

CONCEPTION ET OPTIMISATION DU MAPPING-PRODUIT D'UN SYSTÈME DE DISTRIBUTION AUTOMATISÉ

Olivier DEVISE, Christophe CAUX, Frédéric LECHA

IFMA - LIMOS UMR CNRS 6158

Campus des Cézeaux, B.P. 265

63175 Aubière Cedex

Olivier.Devise@ifma.fr, Christophe.Caux@ifma.fr

RÉSUMÉ : *L'objectif de cet article est de présenter une méthode pour générer et optimiser un mapping-produit performant d'un système de distribution comme par exemple, un système de préparation de commande. Le mapping-produit consiste à définir l'affectation des produits dans les différentes gares du système de distribution, sous un certain nombre de contraintes (type de support de stockage, contraintes capacitaires, le foisonnement, ...). Le mapping-produit influence notablement non seulement la performance du système de distribution en termes de productivité, d'ergonomie et de qualité. Cet article présente une nouvelle méthode de conception puis d'optimisation du mapping-produit et illustre ces méthodes sur un exemple concret.*

MOTS-CLÉS : *système de distribution, mapping-produit, optimisation d'un système de distribution, recuit simulé*

1. INTRODUCTION

Au cours des dernières années, dans une politique de minimisation du délai client, des systèmes logistiques automatisés de plus en plus performants font leur apparition. En effet, avec l'avènement des nouvelles technologies de communication et l'apparition de nouvelles entreprises basées sur ce moyen de communication, le délai joue un rôle essentiel pour conserver ses parts de marché. Maintenant, avec les évolutions de l'informatique et de l'automatisme, ces systèmes ont trouvé un nouvel essor, en particulier dans le domaine des installations de préparation de commandes. Dans ce type d'installations, la localisation des articles dans le système (le mapping-produit) est un problème complexe et omniprésent. En effet, plusieurs contraintes apparaissent lors de cette phase : la configuration physique des gares (nombre limité d'emplacements de chaque type de support), les règles de sécurité (on ne peut pas mettre des produits d'entretien avec de la nourriture) ...

Dans cette étude, nous présentons une nouvelle méthode débouchant sur un outil informatique permettant d'automatiser le mapping-produit aussi bien lors de la conception d'une installation que lors d'un remappage complet d'un système existant. À partir d'une installation logistique définie, d'une base article correctement renseignée et d'un historique des commandes détaillé, nous allons placer les articles dans les différents supports de stockage de la gare en optimisant certains critères comme la minimisation du nombre d'arrêts en gare tout en respectant certaines contraintes comme l'obligation de placer certains articles dans des gares déterminées.

2. ENJEUX, DÉFINITIONS ET OBJECTIFS

2.1 Enjeux

Actuellement, le mapping-produit est trop souvent réalisé "à la main" rendant cette tâche longue et fastidieuse. En effet, pour une installation de taille moyenne (40 à 50 gares et de 5 000 à 10 000 produits), la réalisation du mapping-produit nécessite environ 240 heures de travail car sa réalisation requiert une étude précise de la base article et de l'historique des commandes. Or, le mapping-produit a un rôle important non seulement en ce qui concerne les performances du système logistique (comme par exemple le temps moyen ou le temps maximum pour réaliser une commande, ou le nombre de commandes traitées par poste de 8 heures) mais aussi sur le plan de l'ergonomie du poste de travail. En outre, dans le suivi d'une politique "zéro défaut", le mapping-produit joue un rôle essentiel en évitant la permutation d'article dans les commandes clients.

Nous allons commencer par définir un certain nombre de termes qui seront utilisés tout au long de cet article, dont en particulier, le mapping-produit.

2.2 Définitions et vocabulaire

Nous rappelons ici un certain nombre de termes et de spécificités des systèmes de distribution :

- *Installation de préparation de commandes :* une installation de préparation de commandes est un système logistique dans lequel des contenants identifiés (cartons ou bacs) se déplacent sur des convoyeurs automatisés. Ces contenants s'arrêtent dans des gares où

les opérateurs les remplissent avec les articles commandés. En sortie, les cartons regroupant l'ensemble de la commande sont livrés au client.

- *Gare* : c'est le lieu physique de prélèvement des articles dans des supports de stockage par des opérateurs.
- *Supports de stockage, emplacements* : c'est le système physique permettant de stocker les produits. On retrouve des supports de types différents: stockage dynamique (exemple des condiments dans les supermarchés), des palettes, stockage statique (étagères). Pour chaque article, un type de support est associé en fonction du volume du produit et de la fréquence de prélèvement de ce dernier. Les palettes peuvent être implantées de deux manières principales : en îlot ou en frontal. Une implantation en îlot permet d'avoir plus de palettes dans la gare, mais elle a un rendement de picking plus faible qu'une implantation en frontal.
- *Le mapping-produit* : c'est le fait d'associer une référence à un emplacement précis sur l'installation : définir le numéro de gare et numéro de support où se situent les articles d'une référence. Le mapping-produit doit respecter certaines contraintes qui seront explicitées par la suite.
- *La fragilité* : un produit A est dit fragile lorsqu'il n'est pas possible de placer un article B sur le produit A sans l'endommager. Dans une installation logistique, on essaye de placer les produits fragiles en fin de ligne.
- *Le foisonnement* : un produit est dit foisonnable si sa forme géométrique lui permet de recevoir un autre article (exemple des corbeilles à papier dans laquelle on peut mettre une boîte de stylos).
- *Les contrôles pondéraux* : ils permettent d'éviter les erreurs de picking. Le principe revient à peser le colis à divers endroits de l'installation et de le comparer à la somme des poids des articles qui doivent être présents dans le colis à l'endroit du contrôle. Si un écart de poids est détecté, le colis est rejeté et envoyé dans une zone manuelle où il sera contrôlé et modifié.
- *Les produits à poids variable* : ce sont des produits qui ont un poids qui est défini avec une tolérance assez large. Par exemple, un sac de pommes de terre est calibré à $5\text{kg} \pm 100\text{g}$. On peut donc avoir un sac de pommes de terre de 4,9 kg ou 5,1 kg. D'autres articles ont leur poids en fonction de paramètres extérieurs comme le papier qui est plus lourd lorsque l'air est humide.
- *La longueur de façade d'un article* : elle correspond à la longueur de l'objet que l'on retrouve face à l'opérateur. On utilise seulement cette longueur pour des stockages dynamiques ou statiques. A partir de la longueur d'étagère disponible, la longueur de façade nous permet de savoir le nombre de produits que l'étagère va pouvoir accueillir.

