OPTIMISATION COMBINATOIRE PAR ENTROPIE CROISÉE : APPLICATION À DES PROBLÈMES DE GRANDES TAILLES DANS LES ENTREPÔTS LOGISTIQUES

Séverine Durieux, Olivier Devise, Jean-Luc Paris, Pierre-Guillaume Fradet

Optimisation combinatoire par entropie croisée : application à des problèmes de grandes tailles dans les entrepôts logistiques

S. DURIEUX, O. DEVISE, J.-L. PARIS, P.-G. FRADET

Clermont Université – IFMA, Institut Pascal, BP 10448, F-63171 Aubière
CNRS, UMR 6602, Institut Pascal, F-63171 Aubière
durieux@ifma.fr, devise@ifma.fr, paris@ifma.fr, pfradet@ifma.fr

RÉSUMÉ : *L'objectif de cette étude est de présenter la méthode de l'entropie croisée pour trouver un placement des produits performant pour un système de préparation de commandes. Ce type de problème intitulé SAP ou SLAP pour Storage Assignment Problem ou Storage Location Assignment Problem consiste à définir l'affectation des produits dans les différentes emplacements de picking du système de préparation de commandes, sous un certain nombre de contraintes (type de support de stockage, contraintes capacitaires, foisonnement lors de la préparation, ...). Nous allons montrer que la méthode de l'entropie croisée est facilement et efficacement applicable à un tel problème puis nous allons développer cette méthode sur un exemple concret pour en montrer les performances..*

MOTS-CLÉS : *Optimisation combinatoire, entropie croisée, entrepôts logistique, storage assignment problem.*

1 INTRODUCTION

Le développement, ces dernières années, de plateformes logistiques et de systèmes de distribution a conduit à l'émergence de nouveaux enjeux pour réduire les temps d'attente des clients. Le but premier de la gestion d'entrepôts logistiques est de trouver la façon la plus efficace d'assurer la fonctionnalité de l'entrepôt au sein de la chaîne d'approvisionnement (Supply Chain), de l'opération de distribution de base jusqu'à d'autres activités innovantes à valeur ajoutée. Conduite par la demande constante des clients pour un service en un temps minimum, performant, de qualité, mais le moins onéreux possible, l'efficience des opérations logistiques est devenue cruciale dans le marché concurrentiel d'aujourd'hui. L'ordre de picking – le fait de retirer des produits des zones de stockage en réponse à une demande spécifique d'un client – est considéré comme étant le processus le plus coûteux en main d'œuvre et le plus critique en terme de temps dépensé (Frazelle 2002, Ten Hompel & Schmidt 2007, Tompkins, White, Bozer, Frazelle & Tanchoco 2003). Il est responsable de près de 60% de toutes les activités du travail d'un entrepôt selon Merkuryev, Merkuryeva & Burinskiene (2009).

L'optimisation du placement des produits fait partie de ces nouveaux enjeux, principalement dans les entreprises de distribution et de préparation de commandes. Les performances en termes de temps, de productivité et de qualité sont notablement améliorées par un placement des produits efficace. En plus de devoir traiter un problème complexe d'affectation, plusieurs contraintes viennent s'ajouter : configuration physique des entrepôts (capacité par type de support de stockage), règles de sécurité et de qualité (produits lourds, fragiles, ...).

De nombreuses méthodes ont été appliquées sur des problèmes d'optimisation combinatoire tels que le recuit simulé, les algorithmes génétiques ou encore la recherche Tabou. Nous allons dans cette étude nous intéresser à une méthode récente d'optimisation qui est la méthode de l'entropie croisée et voir comment cette méthode peut être appliquée au placement des produits.

La méthode de l'entropie croisée est une méthode générique récente d'estimation d'événements rares, adaptable à des problèmes d'optimisation. Elle se base sur un algorithme probabiliste dans lequel des densités de probabilités évoluent pour converger vers une densité fournissant un ensemble contenant la ou les solutions optimales. Cette méthode utilise donc une population de solutions à chaque itération, population générée selon des densités de probabilité. Ainsi, nous verrons que cette méthode présente une convergence rapide et une faible sensibilité aux optimums locaux.

Afin de présenter cette méthode encore peu connue et peu appliquée aux problèmes d'optimisation, nous allons en décrire les principes généraux dans une première partie. Puisque nous sommes dans le cas d'un problème d'optimisation combinatoire, nous nous attarderons plus sur la méthode appliquée à ce type de problème. Nous verrons que l'entropie croisée a été exploitée sur des estimations d'événements rares et beaucoup moins sur l'optimisation combinatoire. Ensuite, nous présenterons le Storage Assignment Problem (SAP) ainsi que les travaux qui peuvent nous éclairer sur la complexité d'un tel problème. Nous verrons notamment quels sont les objectifs et les contraintes dans le SAP et comment cela a été traité avec les méthodes classiques d'optimisation. Nous terminerons par un exemple basé sur un cas réel d'entreprise. Nous chercherons à montrer que l'entropie croisée peut s'appliquer au SAP et fournit des solutions optimales en un temps limité. Nous montrerons également que le problème est NP difficile au sens faible, c'est-à-dire que le fait de trouver des solutions optimales vérifiables en un temps polynomial dépend des entrées et non pas de la structure du problème. Nous verrons également les difficultés liés à l'utilisation de la méthode de l'entropie croisée et les manières de les éviter.

2 L'ENTROPIE CROISÉE

Beaucoup de tâches dans les travaux de recherche impliquent de résoudre des problèmes d'optimisation complexes. Ces problèmes peuvent être classés en différentes catégories selon leur temps de résolution et l'optimalité des solutions. Le problème du voyageur de commerce (TSP), le problème d'assignement quadratique (QAP) ou encore le problème de découpage maximal (max-cut problem) sont des exemples de problèmes d'optimisation combinatoire. Cependant, dans ces problèmes, les données sont connues et statiques. Il peut arriver qu'il faille évaluer la fonction objectif, celle-ci étant inconnue et soumise à des bruits, c'est-à-dire à des perturbations d'ordre statistique.

