	10
	8-4-4-4	1	302.98s	10
		2	226.54s	9
		3	x	x
10	2-6-6-6	1	709.43s	10
		2	473.96s	10
		3	2299.17s	10
	5-5-5-5	1	1868.07s	10
		2	3510.03s	11
		3	x	x
	8-4-4-4	1	1799.01s	10
		2	292.93s	9
		3	x	x

Figure 13 : simulation avec 20 produits et 4 destinations

Concernant les tests que nous avons effectués avec 40 produits, nous avons constaté que les temps d'exécution se sont considérablement détériorés, même que pour un nombre important d'instances, nous avons dû arrêter l'exécution faute de production de solution après plus de 6 heures d'attente.

Comme nous pouvons donc le constater, la performance de cette modélisation est assez efficace pour des instances de petites tailles. Mais, dès que ces tailles deviennent modérées, les temps d'exécution explosent et deviennent prohibitifs.

6 CONCLUSION

Ce problème de *picking a priori* simple à décrire s'avère plus complexe qu'il n'y paraît. Partiellement relié à un jeu dont la complexité est toujours recherchée, mais reste un problème ouvert, le dénombrement même du nombre de stratégies possibles et de solutions faisables semble faire appel, d'une part, à un sous-ensemble des nombres de Catalan et, d'autre part, au dénombrement d'ordres totaux supérieurs à un ordre partiel donné. Il est donc au confluent de plusieurs domaines de recherche, mais souvent dans un contexte de problèmes ouverts et non encore résolus.

Nous avons proposé quelques définitions et montré que la politique consistant à retirer au plus tôt les groupes de taille maximale est dominante. Nous avons donné quelques bornes sur le nombre de mouvements le chariot aura à effectuer.

Nous avons aussi proposé une modélisation mathématique, qui permet de résoudre à l'optimum des instances de problèmes de petite taille, comme le montrent les résultats expérimentaux de la Section 5, et nous aidera ensuite à mieux calibrer les approches heuristiques.

Le but de la recherche n'est pas de résoudre à l'optimum les problèmes réels, mais de tester des heuristiques que les opérateurs pourront appliquer sur le terrain. Pour cela, nous avons besoin d'un modèle mathématique permettant de créer des benchmarks qui seront indispensables dans le futur à l'évaluation des heuristiques.

7 REMERCIEMENTS

Tous nos remerciements à Alain Hertz, professeur au département de mathématiques et de génie industriel de l'École Polytechnique de Montréal qui nous a pointé l'idée de regarder les liens entre le problème de *picking* et jeu du *Samegame*, point de départ important dans la compréhension de notre problématique.

Cette étude est financée par le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRNG).

8 REFERENCES

- Baptiste, P., Alsene, E., Gaudimier, R. Integration of production and shipping planning: a co-operative approach, *Production Planning & Control*. Vol. 19, no. 7, pp. 645-654. Oct. 2008.
- Maknoon M.Y.; Baptiste P., Cross-docking: increasing platform efficiency by sequencing incoming and outgoing semi-trailers, *International Journal of Logistics*, Volume 12, Number 4, pp. 249-261, August 2009.
- Rim L., Gülgün A., Baptiste P., Penz B. Scheduling cross docking operations under full, partial and no information on inbound arrivals, *Computers & Operations Research* Volume 38, Issue 6, pp. 889-900, June 2011.

Morabito R., Morales S. R., Widmer J. A. Loading optimization of palletized products on trucks, *Transportation Research Part E: Logistics and Transportation Review*, Volume 36, Issue 4, pp. 285-296, December 2000.

Chana F. T.S and Chan H.K. Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage, *Expert Systems with Applications*, Volume 38, Issue 3, pp. 2686-2700, March 2011.

Yüceer Ü, Özakça A. A truck loading problem, *Computers & Industrial Engineering*, Vol. 58, pp. 766-773, 2011.

Biedl, T., Demaine, E., Demaine, M., Fleischer, R., Jacobsen, L., and Munro, J. The Complexity of Clickomania. *More Games of No Chance* (ed. R. Nowakowski), pp. 389-404, Cambridge University Press, 2002.

Schadd, M. P. D., Winands, M. H. M., van den Herik, H. J., and Aldewereld, H. Addressing NP-Complete Puzzles with Monte-Carlo Methods. *AISB 2008 Convention: Communication, Interaction and Social Intelligence*.

Kendall G., Parkes A., Spoerer K., A survey of NP complete Puzzle, *ICGA Journal*, pp. 13-35, March 2008.

Demain E. D., *Playing Games with Algorithms: Algorithmic Combinatorial Game Theory*, J. Sgall, A. Pultr, and P. Kolman (Eds.): MFCS 2001, LNCS 2136, 2-1, pp. 18-33.