2.3 L'objectif visé, problématique général

À partir d'une installation logistique définie, d'une base article correctement renseignée et d'un historique des commandes détaillé, nous devons définir une méthode et un outil pour placer les articles dans les différents supports de stockage de la gare. Il s'agit donc de définir une association entre articles et emplacements disponibles dans les gares. Cette association devra suivre des contraintes physiques comme le respect de la configuration physique des gares (nombre d'emplacements dynamiques, nombre d'emplacements palettes en frontal, nombre d'emplacements palettes en îlot, longueur en stock dynamique, longueur en stock statique, ...), le respect de la capacité des gares en termes de ligne de commande par unité de temps (nombre d'opérateurs limité dû à une surface réduite au niveau de la gare) et sous des contraintes de gestion comme :

- certains articles sont obligatoirement dans une gare déterminée ;
- certains produits ne doivent absolument pas être dans la même gare que d'autres produits ;
- certains produits doivent obligatoirement être dans la même gare que d'autres produits ;
- certains produits ne doivent absolument pas être dans une gare déterminée ;
- les produits foisonnables doivent prioritairement être placés en début de ligne ;
- le placement des produits à poids variable dépend de la position des contrôles pondéraux ;
- on ne peut retrouver les produits que dans un nombre limité de gares (généralement une seule).

Une analyse des systèmes existants montre que le temps de préparation d'une commande est composé du temps de préparation des colis dans les gares, des temps de transfert entre les gares et de temps d'attente dans les gares (Lecha, 2002). Or, ce temps d'attente, d'après les constructeurs - concepteurs de ces systèmes représente généralement 90 % du temps total.

2.4 Les données disponibles

Les données disponibles décrivent les produits (à partir de la base article), l'opération de picking ou la productivité d'un opérateur suivant le type d'opération de picking, la représentation physique des gares et du circuit des convoyeurs. Le tableau 1 présente ces données.

3. PROBLEMES SIMILAIRES

Ce type de systèmes de distribution est apparu assez récemment et ne se trouve encore que très peu dans la littérature. En effet, celle-ci traite souvent d'un autre problème : de nombreux écrits concernent le Generalized Assignment Problem (GAP), le Stock Location Assignment Problem (SLAP), le Cellular Manufacturing (CM) qui sont des problèmes similaires par certains aspects.

Type de données	Données prises en comptes
Caractéristiques des produits (issues de la base article)	<ul style="list-style-type: none"> - Référence, - Désignation, - Poids, - Longueur, - Largeur, - Hauteur, - type de support utilisé (palettes en frontal, en îlot, statique, dynamique), - nombre de supports nécessaires (fonction du réapprovisionnement), - longueur de façade des produits, - indice de foisonnement (savoir si un produit est foisonnable ou non), - indice de variation de poids (savoir si un produit a une forte variation de poids ou non), - nombre de gares maximum où on peut retrouver la référence, - volume consommé par unité de temps.
Opérateurs	<ul style="list-style-type: none"> - rendement de picking selon le type de support (nombre de lignes de commandes par unité de temps), - temps de déplacement d'un opérateur pour aller d'une gare à une autre.
Caractéristiques du circuit	<ul style="list-style-type: none"> - dimensions, - position des gares, - vitesse des convoyeurs, - configuration (en U, en ligne, position des "shunts",...), - position des contrôles pondéraux.
Caractéristiques de chaque gare	<ul style="list-style-type: none"> - numéro de gare, - numéro de boucle où est située la gare, - nombre d'emplacements de chaque type, - capacité de la gare en nombre de lignes par unité de temps, - temps d'attente moyen dans la gare g.
Historique des commandes	<ul style="list-style-type: none"> - numéro de commande, - référence article, - quantité, - date.
Ensemble des cartons traités (après colisage volumique)	<ul style="list-style-type: none"> - quantité de carton, - référence, - quantité de chaque référence.

Tableau 1. données disponibles

3.1 Littérature des systèmes de distribution

Dans la littérature du domaine de la logistique (Van Den Berg et Zijm, 1999), les auteurs distinguent les différents entrepôts logistiques existants et proposent des modèles des problèmes les plus fréquemment traités dans ce type d'installation comme la réduction du stock, l'affectation des zones de stockage, l'affectation des articles dans les zones de stockage, ... Cependant, les modèles proposés sont utilisés pour des entrepôts manuels et non pas des systèmes logistiques automatisés. Ils concernent en effet principalement le Traveling Salesman Problem (TSP) ou comment hiérarchiser les ordres de prélèvement afin de minimiser la distance parcourue par les opérateurs (Ek et Loon, 1999, Roodbergen et de Koster, 2001), la stratégie de prélèvement des produits (Lin et Lu, 1999) ("batching", "zoning", ...).