Beaucoup de problèmes d'optimisation combinatoire peuvent être ramenés à des graphes pondérés. Comme énoncé dans cet article sur l'entropie croisée (De Boer, Krose, Mannor & Rubinstein 2005), nous passons d'un problème déterministe à un problème stochastique pour pouvoir appliquer la méthode des événements rares. Sur le graphe, l'aléatoire peut être introduit au niveau des nœuds ou au niveau des arcs. Nous parlons alors de stochastic node networks (SNN) dans le premier cas, ou de stochastic edge networks (SEN) dans le second. Ainsi, le TSP peut être traité comme un SEN, tandis que les problèmes de découpage maximal ou d'allocation de stocks tampons sont traités dans des SNN.

À l'origine, la méthode de l'entropie croisée a été créée par le besoin d'estimer les probabilités d'événements rares dans des réseaux stochastiques complexes (Rubinstein 1997). Utilisant un algorithme adaptatif, la méthode de l'entropie croisée vise à réduire les variations de la variance et ainsi converger vers les probabilités des événements rares. Cependant, il est apparu assez rapidement que cette méthode pouvait être adaptée à des problèmes d'optimisation combinatoire (Rubinstein 1999, Rubinstein 2001a, Rubinstein 2001b). Cela est possible en transformant un problème d'optimisation avec des données déterministes en un problème stochastique associé puis en appliquant à ce problème les techniques des événements rares. Nous verrons cela dans la partie concernant la méthode appliquée à des problèmes d'optimisation combinatoire. Des applications récentes montrent que la méthode de l'entropie croisée est très puissante en tant qu'outil générique et pratique pour étudier des problèmes NP-difficiles.

En effet, les problèmes plus complexes peuvent être étudiés par des techniques de simulation. La plus simple mais la moins efficace d'entre elle est la simulation de Monte Carlo (Rubinstein & Kroese 2004) dans laquelle le système est simulé sous les paramètres normaux opératoires sur une longue durée. Une méthode plus intelligente est la méthode de l'Importance Sampling (IS), où les paramètres sont choisis de manière à mieux rendre compte des apparitions des événements rares. L'inconvénient majeur est de trouver un échantillon optimal alternatif sous lequel nous voudrions simuler le système. Ainsi, la méthode de l'entropie croisée agit comme une technique adaptative de simulation IS. Elle permet de trouver par itérations successives ces paramètres optimaux en les faisant évoluer à chaque itération.

D'autres méthodes sont également utilisées. Les plus connues sont le recuit simulé développé par Aarts & Laarhoven (1985), la recherche Tabu de Glover & Laguna (1997) et les algorithmes génétiques développés par Goldberg (1989). D'autres travaux plus récents sur les problèmes d'optimisation combinatoires stochastiques sont l'optimisation par colonies de fourmis de Dorigo, Maniezzo & Colorni (1996) et la méthode de partition sans imbrication, décrite par Shi & Olafsson (2000).

Nous allons dans cette partie développer la méthode générale de l'entropie croisée afin de présenter son fonctionnement. Il est à noter que pour des raisons de concision nous n'étudierons cette méthode que dans le cadre de l'optimisation combinatoire puisque nous cherchons à résoudre un problème de type SAP. Pour cela nous expliquerons comment passer d'un problème d'optimisation à une estimation d'événements rares pour pouvoir utiliser les outils de l'entropie croisée. Ensuite, nous approfondirons en nous intéressant à cette méthode appliquée à des problèmes d'optimisation combinatoire discrets et continus. En vue d'étudier un problème de SAP, nous nous consacrerons plus à l'optimisation combinatoire discrète que nous illustrerons par un problème simple. Pour finir, nous chercherons les applications faites de la méthode de l'entropie croisée dans des problèmes d'optimisation combinatoire. Pour les lecteurs voulant appliquer la méthode de l'entropie croisée, nous leur conseillons la lecture de (De Boer et al. 2005, Devise, Paris, Durieux & Fradet 2014) car nous n'avons pas la place ici, de détailler complètement sont application.

2.1 Principe général

Nous allons ici retracer les étapes générales de la méthode de l'entropie croisée (Rubinstein & Kroese 2004). Pour plus de détails, le lecteur peut consulter le site http ://www.cemethod.org.

La particularité de l'entropie croisée est de fournir un cadre de travail précis pour dériver rapidement et ainsi obtenir des itérations par apprentissage optimales basées sur des théories de simulation.

La méthode de l'entropie croisée fait appel à une procédure itérative ou chaque itération peut être décomposée en deux phases :

- Générer un échantillon aléatoire de solutions selon un mécanisme probabiliste donné.
- Mettre à jour les paramètres probabilistes du mécanisme de génération en ne conservant que les solutions qui donnent les meilleurs résultats afin d'obtenir un échantillon de solutions de meilleure qualité à la prochaine itération.

Nous allons donner l'algorithme général de la méthode de l'entropie croisée dans le cadre de l'optimisation combinatoire en définissant les outils mathématiques utilisés.

Considérons le problème d'optimisation comme un problème de maximisation. Prenons X un ensemble fini d'états et S la fonction objectif, ou fonction de performance, à valeurs réelles sur X. Nous voulons trouver le maximum de S sur X et x^* l'état auquel ce maximum est atteint. Notons γ^* ce maximum. Ainsi

$$S(x^*) = \gamma^* = \max_{x \in X} S(x) \tag{1}$$

Le premier point dans la méthodologie de la méthode de l'entropie croisée est d'associer un problème d'estimation au problème de maximisation énoncé au dessus. Pour cela, nous définissons une collection de fonctions indicatrices $\{I_{\{S(x) \geqslant \gamma\}}\}$, qui sont des fonctions de Kroenecker adaptées au problème, qui sont définies sur X et varient selon des paramètres ou des niveaux γ. Ensuite, prenons comme ensemble de fonction densité de probabilité (fddp) sur X. Chaque ffdp peut être discrète ou continue et est paramétrée par un paramètre (vecteur) à valeurs réelles noté v.

Pour un certain vecteur paramétrique u nous associons à la formule précédente le problème d'estimation du nombre

$$\begin{aligned} \ell(\gamma) &= \mathbb{P}_u(S(X) \geqslant \gamma) && (2)\\ &= \sum_x I_{\{S(x) \geqslant \gamma\}} f(x;\, u) && (3)\\ &= \mathbb{E}_u\, I_{\{S(x) \geqslant \gamma\}} && (4) \end{aligned}$$

où $\mathbb{P}_u$ est la probabilité sous laquelle l'état X a la fddp $f(.;\, u)$ et $\mathbb{E}_u$ correspond à l'espérance sous les mêmes conditions. Nous appellerons le problème d'estimation (Eq. 4) le problème stochastique associé, autrement dit associé au problème de maximisation de départ. Pour mieux comprendre comment les deux problèmes sont associés, supposons que γ ait la valeur γ^* et que $f(.;\, u)$ soit la densité de probabilité uniforme sur X. Ainsi, nous remarquons que $\ell(\gamma^*) = f(x^*;\, u) = 1/l\chi l$ où $l\chi l$ correspond au cardinal de X, est typiquement un petit nombre. Alors, pour estimer la valeur de $\ell(\gamma)$ une solution serait d'utiliser le ratio de vraisemblance (LR) avec le paramètre de référence v^* donné par :

$$v^* = \arg\max_v \mathbb{E}_u\, I_{\{S(X) \geqslant \gamma\}} \ln f(X;\, v) \tag{5}$$

Ce paramètre peut être estimé par :

$$\widehat{v^*} = \arg\max_v \frac{1}{N} \sum_{i=1}^{N} I_{\{S(X_i) \geqslant \gamma\}} \ln f(X_i;\, v) \tag{6}$$

où les états X_i sont générés à partir de la fddp $f(.;\, u)$.

Nous nous trouvons alors dans le cas où estimer avec $\ell(\gamma)$ très proche de γ^* revient à estimer un événement rare. Nous pouvons alors appliquer les algorithmes de l'entropie croisée sur les événements rares en les adaptant à notre problème. Le résultat est le suivant :

1. Définir un paramètre initial $\widehat{v_0} = u$ et mettre le compteur d'itération à $t = 1$.
2. Générer un échantillon de solutions X_1, X_2, ..., X_N à partir de la fonction de densité de probabilité $f(.; v_{t-1})$ et traiter l'échantillon $(1 - \rho)$, correspondant au quantile $\widehat{\gamma_t}$ des meilleures fonctions performances avec $\widehat{\gamma_t} = S_{(\lceil (1-\rho)N \rceil)}$.
3. Utiliser l'échantillon $X_1, X_2, \ldots, X_N$ pour résoudre l'équation stochastique
$$\begin{aligned} \max_v \widehat{D}(v) = \max_v \frac{1}{N} \sum_{i=1}^{N} I_{\{S(X_i) \geqslant \widehat{\gamma}_t\}}. \\ .W(X_i;\, u;\, \widehat{v}_{t-1}) \ln f(X_i;\, v) \end{aligned} \tag{7}$$
avec $W = 1$ puisque nous sommes en optimisation combinatoire.
4. Si pour $t \geqslant d$, $\widehat{\gamma}_t = \widehat{\gamma}_{t-1} = \ldots = \widehat{\gamma}_{t-d}$, alors arrêter le programme ; sinon incrémenter t de 1 et réitérer depuis l'étape 2.

2.2 Remarques

Une différence principale entre l'estimation d'événements rares et l'optimisation combinatoire est le rôle du vecteur de paramètre initial u. Pour l'estimation, u caractérise le système considéré tandis que pour l'optimisation u est déterminé arbitrairement pour initialiser le problème. Dans le cas du SAP, cela consistera à générer une solution initiale à notre problème d'affectation en terme probabiliste.

Plutôt que de mettre à jour $\widehat{v}_t$ selon (Eq. 5), il est souvent bénéfique d'utiliser la fonction suivante :

$$\widehat{v}_t = \alpha\widehat{w}_t + (1-\alpha)\widehat{v}_{t-1} \tag{8}$$

où $\widehat{w}_t$ est le vecteur dérivé de (Eq. 5). En utilisant cette fonction, nous évitons de trop grandes récurrences de zéros ou de uns dans les vecteurs de paramètres. En effet, une fois que l'on trouve un 0 ou un 1 dans un vecteur paramètre, nous allons les retrouver sans arrêt par la suite ce qui est indésirable. Selon (Rubinstein & Kroese 2004), une valeur de α comprise entre 0,4 et 0,9 donne les meilleurs résultats. L'intérêt de ce système est aussi d'implémenter un effet mémoire entre deux itérations.

Une version déterministe de l'algorithme présentée précédemment peut être définie. Un exemple sera développé dans les prochains paragraphes.

2.3 Les paramètres

Le choix de la taille de l'échantillon de solutions pour chaque itération, que nous avons noté N, et du paramètre ρ dépend de la taille du problème et du nombre de paramètres dans le problème stochastique associé. En particulier pour un problème de type SNN il est suggéré de prendre une taille d'échantillon $N = vn$ où n est le nombre de noeuds et c une constante strictement supérieure à 1. Par contraste, pour un problème de type SEN, il convient de prendre $N = cn^2$, où n^2 est le nombre d'arcs du graphe. Il est crucial de comprendre que ces tailles sont également associées au nombre de paramètres du problème stochastique associé.