Traditionnellement, dans les entrepôts manuels, l'opérateur s'occupe d'une et une seule commande à la fois. Dans un fonctionnement en "batching", les articles de plusieurs commandes sont prélevés ensembles puis

redistribués. L'avantage du "batching" est que l'on minimise la distance totale parcourue par l'opérateur. Par contre, la commande nécessite plus de temps pour être réalisée. En "zoning", la commande principale est découpée en sous-commandes qui sont envoyées dans différentes zones de l'entrepôt. L'avantage d'une telle stratégie de picking permet de traiter une commande plus rapidement car les sous-commandes sont traitées simultanément. Cependant, cette méthode nécessite une bonne synchronisation de toutes les zones pour être efficace.

Une installation de préparation de commandes pourrait donc être équivalente en terme de fonctionnement à un entrepôt traditionnel géré en "zoning" et où les sous-commandes sont traitées consécutivement.

Au niveau de l'affectation de stock, le SLAP semble pouvoir apporter des réponses. Dans (Hesket, 1963), l'auteur propose une méthode afin de placer les produits dans un entrepôt manuel. Il définit un coefficient nommé COI qui est le ratio entre l'espace de stockage requis et la fréquence de prélèvement de l'article. Ensuite, il place

les produits avec le plus petit COI près du point d'entrée/sortie de l'entrepôt.

On voit donc que cette méthode d'affectation des produits au stock ne peut pas être utilisée dans notre cas. En effet, nous nous intéressons principalement aux relations entre les produits afin de déterminer les références qui doivent être dans les mêmes gares. Or dans ce dernier article, l'auteur n'a pas besoin de traiter ces aspects car il se trouve dans le cas d'un entrepôt manuel en fonctionnement traditionnel (un opérateur traite une et une seule commande à la fois).

Les auteurs (Brynzér et Johansson, 1996), traitent le SLAP avec une approche différente. Leur but est de faciliter la lecture des feuilles de commandes fournies à l'opérateur grâce à une organisation plus logique des produits. Ils regroupent les produits ayant les mêmes caractéristiques ou propriétés (par exemple les pièces utilisées dans un turbo de voiture doivent être ensemble) et placent ces familles dans l'atelier en mettant les groupes qui génèrent le plus de prélèvement, près du point d'entrée/sortie de l'entrepôt.

Par contre, lorsqu'on prend le problème de conception d'îlots de production, on retrouve certaines similitudes. On peut donc se poser la question : affecter des produits dans des gares, ne serait-il pas équivalent à affecter des machines dans des îlots ?

3.2 Analogie avec les îlots de production

Dans cette section, nous allons essayer de faire une analogie entre les îlots de production et le problème de mapping-produit dans une installation de préparation de commandes.

Îlot	Mapping
Pièces	Colis
Machines	Produits
Îlots	Gares
Gammes	Commandes

Tableau 2. Analogie îlots de production, mapping-produit

On peut voir de nombreuses analogies entre ces deux problèmes. Cependant le mapping-produit garde certaines particularités :

- Dans le cas de la constitution d'îlots de production, les pièces ont des gammes ordonnées mais dans le cas du mapping-produit, cette contrainte d'ordre n'intervient pas.
- Dans la majorité des cas, chaque colis a sa propre commande et on a très peu de répétitions. Il est très rare de trouver deux colis avec exactement les mêmes produits.

En termes de contraintes, les deux problèmes sont assez similaires. Dans (Plaquin, 1998), l'auteur prend en compte des contraintes identiques : capacité, certaines machines ne peuvent pas être déplacées, certaines machines doivent absolument être ensemble, ...

Cependant, les ordres de grandeur ne sont pas les mêmes :

- Le nombre de colis : en effet, dans notre cas, on aura un nombre de colis important et dépendant de l'historique fourni et de l'horizon souhaité. Par exemple, pour l'installation que nous avons testée, il y a environ 5 000 colis par jour. On se retrouve donc rapidement avec de très nombreux colis à gérer. Si on compare avec les îlots de production, on ne trouve que rarement des installations gérant plusieurs dizaines voire centaines de milliers de pièces.
- Le nombre de produits : dans une installation de préparation de commandes, un grand nombre d'articles est traité. On peut rapidement se retrouver avec 5 000 produits pour des installations moyennes. Par analogie avec les îlots de production, cela reviendrait à traiter 5 000 machines ce qui correspond à de grandes installations industrielles.

On se rend donc vite compte que les ordres de grandeur sont beaucoup plus grands pour les installations de préparation de commandes que pour les îlots de production. La plupart du temps, les heuristiques pour la constitution d'îlots de production sont utilisées pour des systèmes de taille plus réduite. De plus, les fonctions objectifs de ces deux problèmes sont différentes.

Néanmoins, l'analogie entre ces deux problèmes semble intéressante et nous a amené à étudier les méthodes de constitution d'îlots de production qui sont en abondance dans la littérature pour tenter de les appliquer à notre problème. Les principales méthodes utilisées pour le regroupement de machines en îlots sont les suivantes :

- La méthode Tabu (Adenso-Diaz, Lozano, Racero et Guerrero, 2001, Aljaber, Baek et Chen, 1997, Logendran et Seong Ko, 1997, Sun, Lin et Batta, 1995).
- Les algorithmes évolutionnistes (De Lit, Falkenauer et Delchambre, 2000, Hwang et Sun, 1996, Onwubolu et Mutingi, 2001, Plaquin, 1998).
- Le recuit simulé (Abdelmola et Taboun, 1999, Caux, Bruniaux et Pierreval, 2000).
- L'analyse des flux de production (Chan, Lam et Lee, 1999).
- Le Branch and Bound (Spiliopoulos et Sofianopoulou, 1998).