En ce qui concerne ρ, il est généralement suggéré de prendre ρ autour de 0,001 si $n > 100$ et ρ de l'ordre de $\frac{\ln n}{n}$ pour $n < 100$.

2.4 Application à l'optimisation combinatoire discrète

Nous allons présenter ici un exemple d'application extrêmement simple de l'entropie croisée à un problème d'optimisation combinatoire en discret.

Considérons un vecteur binaire $y = (y_1, y_2, \ldots, y_n)$. Supposons que nous ne puissions pas déterminer les valeurs prises par chaque élément de ce vecteur. Cependant, nous avons accès à une fonction « oracle » qui pour tout vecteur binaire $x = (x_1, x_2, \ldots, x_n)$ en entrée nous renvoie la valeur :

$$S(x) = n - \sum_{j=1}^{n} |x_j - y_j| \tag{9}$$

Cette valeur correspond au nombre d'éléments identiques dans les vecteurs x et y. L'objectif est alors de construire un algorithme de recherche pour reconstruire le vecteur inconnu y. Pour cela, nous sommes dans le cadre d'un problème de maximisation de la fonction S dans l'espace des vecteurs binaires à n éléments.

Nous pouvons trouver y en générant de manière répétée des vecteurs binaires $X = (X_1, X_2, \ldots, X_n)$ tels que $X_1, X_2, \ldots, X_n$ soient des variables aléatoires indépendantes de Bernoulli avec les probabilités p_1, p_2, $\ldots$, p_n. Nous écrivons $X \sim Ber(p)$ où $p = (p_1, p_2, \ldots, p_n)$. Nous pouvons noter que si $p = y$, ce qui correspond au cas de dégénérescence de la distribution de Bernoulli, nous avons $S(X) = 0$ et $X = y$ et l'algorithme de recherche de la solution optimale fournit la solution avec une probabilité de 1. Ici, nous adaptons un problème d'optimisation combinatoire en créant une séquence de vecteurs paramètres $\widehat{p}_0$, $\widehat{p}_1$, ... et de niveaux $\widehat{\gamma}_1$, $\widehat{\gamma}_2$, ... tels que la séquence des niveaux converge vers l'optimum, ici n, et la séquence des vecteurs $\widehat{p}_0$, $\widehat{p}_1$, ... converge vers le vecteur y.

Pour cela nous développons l'algorithme suivant :

1. Commençons avec le vecteur $\widehat{p}_0$ tel que $\widehat{p}_0$= (0,5 ; 0,5 ;... ; 0,5). Prenons $t = 1$, la première itération.
2. Générons un échantillon $X_1, X_2, \ldots, X_N$ de vecteurs de Bernoulli avec le vecteur de probabilités $\widehat{p}_{t-1}$. Calculons la performance $S(X_i)$ pour chaque vecteur X_i et classons les selon leur performance par ordre croissant : $S_1 < S_2 < \ldots < S_N$. Prenons ensuite $\widehat{\gamma}_t$ comme le $(1\breve{}\rho)$ quantile des performances. Nous écrivons $\widehat{\gamma}_t = S_{(\lceil(1-\rho)N\rceil)}$.
3. Utilisons ce même échantillon pour calculer $\widehat{p}_t = (\widehat{p}_{t,1}, \ldots, \widehat{p}_{t,n})$ selon la formule :
$$\widehat{p}_{t,j} = \frac{\sum_{i=1}^{N} I_{\{S(X_i)\geqslant\widehat{\gamma}_t\}} I_{\{X_{ij}=1\}}}{\sum_{i=1}^{N} I_{\{S(X_i)\geqslant\widehat{\gamma}_t\}}} \tag{10}$$
avec $j = 1, \ldots, N$ et où $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})$ et incrémenter t de 1.
4. Si le critère d'arrêt est atteint, arrêter le programme sinon recommencer à l'étape 2.

2.5 Critère d'arrêt

Un possible critère d'arrêt est de stopper lorsque $\widehat{\gamma}_t$ n'évolue plus durant un nombre d'itérations fixé. Nous pouvons également nous arrêter lorsque le vecteur de probabilité $\widehat{p}_t$ a convergé et est devenu un vecteur binaire, auquel cas il n'est plus pertinent de poursuivre l'algorithme. En effet, il est montré par Homem de Mello (2007) que pour des modèles statiques avec des contraintes faibles, la méthode de l'entropie croisée donne des résultats avec une probabilité de 1 ou 0 en un nombre fini d'itérations.

Notons que la formule de calcul de $\widehat{p}_{t,j}$ est simple de compréhension. Nous avons au départ un échantillon de vecteurs puis nous les classons selon leur performance. Ensuite, pour chaque élément du vecteur de probabilité, nous faisons une division où :

- Le numérateur correspond au nombre de fois où l'on a 1 dans les vecteurs donnant les meilleures performances et gardées dans le quantile déterminé par la borne $\widehat{\gamma}_t$.
- Le dénominateur correspond au nombre de vecteurs du quantile.

Ainsi, en effectuant la division, nous obtenons un nombre entre 0 et 1 pour chaque $\widehat{p}_{t,j}$ qui augmente si les meilleures solutions ont $X_j = 1$ et décroît si $X_j = 0$. Nous faisons alors converger chaque $\widehat{p}_{t,j}$ vers 0 ou 1 selon que les vecteurs X_i ayant les meilleures performances ont plutôt des 0 ou des 1 à l'emplacement j.

2.6 Exemple simple

Nous avons développé un programme simple permettant de tester l'entropie croisée sur le problème de recherche d'un vecteur binaire que nous avons présenté précédemment.

Pour cela, nous avons généré un vecteur binaire aléatoire de dimension 5 puis nous avons défini la fonction objectif qui renvoie le nombre d'éléments correspondant entre le vecteur recherché et une solution.
Le vecteur recherché est le suivant :

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Pour ce problème de petite taille, nous avons choisi une taille de population de 50 solutions dont nous ne gardons que les 10 meilleures, soit 20% de la population. Ainsi, le quantile est défini par la performance de la dixième moins bonne solution.

Nous obtenons alors les résultats présenté dans le tableau 1 :

Itérations	n° 1	n° 2	n° 3	...	n° 7
fddp de x_1	0,10	0	0	...	0
fddp de x_2	0,70	1	1	...	1
fddp de x_3	0,10	0	0	...	0
fddp de x_4	0,70	1	1	...	1
fddp de x_5	0,80	1	1	...	1

Tableau 1 – Évolution des fddp à chaque itération

Nous affichons le vecteur de probabilité à chaque itération. Nous remarquons que le vecteur de probabilité converge très rapidement vers la solution avec une probabilité de 1 (le tirage aléatoire est fait en nombre entiers entre 0 et 9, 9 correspond donc à la probabilité 1).

Le critère d'arrêt porte ici sur le nombre de fois où l'on retrouve un quantile inchangé. Ainsi, nous incrémentons un compteur à l'itération 2 jusqu'à l'itération 7. Notre condition d'arrêt était compteur = 5.

Nous remarquons déjà la puissance de la méthode sur un problème NP complet de petite dimension. Nous pouvons supposer qu'une telle méthode peut être étendue à ce type de problème pour des dimensions plus grandes en augmentant la taille de la population.

2.7 Littérature sur l'entropie croisée

De nombreux articles et thèses ont été effectués sur l'entropie croisée et ces applications. Cependant, très peu confrontent l'entropie croisée à des problèmes d'optimisation combinatoire. En effet, la majorité concerne l'estimation d'événements rares dans les communications, les structures ou encore les problèmes de réseau. Nous avons cependant pu trouver certains documents se rapportant à l'entropie croisée appliquée aux problèmes d'optimisation combinatoire.

C'est par exemple le cas de Alon, Kroese, Raviv & Rubinstein (2005) qui ont travaillé sur les stocks tampons en utilisant l'entropie croisée. Dans ce problème, la solution est représentée par un vecteur et ensuite les auteurs cherchent à appliquer la méthode de l'entropie croisée. Nous remarquons alors que ce problème est comparable à l'exemple que nous avons pris pour illustrer la méthode en optimisation combinatoire discrète. En effet, l'allocation de stock tampon revient à rechercher un vecteur optimal que nous ne connaissons pas mais dont vers lequel il est possible de converger en estimant les performances des solutions puis en ne conservant que les meilleures. Les résultats obtenus sont optimaux à 99% et cela dans un temps optimal. Cette étude est tout à fait intéressante pour comprendre la méthode dans le cadre de l'optimisation combinatoire ainsi que les façons de représenter les données du système pour pouvoir l'appliquer.

D'autres auteurs tels que Caserta & Nodar (2009) ont également cherché à développer la formulation des problèmes d'optimisation combinatoire pour pouvoir appliquer l'entropie croisée. De nouveaux algorithmes ont été proposés pour prendre en compte de manière générale les contraintes. Les auteurs évoquent la méthode d'acceptation rejet que nous utiliserons également dans le problème du SLAP, qui consiste à générer une solution pour ensuite tester son respect des contraintes. Si la solution est viable au vu des contraintes, elle est stockée pour former l'échantillon nécessaire à chaque itération de la méthode. Les contraintes qu'ils ont alors choisi d'imposer sont assimilables aux contraintes du problème bin packing.

Comme nous l'avons dit précédemment, la méthode de l'entropie croisée appliquée à des problèmes d'optimisation combinatoire discrète fournit une solution optimale avec une probabilité proche de 1 dans le cas où cette solution est unique. Ce résultat démontré par Costa, Jones & Kroese (2007) est une garantie que les solutions convergent vers une solution optimale sans risque de s'arrêter à un optimum local. En outre, la mise à jour des probabilités à partir d'un ensemble de solutions à chaque itération rend la convergence plus rapide que dans d'autres méthodes où la recherche se fait solution par solution.

3 LES PROBLEMATIQUES DES ENTREPÔTS LOGISTIQUES

Le présent document s'attachera à poser les bases des enjeux de conception de systèmes d'entrepôts afin de prendre en compte les facteurs qui influent sur les performances du système de préparation de commandes. En effet, lors de la conception d'un entrepôt logistique, on peut dénombrer sept facteurs très influents selon Bukchin, Khmelnitsky & Yakuel (2012) : le lotissement (batching en anglais), la séquence de prélèvement, la politique de stockage, le zonage, l'aménagement de l'entrepôt, les équipements de picking et la conception de l'information de picking. Plus généralement, nous référençons dans la littérature 4 grands axes fondamentaux dans la conception et la modélisation d'un entrepôt (Hsieh & Tsai 2005) que nous adopterons pour définir notre cadre d'étude :

1. le schéma de configuration de l'entrepôt et son aménagement ;
2. la stratégie d'assignement du stockage ;
3. la politique de chemin à suivre par le préparateur de la commande ;
4. la combinaison et l'ordonnancement des commandes.

Dans notre problématique, la question est plus complexe puisqu'il s'agit d'optimiser l'implantation des références de produits en picking dans le but de minimiser la distance totale parcourue. Enfin la difficulté majeure à laquelle nous sommes confrontés réside dans le fait que nous ne disposons pas de précurseurs quant à la technique que nous proposons d'utiliser, l'entropie croisée, appliquée à des techniques d'optimisation d'implantations picking.