Nous nous sommes inspiré de ces méthodes pour modéliser puis résoudre le problème de mapping-produit.

4. MODÉLISATION

4.1 Hypothèses

La complexité du problème nous a conduit à simplifier notre étude.

- Pas d'optimisation directe au niveau du temps car dans la majorité des cas, le temps de transport est négligeable devant le temps d'attente dans les gares de picking. Pour cette raison, nous allons seulement nous focaliser sur la minimisation du nombre d'arrêts en gare et la minimisation du nombre de passages dans les boucles. En effet, un colis met environ 7 min. pour parcourir le circuit complet en ne s'arrêtant pas dans les gares. Le temps minimum mis par un colis qui fait des arrêts est de plus de deux heures en raison des temps d'attente aux gares.
- Au niveau des supports dynamiques cartons et statiques, nous avons décidé de gérer une longueur linéaire de support et de ne pas gérer les étages. En effet, compte-tenu de la nature des produits, la variation du temps de picking n'est pas significative entre les différents employés et les différentes hauteurs d'étagères.
- On ne tiendra pas compte des notions de foisonnement et de fragilité afin de seulement se focaliser sur les performances de rendement du système.
- Pour ces mêmes raisons, on ne prendra pas en compte les problèmes posés par le positionnement des contrôles pondéraux.
- Pas de notion d'éclatement des produits : un article se retrouve dans une et une seule gare. En effet, le fait de mettre la même référence produit dans différentes gares crée de nombreux problèmes comme la gestion informatique ou le réapprovisionnement : le nombre de mission de réapprovisionnement par jour est plafonné.
- Pas de prise en compte des problèmes d'ergonomie. La diversité des articles que nous aurons dans chaque gare nous permettra de réaliser un placement ergonomique. Nous préférons plutôt nous focaliser sur les performances du système.

4.2 Les notations

Les données nécessaires à la résolution du problème peuvent être notées de la manière suivante :

- I : l'ensemble des articles à placer dans l'installation.
- C : l'ensemble des cartons traités sur une période donnée.
- G : l'ensemble des gares du système.
- K : l'ensemble des boucles du système.
- $ref(i)$: référence de l'article i .
- $a[i][c] = 1$ si le produit i est dans le colis c , 0 sinon.
- $suppDP[i] = 1$ si l'article i est sur support dynamique palette, 0 sinon.
- $suppPS[i] = 1$ si l'article i est sur support palette en frontal, 0 sinon.

- $suppPI[i] = 1$ si l'article i est sur support palette îlot, 0 sinon.
- $suppDC[i] = 1$ si l'article i est sur support dynamique carton, 0 sinon.
- $suppS[i] = 1$ si l'article i est sur support statique, 0 sinon.
- $nbsupp(i)$: nombre de supports nécessaires pour stocker l'ensemble des articles i (fonction de la fréquence de réapprovisionnement souhaitée. Plus le stock est important, plus la fréquence de réapprovisionnement est faible).
- $longfac(i)$: longueur de façade de l'article i .
- $v(i)$: nombre de gares où on peut retrouver la référence i .
- $vol(i)$: volume d'articles i consommé par unité de temps.
- RDP : rendement de picking pour un support de type dynamique palette.
- RPS : rendement de picking pour un support de type palette au sol.
- RPI : rendement de picking pour un support de type palette en îlot.
- RDC : rendement de picking pour un support de type dynamique carton.
- RS : rendement de picking pour un support de type statique.
- $b[g, k] = 1$ si la gare g est dans la boucle k , 0 sinon.
- $NDP(g)$: nombre d'emplacements en dynamique palette dans la gare g .
- $NPS(g)$: nombre d'emplacements en palette en frontal dans la gare g .
- $NPI(g)$: nombre d'emplacements en palette en îlot dans la gare g .
- $LDC(g)$: longueur de dynamique carton dans la gare g (hypothèse simplificatrice).
- $LS(g)$: longueur de statique dans la gare g (hypothèse simplificatrice).
- $Cap(g)$: capacité de la gare g en nombre de lignes de commande par unité de temps.

Attention : un produit ne peut être que sur un seul type de support, on a donc $suppDP[i] + suppPS[i] + suppDC[i] + suppS[i] = 1$ pour tout i dans I .

Les variables de décision sont alors $x[i, g] = 1$ si l'article i est dans la gare g , 0 sinon

Le choix de la modélisation s'est porté vers des variables de décision binaires afin d'avoir des formulations de contraintes simplifiées et facilement compréhensibles.

4.3 Fonction objectif

Nous avons construit deux fonctions objectifs. La première minimise le nombre d'arrêts en gare et la seconde minimise le nombre de passage dans les boucles.

4.3.1 Minimiser le nombre d'arrêts en gare

Définissons la fonction à minimiser, c'est à dire la fonction calculant le nombre d'arrêts en gares. Soit $u[c, g]$ le nombre de références que l'on a placé dans le colis c au niveau de la gare g . On a donc

$$u[c, g] = \sum_{i \in I} a[c, i] \cdot x[i, g] \quad (1)$$

On en déduit donc la fonction d'évaluation suivante qui correspond au nombre d'arrêts en gare.

$$\min_{x[i, g]} \left(\sum_{g \in G} \sum_{c \in C} \frac{u[c, g]}{\delta(u[c, g]) + u[c, g]} \right) \quad (2)$$

avec $\delta(y) = 1$ si $y=0$ et 0 sinon.