Notre problème fait référence au *Storage Location Assignment Problem* (SLAP) ou *Storage Assignment Problem* (SAP) ou encore *Storage Allocation Strategy* qui sont souvent abordés dans la littérature. De nombreux chercheurs ont travaillé sur ce sujet car il est source de nombreuses améliorations et de gains, à la fois en terme de taux d'occupation de l'espace et en coût de mise en stock et de déchargement, etc. La résolution de ce problème vise à fournir un moyen efficace de localiser les produits dans l'ordre dans un entrepôt dans le but de réduire les efforts nécessaires à la préparation de la commande. Cela influe sur quasiment tous les indicateurs de performance d'un entrepôt comme par exemple la préparation des commandes et leur coût associé (Muppani & Adil 2008).

Nous prendrons simplement ici l'exemple d'Ene & Oztürk (2012) qui allient à l'étude de l'optimisation des implantations de produits dans l'industrie automobile à l'optimisation du prélèvement par picking des commandes de produits. Pour ce faire, ils procèdent en deux étapes : à la première, le problème d'affectation de l'emplacement de stockage est résolu avec une politique de stockage basée sur les classes de produits dans le but de minimiser le flux d'éléments transmis au sein de l'entrepôt ; à la seconde, les problèmes de lotissement de produits et de routage sont considérés simultanément pour réduire les coûts des opérations de transport dans l'entrepôt en question. Ils parviennent, au moyen d'algorithmes génétiques à appliquer cela à tout type de disposition d'entrepôt dans l'industrie de l'automobile. Enfin, les problèmes de type SAP font partie de la classe des problèmes NP-difficiles (Frazelle & Sharp 1989). Par conséquent, un grand nombre de méthodes heuristiques ont été proposées. D'autre part, le routage ou le chemin du préparateur de la commande au sein d'un entrepôt a également reçu une attention considérable dans la littérature ayant trait à la recherche opérationnelle. Il a été largement démontré et maintenant reconnu que la détermination du chemin du préparateur de commande sous contrainte de minimisation de la distance dans un entrepôt est une variante du problème bien connu du Traveller Salesman Problem (TSP), qui est NP-difficile(Laporte 1992, Lawler, Lenstra, Rinnooy Kan & Shmoys 1985). En bref, le TSP consiste à déterminer la distance minimale d'un cycle passant par chaque sommet une fois et une seule. Dans le cas du chemin du préparateur de commande, chaque produit ou chaque article à prélever correspond à un sommet, et la distance qui sépare deux sommets entre eux doit être la plus courte possible entre les emplacements correspondants dans l'entrepôt. Habituellement, puisque les distances sont symétriques, le fait de trouver l'itinéraire le plus court pour effectuer le picking correspond à un TSP symétrique. Toutefois, dans certains entrepôts à allées unidirectionnelles, la matrice de distance est asymétrique, et de fait, le TSP correspondant aussi. De nombreux algorithmes efficaces de TSP existent et peuvent être utilisé pour optimiser les itinéraires du préparateur de la commande.

Le problème de trouver le chemin optimal du préparateur de la commande, sous le nom de TSP, masque en fait un problème sous-jacent encore plus connu qui est le problème du plus court-chemin (shortest path problem en anglais). Ce problème trouve de très nombreuses applications dans la recherche opérationnelle, mais nous n'avons pu trouver de travaux se rapportant à des problèmes d'implantations dans le but de trouver le plus court chemin. En revanche, nous avons développé le problème du plus court chemin dans cet article (Devise et al. 2014). Il peut être résolu et applicable à notre cas selon certaines hypothèses.

Ajoutons à cela que les stratégies de localisation des produits (c'est-à-dire l'affectation des produits dans les emplacements en vue de faciliter la préparation des commandes) ont reçu moins d'attention. Les contributions théoriques sont rares, étant donné que l'emplacement du produit dépend fortement de la configuration de l'entrepôt, de la stratégie de préparation des commandes et de la technologie en place. Ainsi, la plupart des recherches dans la littérature traitent de cas pratiques bien spécifiques.

4 APPLICATION INDUSTRIELLE

Notre cas industriel est un entrepôt réel d'un des leader européen en logistique, plus particulièrement spécialisé dans la distribution des grandes et moyennes surfaces (GMS). Cet entrepôt est séparé en plusieurs blocs (en fonction de la disposition des quais d'entrée/sortie). Nous avons étudié un bloc possédant environ 1 100 emplacement de préparation de commande. On place un produit par emplacement en tenant compte d'un certain nombre de contraintes supplémentaires : les produits les plus lourd doivent être placés en amont des produits plus léger et identifiés comme plus fragiles ; les produits doivent être placés dans le bon type d'emplacement. Il existe, suivant les entrepôts, de trois à cinq types d'emplacement. Dans cet entrepôt, on retrouvait des emplacements palettes, des demi-emplacements et des emplacements en dynamique. L'estimation de l'ordre de grandeur de la combinatoire du problème est de $8.35\,10^{750}$ solutions différentes (Devise et al. 2014).

La fonction d'évaluation est la distance totale parcourue par les préparateurs basée sur un historique de commande d'un mois (novembre 2013). Cela représente un total pour ce bloc d'environ 11 000 commandes différentes. La distance calculée prend en compte les commandes nécessitant plusieurs palettes et intègre donc les allers retours pour prendre une nouvelle palette. Nous calculons également le nombre de palettes préparées car, pour les commandes nécessitant plusieurs palettes, l'ordre dans lequel on prend les produits de la commande pour constituer des palettes influe sur le nombre de palette préparée car, un lot de produit identique ne doit pas être éclaté sur des palettes différentes. Par exemple, si l'on doit placer 10 packs de 6 jus de fruits, il faut vérifier qu'il reste assez de place sur la palette, sinon, on doit attaquer une nouvelle palette avec ces 10 packs. Le chemin de préparation est fixé en partie à l'avance pour assurer des flux fluides et pour respecter des règles de sécurité dans les allées. En effet, suivant la largeur des allées, on peut autoriser ou non le croisement des opérateurs.

Nous avons programmé en C++ le programme d'optimisation. Il lit un fichier excel standard de l'entreprise pour connaître la disposition de l'entrepôt ainsi que la codification et le type des emplacements. Il lit aussi une extraction du WMS (Warehouse Management System) afin de récupérer la liste des commandes à traiter. Ensuite, une méthode est utilisée pour optimiser par entropie croisée le placement des produits dans l'entrepôt.

Nous avons mené pour le moment des essais succincts sur les paramètres de réglage de l'entropie croisée. Les principaux paramètres sont
- le nombre de solutions générées à chaque itération
- le taille du quantile qui permet de mettre à jour les fonctions de densité de probabilité
- le nombre maximum d'itération.

Dans la figure 1, nous générons $N = 1\,500$ solutions à chaque itération. Le quantile prélevé des meilleures solutions est $(1 - \rho) = 10\%$. Nous avons également choisi d'avoir un effet mémoire en ajoutant à ce quantile les cinq meilleures solutions trouvées depuis la première itération afin, au vue de la taille du problème, qu'elle ne risque pas d'être "oubliées". En plus, nous faisons 10 réplications avec un générateur aléatoire différent afin de tester la robustesse de la méthode. Ces réplications entraîne évidemment une augmentation du temps de calcul par approximativement 10. Actuellement, le programme d'optimisation prend environ 3 heures pour traiter le cas industriel sur un PC ou un Mac portable. Par rapport aux méthodes utilisées et programmées par le service IT de l'entreprise dans les WMS nous avons un gain de 25 % sur la distance totale parcourue par les préparateurs des commandes.

5 CONCLUSION

Dans cet article, nous avons rappelé le problème général de la conception des entrepôts et nous nous

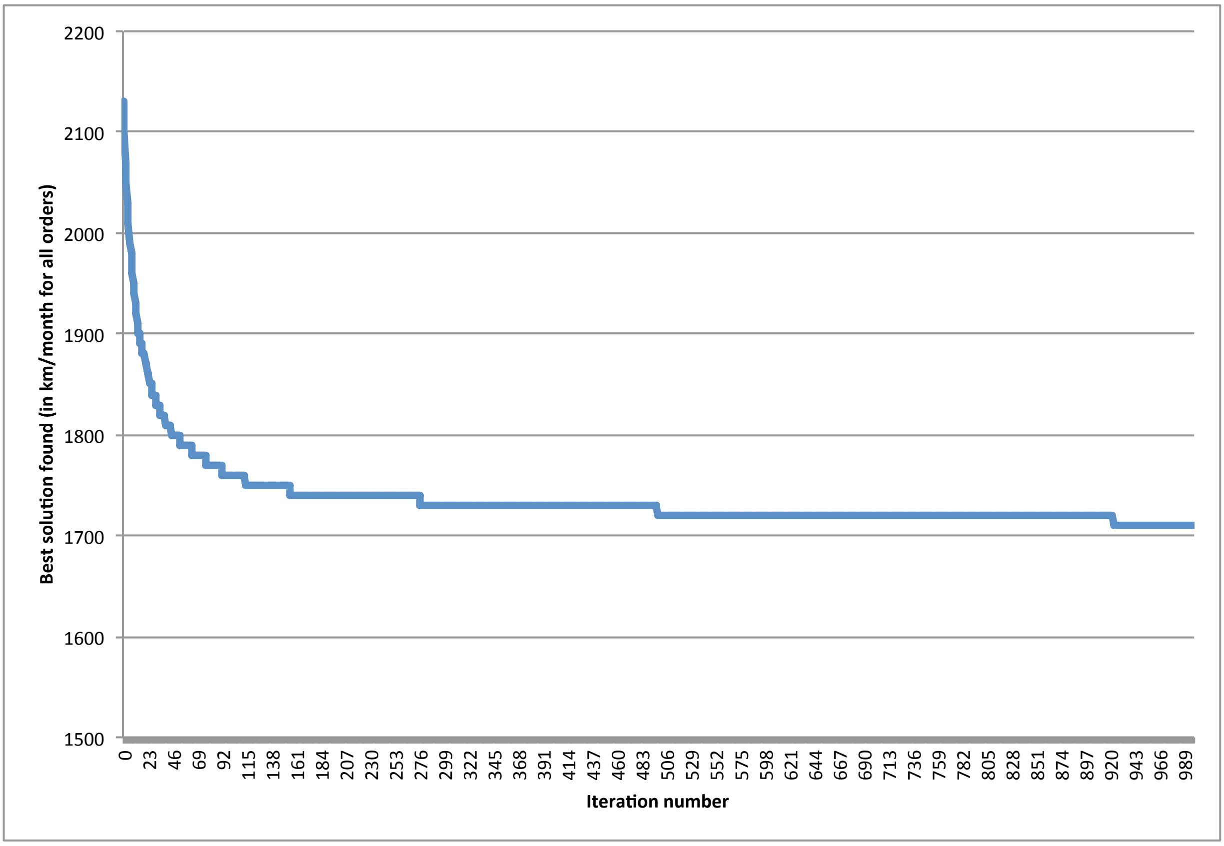

FIGURE 1 – Évolution de la meilleure solution à chaque itération

sommes plus particulièrement intéressé à la résolution du problème d'optimisation combinatoire intitulé *Storage Location Assignment Problem*. Nous avons mis en évidence l'intérêt de l'utilisation de la méthode de l'entropie croisée inventée par Rubinstein (1997). Les premiers résultats sur un problème industriel de grande taille sont prometteurs. Par rapport aux solutions utilisées jusque là, nous obtenons une amélioration de 25 % de la distance parcourue par les opérateurs en proposant une solution issue de l'entropie croisée au SLAP. Lors d'une étude utilisant le recuit simulé (Devise & Caux 2010) sur un système légèrement différent (les chemins étaient fixes), nous obtenions une amélioration de la performance du système de préparation de commande beaucoup plus faible alors que l'entreprise ne possédait aucun système pour optimiser ses emplacements de picking pour un temps de calcul légèrement supérieur.