4.3.2 Minimiser le nombre de passage dans les boucles

Bien souvent, les installations sont composées de différentes boucles au sein desquelles on retrouve des gares (voir figure 1). Dans chaque boucle, nous avons un rendement limité, par exemple 500 colis par heure. Si les colis passent par les 2 boucles, le rendement de l'installation sera de 500 colis par heure. Par contre, si nous arrivons à faire passer la moitié des colis par une seule boucle et l'autre moitié des colis par l'autre boucle, nous pouvons arriver à atteindre un rendement de 1000 colis par jour. On peut donc constater que cette fonction objectif peut aussi jouer un rôle très important pour l'optimisation des systèmes logistiques.

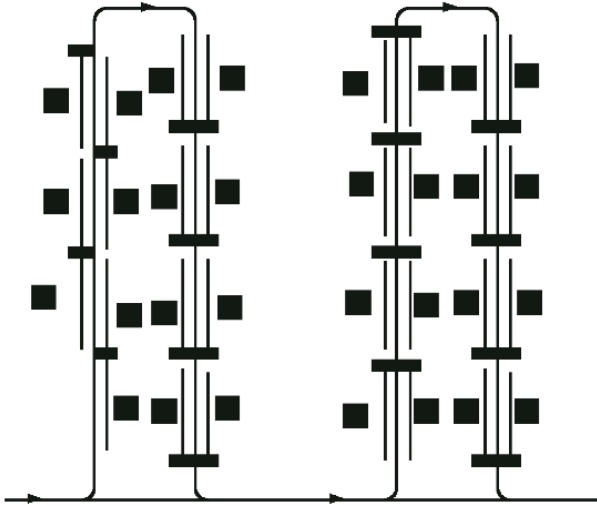


Figure 1. Installation de préparation de commande avec 31 gares et 2 boucles

On en déduit la fonction d'évaluation suivante, qui correspond au nombre de passages dans les boucles. Soit $v[c, k]$ le nombre de références que l'on a placé dans le colis c au niveau de la boucle k . On a donc

$$v[c, k] = \sum_{j \in G} \sum_{i \in I} a[c, i] \cdot x[i, j] \cdot b[j, k] \quad (3)$$

On obtient donc la fonction objectif suivant

$$\min_{x[i, g]} \left(\sum_{k \in K} \sum_{c \in C} \frac{v[c, k]}{\delta(v[c, k]) + v[c, k]} \right) \quad (4)$$

4.4 Expression des contraintes

Chaque référence doit être présente dans une et une seule gare.

$$\forall i \in I, \sum_{g \in G} x[i, g] = 1$$

nombre d'emplacements dynamiques palettes limité

$$\forall g \in G, \sum_{i \in I} \text{suppDP}(i) \cdot \text{nbsupp}(i) \cdot x[i, g] \leq \text{NDP}(g)$$

nombre d'emplacements palettes en frontal limité

$$\forall g \in G, \sum_{i \in I} \text{suppPS}(i) \cdot \text{nbsupp}(i) \cdot x[i, g] \leq \text{NPS}(g)$$

nombre d'emplacements palettes en îlot limité

$$\forall g \in G, \sum_{i \in I} \text{suppPI}(i) \cdot \text{nbsupp}(i) \cdot x[i, g] \leq \text{NPI}(g)$$

longueur en stock dynamique limitée

$$\forall g \in G, \sum_{i \in I} \text{suppDC}(i) \cdot \text{longfac}(i) \cdot x[i, g] \leq \text{LDC}(g)$$

longueur en stock statique limitée

$$\forall g \in G, \sum_{i \in I} \text{suppS}(i) \cdot \text{longfac}(i) \cdot x[i, g] \leq \text{LS}(g)$$

Certains articles sont obligatoirement dans une gare déterminée (cas des produits "tentants"...). Ici, l'article i est dans la gare g : $x[i, g] = 1$.

Le produit i ne doit absolument pas être dans la même gare que le produit j (cas de la nourriture avec les produits d'entretien), $\forall g \in G, x[i, g] \neq x[j, g]$.

Les produits i et j doivent obligatoirement être dans la même gare (cas d'une organisation par marque ou par produit), $\forall g \in G, x[i, g] = x[j, g]$.

Le produit i ne doit absolument pas être dans une gare g déterminée (cas de produits chimiques qui ne doivent pas être trop près d'une installation électrique) $x[i, g] = 0$.

Respecter la capacité des gares en termes de lignes de commande par unité de temps

$$\forall g \in G, \sum_{i \in I} \{ \text{suppDP}(i) \cdot \text{RDP} + \text{suppPS}(i) \cdot \text{RPS} \\ + \text{SuppPI}(i) \cdot \text{RPI} + \text{suppDC}(i) \cdot \text{RDC} \\ + \text{suppS}(i) \cdot \text{RSD} \} \cdot x[i, g] \cdot \text{vol}(i) \leq \text{Cap}(g)$$

4.5 Heuristique de construction du mapping initial

Nous avons décidé de développer une première heuristique simple permettant de concevoir un mapping-produit de départ qui pourra être optimisé par la suite. Le principe général de cet algorithme est de créer une matrice **Com** de taille *produits* x *produits* et de la remplir de manière à ce que *Com[i, j]* représente le nombre de fois où les produits *i* et *j* ont été commandés simultanément.

Ainsi, plus on va grouper les produits avec le plus gros coefficient *Com[i, j]* plus on aura de chance d'obtenir une "bonne" solution. Cependant, afin d'éviter les contraintes engendrées par les produits qui doivent être ensemble dans la même gare, on traite en premier lieu ces articles.

Ainsi l'algorithme se décompose de la façon suivante.

Recherche des produits qui ne doivent absolument pas être dans une gare déterminée

```
Pour chaque produit du groupe,
    Vérifier si le produit doit respecter une
    contrainte de type "produit interdit dans
    une gare déterminée"
    Si le produit doit respecter une contrainte
    de ce type,
        Mettre le numéro de gare correspondante
        dans la liste des gares interdites pour
        le groupe.
    fin Si
fin Pour
```

Recherche des produits qui ne doivent absolument pas être ensemble

```
Pour chaque produit du groupe,
    Pour chaque produit déjà placé dans les
    gares,
        Vérifier si le produit du groupe respecte
        les contraintes du type "produits pas
        ensemble" avec les produits déjà placés.
        Si le produit doit respecter une
        contrainte de ce type,
            Ajouter dans les gares interdites le
            numéro de gare des produits déjà
            placés et qui ne doivent pas être
            ensemble.
        fin Si
    fin Pour
fin Pour
```

Placement du groupe de produits

```
Pour chaque gare
    Si la gare ne fait pas partie de la liste
    des gares interdites.
        Si le groupe respecte les contraintes de
        limitation capacitaire (temps de picking
        et nombre de supports)
            Placer les produits du groupe dans la
            gare.
        fin Si
    fin Si
```

fin Pour

Construction de la matrice Com[i, j]

```
Pour chaque colis de l'historique
    Pour chaque combinaison de produit (i, j) du
    colis,
        Ajouter 1 au coefficient de la ligne i et
        colonne j de la matrice Com (i) (j).
    fin Pour
fin Pour
```

Placement des produits

```
Pour H allant de 1 à P (paramètre défini par
l'utilisateur)
    Rechercher le plus gros coefficient de la
    matriceCom(i) (j).
    Soit maxi le numero de ligne de la matrice
    Com (i) (j) pour le plus gros coefficient.
    Soit maxj le numero de colonne de la matrice
    Com (i) (j) pour le plus gros coefficient.
    Mettre à 0 Com (maxi) (maxj).
    Si les produits maxi et maxj respectent bien
    les contraintes de type "produits pas
    ensemble".
        Si les produits maxi et maxj ne sont pas
        affectés à une gare.
            Selon le même principe que
            précédemment, générer la liste des
            gares interdites.
            Selon le même principe que
            précédemment, compléter la liste des
            gares interdites.
            Selon le même principe que
            précédemment, placer le produit dans
            une gare.
        fin Si
        Si maxi (ou maxj) est déjà affecté à la
        gare g.
            Selon le même principe que
            précédemment, générer la liste des
            gares interdites.
            Selon le même principe que
            précédemment, compléter la liste des
            gares interdites.
            Si la gare g n'est pas interdite pour
            le produit maxi (ou maxj) et qu'elle
            peut accueillir maxj (ou maxi),
                Placer le produit maxj (ou maxi)
                dans la gare g
            fin Si
            Incrémenter H
        fin Si
    fin Pour
```

```
fin Si
fin Pour
Pour chaque produit non affecté à une gare,
    Selon le même principe que précédemment,
    générer la liste des gares interdites.
    Selon le même principe que précédemment,
    compléter la liste des gares interdites.
    Selon le même principe que précédemment,
    placer le produit dans une gare.
fin Pour
```

L'étape suivante consiste en l'optimisation de la solution obtenue après application de cette heuristique.

4.6 Optimisation par le recuit simulé

Le recuit simulé est une métaheuristique dont la mise en œuvre nécessite la définition de différents points : le système de voisinage, la fonction d'évaluation, la fonction de décroissance de la température, la température initiale.

4.6.1 Le système de voisinage

Pour utiliser le recuit simulé, un des aspects primordiaux est la connexité de la recherche de voisins. Nous allons donc exposer les quatre méthodes utilisées afin de générer un voisin : le déplacement, la permutation de 2 produits, la permutation d'un produit avec un groupe de produits, la permutation de produits vérifiant la contrainte.

Dans ces quatre méthodes, dans un premier temps, on ne prend en compte que les contraintes capacitaires pour générer les voisins. Ce n'est que par la suite que l'on vérifie si les contraintes de type "produits ensemble", "produits pas ensemble", "produits pas dans gare déterminée" sont vérifiées. Si une des 3 dernières contraintes n'est pas respectée, on retire un nouveau voisin à partir de la solution précédente jusqu'à obtenir un voisin valide.

- *Le déplacement* : il consiste à changer un produit de gare. Pour chaque déplacement, nous recherchons les gares aptes à recevoir le produit à déplacer. Pour cela, on ne prend en compte que les contraintes capacitaires (temps de picking et nombre de supports).
- *La permutation de deux produits* : elle consiste à passer le produit i dans la gare du produit j et inversement. Toujours selon le même principe, on ne prend en considération que les contraintes capacitaires dans cette partie.
- *La permutation d'un produit avec un groupe de produit* : il reprend principalement le cas de produits rares nécessitant plusieurs emplacements pour être stocké. En effet, si on ne prend en compte qu'une permutation de 2 produits, il est quasiment impossible que le produit puisse permuter de gare avec un autre produit de part le grand nombre d'emplacements qui lui sont nécessaires. Toujours selon le même principe, on ne prend en considération que les contraintes capacitaires dans cette partie car les contraintes de gestion ne sont testées que par la suite.
- *La permutation des produits qui doivent être ensemble avec un groupe de produits* : toujours pour des problèmes de connexité, nous devons prendre en considération les produits qui doivent respecter la contrainte de type "produits qui doivent être ensemble". Cela revient à permuter un groupe de "produits qui doivent être ensemble" avec un groupement de produits d'une autre gare. A la différence des 3 autres fonctions, celle-ci pose des problèmes car dans le groupe considéré, on peut avoir différents types de support alors que pour les autres, on ne gère qu'un seul type à la fois.