Il nous reste maintenant à confirmer que la méthode de l'entropie croisée permet effectivement d'être, ainsi qu'il le semble au vue des premiers résultats, plus performantes que les méthodes d'optimisation que nous avions déjà tester. Pour ce faire, nous envisageons de traiter le même cas industriel avec d'autres méthodes d'optimisation (en particulier, le recuit simulé et les algorithmes génétiques).

En parallèle, il est nécessaire également de conduire une étude plus fine sur les deux principaux paramètres de réglages de la méthode de l'entropie croisée, à savoir la taille N des échantillons et le quantile $(1-\rho)$ des meilleures solutions utilisées pour mettre à jour la fonction de densité de probabilité.

REMERCIEMENTS

Les auteurs veulent remercier le grand groupe industriel français spécialisé dans la logistique pour avoir fourni l'ensemble des données réelles et permis de tester l'efficacité de cette méthode dans leurs entrepôts.

REFERENCES

Aarts, E. H. L. & Laarhoven, V. (1985). Statistical cooling : A general approach to combinatorial optimization problems, *Philips Journal Research* **40**(4) : 193–226.

Alon, G., Kroese, D. P., Raviv, T. & Rubinstein, R. Y. (2005). Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment, *Annals of Operations Research* **134**(1) : 137–151.

Bukchin, Y., Khmelnitsky, E. & Yakuel, P. (2012). Optimizing a dynamic order-picking process, *European Journal of Operational Research* **219**(2) : 335 – 346.

Caserta, M. & Nodar, M. C. (2009). A cross entropy based algorithm for reliability problems, *Journal of Heuristics* **15**(5) : 479–500.

Costa, A., Jones, O. D. & Kroese, D. P. (2007). Convergence properties of the cross-entropy method for discrete optimization, *Operations Research Letters* **35**(5) : 573–580.

De Boer, P.-T., Krose, D. P., Mannor, S. & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method, *Annals of Operations Research* **134**(1) : 19–67.

Devise, O. & Caux, C. (2010). Modeling for design and optimization of product-mapping by simulated annealing in an automated distribution system, *Journal Européen des Système Automatisé* **44**(2) : 209–228.

Devise, O., Paris, J.-L., Durieux, S. & Fradet, P.-G. (2014). Storage assignment problem in logistics warehouses : Optimization of picking locations by cross-entropy method, *IDMME – Virtual Concept – Improve – Ingegraf 2014*, Toulouse, France.

Dorigo, M., Maniezzo, V. & Colorni, A. (1996). Ant system : optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics–Part B* **26**(1) : 29–41.

Ene, S. & Oztürk, N. (2012). Storage location assignment and order picking optimization in the automotive industry, *International Journal of Advanced Manufacturing Technology* **60**(5-8) : 787–797.

Frazelle, E. H. (2002). *World-Class Warehousing and Material Handling*, McGraw-Hill, New-York.

Frazelle, E. H. & Sharp, G. P. (1989). Correlated assignment strategy can improve any order-picking operation, *Industrial Engineering* **21** : 33–37.

Glover, F. & Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Norwell, MA.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Boston, MA, USA.

Homem de Mello, T. (2007). A study on the cross-entropy method for rare event probability estimation, *INFORMS Journal on Computing* **19**(3) : 381–394.

Hsieh, L. & Tsai, L. (2005). The optimum design of a warehouse system on order picking efficiency, *International Journal of Advanced Manufacturing Technology* **28**(5-6) : 626–637.

Laporte, G. (1992). The traveling salesman problem : An overview of exact and approximate algorithms, *European Journal of Operational Research* **59** : 231–247.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. & Shmoys, D. B. (1985). *The traveling salesman problem. A guided tour of combinatorial optimisation*, John Wiley & Sons, Chichester.

Merkuryev, Y., Merkuryeva, G. & Burinskiene, A. (2009). *Warehouse Order Picking Process*, Springer-Verlag, London, chapter Simulation-Based studies in Logistics : Education and Applied Research, pp. 147–165.

Muppani, V. & Adil, G. (2008). Efficient formation of storage classes for warehouse storage location assignment : A simulated annealing approach, *International Journal Management Science* **36** : 609–618.

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events, *European Journal of Operational Research* **99** : 89–112.

Rubinstein, R. Y. (1999). The cross-entropy method for combinatorial and continuous optimization, *Methodology and Computing in Applied Probability* **2** : 127–190.

Rubinstein, R. Y. (2001a). Combinatorial optimization, cross-entropy, ants and rare events, *in* S. U. Pardalos & P. M. (eds), *Stochastic Optimization : Algorithms and Applications*, Kluwer, pp. 304–358.

Rubinstein, R. Y. (2001b). Combinatorial optimization via cross-entropy, *in* S. G. Harris & C. (eds), *Encyclopedia of Operations Research and Management Sciences*, Kluwer, pp. 102–106.

Rubinstein, R. Y. & Kroese, D. P. (2004). *The Cross-Entropy Method : A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning*, Springer Verlag.

Shi, L. & Olafsson, S. (2000). Nested partitions method for stochastic optimization, *Methodology and Computing in Applied Probability* **2**(3) : 271–291.

Ten Hompel, M. & Schmidt, T. (2007). *Warehouse Management : Automation and Organisation of Warehouse and Order Picking Systems*, Springer, Berlin, Germany.

Tompkins, J., White, J., Bozer, Y., Frazelle, E. & Tanchoco, J. (2003). *Facilities Planning*, John Wiley & Sons, New-Jersey.