4.6.2 La connexité

Avec l'ensemble des fonctions de recherche de voisins que nous avons présentées, nous pouvons, au bout d'un nombre fini d'itérations, obtenir n'importe quelle solution à partir d'une solution initiale. Notre recherche de voisinage est donc connexe dans l'espace des solutions

ce qui nous permet d'utiliser la méthode du recuit simulé.

4.6.3 La fonction d'évaluation

Pour comparer et évaluer les solutions proposées, nous avons développé une fonction calculant le nombre d'arrêts dans les gares et une autre calculant le nombre de passages dans les boucles.

4.6.4 La fonction de décroissance de la température

Dans le recuit simulé, la fonction de décroissance de la température joue un rôle important car c'est elle qui va assurer la convergence de l'heuristique. Afin de respecter au mieux cette contrainte, nous avons décidé d'utiliser la fonction de décroissance proposée par Hajeck (Hajeck, 1988) qui nous assure la convergence de l'heuristique.

Pour cela, nous avons une fonction de décroissance de la température de la forme:

$$T_n = \frac{T_0}{\ln(n+1)}$$

avec T_n température du recuit à l'itération n et T_0 température initiale.

La température initiale est un paramètre difficile à déterminer car elle dépend de la taille du problème posé et du nombre d'itérations que l'on souhaite effectuer. Pour chacun des problèmes que nous avons traités le choix des températures initiales s'est réalisé en fonction du temps de calcul à disposition et de la taille du problème à traiter.

Plus la température de départ est élevée, plus on va avoir tendance à accepter des solutions détériorées et on risque de ne pas converger vers une "bonne" solution. Par contre, si la température initiale est trop faible, on risque de "tomber" dans un minimum local et de ne pas avoir une meilleure solution.

4.6.5 Le nombre d'itérations

Le nombre d'itérations est un paramètre important de l'étude car il va déterminer le temps de calcul mais aussi la valeur de la solution. En effet, plus on génère de solutions (plus on a d'itérations), plus on a de chances d'avoir une "bonne" solution. Cependant, il est souhaitable de trouver le bon compromis entre temps de calcul et performance. Ce nombre d'itérations est à déterminer en fonction de la taille du problème et du temps de calcul que l'on a à disposition.

4.6.6 Codage de la solution

Pour le codage de la solution, les choix sont assez restreints à cause de l'ordre de grandeur important de notre problème. Par exemple, si nous adoptons un codage sous forme matricielle avec en lignes le numéro de gare et en colonne le numéro de produit. On se retrouve avec une

matrice binaire nécessitant une place mémoire conséquente. Pour une installation de n gares et p produits, nous avons donc besoin d'une place mémoire de $n.p.4$ octets (un entier nécessitant 4 octets pour être codé).

Nous avons donc décidé de nous orienter vers un codage sous forme vectorielle, la $i^{\text{ème}}$ composante du vecteur correspondant au numéro de gare où se situe le produit i . Pour la même installation que précédemment, on limite la place mémoire nécessaire à $p.4$ octets.

5. EXEMPLE

Le cas réel que nous devons traiter est une installation logistique composée de 4 746 articles à répartir dans 31 gares. Après une analyse détaillée du fichier des commandes représentant 4 jours de commandes client, nous nous sommes rendus compte que seuls 3 148 articles étaient utilisés. C'est pour cette raison que nous avons décidé de limiter notre étude de cas à ces 3 148 produits à répartir sur 31 gares.

Pour l'installation actuelle, sur les 4 jours de commandes client que nous traitons, le nombre d'arrêts nécessités pour ces produits est de 54 021. Après une itération pour détermination de la solution initiale et 10 000 itérations, soit au total environ 3 heures et 20 minutes de calcul, le nombre d'arrêts en gare est de 46 855 arrêts soit une amélioration de 13.3 %.

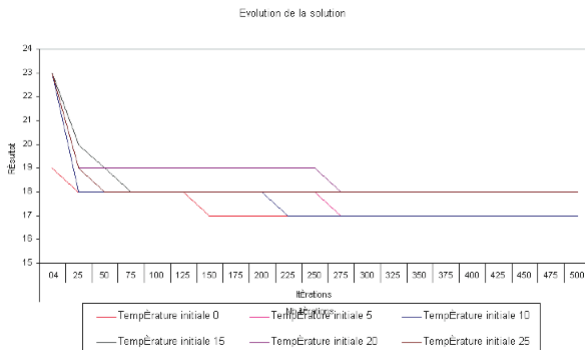


Figure 2. Évolution du résultat de la solution en fonction du nombre d'itérations

Au bout de 50 000 itérations, la solution est améliorée et passe à 46 050 arrêts soit une amélioration d'environ 15 % par rapport au mapping-produit actuel. On a effectué des calculs plus longs, mais à partir d'un certain nombre d'itérations, les améliorations sont minimales comme le montre le graphique d'évolution de la solution (Cf. Figure 2).

6. CONCLUSION

Dans un souci de rapidité et de service au client, les installations de préparation de commandes sont de plus en plus présentes en milieu industriel. On en retrouve dans tous les domaines, que ce soit la livraison de produits pharmaceutiques, l'édition du livre, les courses par inter-

net ... Ces installations sont plus rapides et performantes que les systèmes logistiques traditionnels comme les entrepôts. Leurs performances peuvent être nettement améliorées en affectant les produits au sein des gares de façon "intelligente". En effet, c'est en se basant sur un historique des commandes que nous avons pu définir une nouvelle affectation produits / gares plus performante. Tout au long de cette étude, nous avons pu voir que les résultats apportés par notre heuristique de détermination d'une solution initiale et le recuit simulé sont conséquents. En revanche, la prise en compte de toutes les contraintes n'est pas aisée et nécessite de nombreux tests. Les hypothèses simplificatrices de notre modèle permettent d'obtenir des résultats opérationnels validés par des industriels du secteur. Un des points délicats est aussi la recherche de voisin couplée au recuit simulé qui nécessite plusieurs types de permutations et de déplacements pour obtenir la connexité.

Ce type de problème reste très peu traité dans la littérature et la stratégie de résolution proposée améliore notablement les performances des installations de préparation de commandes puisque que l'on a des améliorations de l'ordre de 15 % sur des cas industriels. D'autres méthodes telles que les algorithmes génétiques et la recherche Tabu pourraient être testées sur ces systèmes logistiques et il serait intéressant de les comparer aux résultats fournis par le recuit simulé.

De plus, notre méthode prend en considération les contraintes les plus fréquentes. Elle peut toutefois être améliorée en intégrant des notions de dynamique, d'ergonomie de poste, de fragilité, de foisonnabilité. L'approche présentée dans cet article est basée sur l'étude d'un historique. Les variations de la demande pourrait donc nous conduire à essayer de prendre en compte cette variabilité afin de proposer un mapping-produit robuste aux industriels.

REMERCIEMENTS

Les auteurs tiennent à remercier la société Savoye Logistics pour avoir participé à cette étude et fourni tous les exemples industriels.

RÉFÉRENCES

- Abdelmola A. I. et S. M. Taboun, 1999. Productivity Model for the Cell Formation Problem : A Simulated Annealing Algorithm. *Computers and Industrial Engineering*, 37, p. 327 - 330
- Adenso-Diaz B., S. Lozano, J. Racero et F. Guerrero, 2001. Machine Cell Formation in Generalized Group Technology. *Computers and Industrial Engineering*, 41, p. 227 - 240
- Aljaber N., W. Baek et C.-L. Chen, 1997. A Tabu Search to the Cell Formation Problem. *Computers and Industrial Engineering*, 1, p. 169 - 185
- Brynzér H. et M. I. Johansson, 1996. Storage Location Assignment: Using the Product Structure to Reduce

- Order Picking Times. *International Journal of Production Economics*, 46 - 47, p. 595 - 603
- Caux C., R. Bruniaux et H. Pierreval, 2000. Cell Formation with Alternative Process Plans and Machine Capacity Constraints: A New Combined Approach. *International Journal of Production Economics*, 64, p. 279 - 284
- Chan C.-Y., F.-W. Lam et C.-P. Lee, 1999. Considerations for Using Cellular Manufacturing. *Journal of Materials Processing Technology*, 96, p. 182 - 187
- De Lit P., E. Falkenauer et A. Delchambre, 2000. Grouping Genetic Algorithms : An Efficient Method to Solve the Cell Formation Problem. *Mathematics and Computers in Simulation*, 51, p. 257 - 271
- Ek P. C. et C. T. Loon, 1999. Travel Time Analysis for General Item Location Assignment in a Rectangular Warehouse. *European Journal of Operational Research*, 112, p. 582 - 597
- Hajek B., 1988. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13, p. 311-329
- Hesket J. L., 1963. Cube Per Order Index. A Key to Warehouse Stock Location. *Transport Distribution Management*, 3, p. 27 - 31
- Hwang H. et J.-U. Sun, 1996. A Genetic-Algorithm-Based Heuristics for the GT Cell Formation Problem. *Computers and Industrial Engineering*, 30, p. 941 - 955
- Lecha F., 2002. *Affectation des produits dans les gares d'installations de préparation de commandes*. DEA Informatique, Productique, Imagerie Médicale, Université Blaise Pascal, Aubière, France.
- Lin C.-H. et I.-Y. Lu, 1999. The Procedure of Determining the Order Picking Strategies in Distribution Center. *International Journal of Production Economics*, 60 - 61, p. 301 - 307
- Logendran R. et C. Seong Ko, 1997. Manufacturing Cell Formation in the Presence of Flexible Cell. *Computers and Industrial Engineering*, 3-4, p. 545 - 548
- Onwubolu G. C. et M. Mutingi, 2001. A Genetic Algorithm Approach to Cellular Manufacturing Systems. *Computers and Industrial Engineering*, 39, p. 125 - 144
- Plaquin M.-F., 1998. *De la construction des algorithmes évolutionnistes à la construction d'ilots de fabrication*. Université Blaise Pascal,
- Roodbergen K. J. et R. De Koster, 2001. Routing Order Pickers in a Warehouse with Middle Aisle. *European Journal of Operational Research*, 133, p. 32-43
- Spiliopoulos K. et S. Sofianopoulou, 1998. An Optimal Tree Search for the Manufacturing Systems Cell Formation Problem. *European Journal of Operational Research*, 105, p. 537 - 551
- Sun D., L. Lin et R. Batta, 1995. Cell Formation Using Tabu Search. *Computers and Industrial Engineering*, 3, p. 485 - 494
- Van Den Berg J. P. et W. H. M. Zijm, 1999. Models for Warehouse Management: Classification and Examples. *International Journal of Production Economics*, 59, p. 519 - 528