

Algorithme d'approximation pour un problème d'affectation d'une commande multiproduit multifournisseur

Azeddine CHEREF, Christian ARTIGUES,
Emmanuel HÉBRARD, Pierre LOPEZ
Sandra U. NGUEVEU

Vincent CHARVILLAT

Yves Régent

CNRS, LAAS-CNRS

Université de Toulouse, INSA, INP, France
cheref@laas.fr, artigues@laas.fr, hebrard@laas.fr
lopez@laas.fr, ngueveu@laas.fr

Université de Toulouse

IRIT, INP, France
Vincent.Charvillat@enseeiht.fr

Devatics

Toulouse, France
yregent@devatics.com

RÉSUMÉ : Dans le contexte de la vente en ligne, étant donné une commande constituée d'un ensemble de produits et un ensemble de magasins, le problème que nous présentons dans ce papier consiste à répartir la commande sur un ensemble de magasins de sorte à minimiser le coût de la commande. Ce problème se retrouve dans plusieurs contextes et comporte beaucoup de similitudes avec le problème de localisation d'entrepôts. Nous traitons dans ce papier deux cas particuliers et pour chacun d'eux un algorithme de résolution est proposé. Dans le premier cas, nous considérons que les produits sont disponibles dans les magasins en quantité illimitée. Dans le deuxième cas, les quantités sont considérées limitées. Le problème étant une généralisation du problème couverture par ensembles, il est par conséquent NP-difficile. Ainsi, une heuristique avec garantie de performance basée sur l'approche d'arrondi en programmation linéaire est proposée pour chacun des cas traités.

MOTS-CLÉS : Approximation, Problème de localisation, Arrondi en programmation linéaire.

1 INTRODUCTION

Dans le contexte de la vente en ligne, une commande d'un client peut être constituée de plusieurs produits en quantités différentes. De plus, les produits commandés sont vendus par différents magasins et chacun d'eux dispose d'une certaine quantité pour chaque produit. L'objectif pour le client est d'acheter tous les produits de la commande en minimisant le prix global. Ce prix dépend du coût des produits ainsi que du coût d'acheminement depuis les magasins. En effet, les coûts des produits peuvent être différents selon les magasins sélectionnés et les coûts de transports dépendent également des magasins.

Le problème considéré dans ce papier consiste alors à sélectionner un ensemble de magasins de sorte à satisfaire la commande du client au meilleur coût. Ce problème comporte beaucoup de similitudes avec le problème de localisation d'entrepôts (*Facility Location*) et peut être considéré comme une extension de celui-ci. En effet, le problème de localisation d'entrepôts consiste à déterminer un ensemble d'emplacements afin d'implanter des installations (voir (Melo, Nickel & da Gama 2009) et (Daskin, Snyder & Berger 2005) pour plus de détails). Le problème général de localisation d'entrepôts standard ne prend pas en compte les capacités ainsi que les caractéris-

tiques des installations potentielles (capacités, types de produits, etc). Ce problème a été étendu et des capacités sont prises en compte dans le choix des implantations ainsi que des caractéristiques spécifiques à chaque emplacement. Ainsi, dans le problème étudié dans ce papier, les commandes du client et les magasins correspondent respectivement aux installations et aux magasins. Cependant, ce problème diffère du problème de localisation d'entrepôts car les magasins ainsi que les produits disponibles dans chacun d'eux sont connus.

Le problème que nous traitons étant NP-difficile, nous nous intéressons dans ces travaux aux algorithmes d'approximations, i.e., des heuristiques avec une garantie sur la qualité de la solution et le temps d'exécution. Une attention particulière est portée au problème de localisation d'entrepôts dans la littérature sur les algorithmes d'approximation. Shmoys et al dans (Shmoys, Tardos & Aardal 1997) proposent le premier algorithme pour ce problème avec un facteur d'approximation constant. Ces travaux sont étendus au cas multiproduit par (Ravi & Sinha 2010). Ainsi, ces auteurs étudient le problème de localisation d'entrepôts multiproduits et où le coût d'ouverture d'un entrepôt comprend un coût fixe et un coût variable qui dépend des produits disponibles dans cet entrepôt. Nos travaux s'inspirent de ces deux papiers

et nous utilisons l'approche d'arrondi en programmation linéaire (*LP-rounding*) introduite par (Lin & Vitter 1992) afin d'approcher la solution. Néanmoins, nous invitons les lecteurs à se référer au livre de Williamson et Shmoys (Williamson & Shmoys 2011) ainsi qu'aux travaux suivants (Chudak 1998), (Guha & Khuller 1999) et (An, Singh & Svensson 2017) pour plus de détails sur les techniques d'approximation pour le problème de localisation d'entrepôts.

2 DESCRIPTION DU PROBLEME

Afin de poser la formulation du problème, nous introduisons les notations suivantes. Soit $D = \{1, \dots, n\}$ l'ensemble des produits demandés et pour chaque produit $j \in D$, on note par d_j la quantité demandée. L'ensemble de magasins est noté $M = \{1, \dots, t\}$ et pour chaque magasin $i \in M$, on note par $P_i \subseteq D$ l'ensemble des produits disponibles dans le magasin i . Les stocks des magasins étant connus, on note par u_{ij} la quantité de produits j disponible dans le magasin i . Nous supposons sans perte de généralité que la quantité demandée par le client est disponible dans les magasins (i.e., $\sum_{i \in M} u_{ij} \geq d_j, \forall j \in D$). En effet, si un produit $j \in D$ n'est pas disponible en quantité suffisante, on pose $d_j = \sum_{i \in M} u_{ij}$. L'objectif considéré consiste à satisfaire la demande du client à coût minimum. Les coûts pris en compte dans notre modèle représentent les coûts des produits ainsi que les coûts de transport. Le coût d'un produit j dans le magasin i est noté c_{ij} et le coût de transport à partir d'un magasin i est noté f_i . Notez que ce coût ne dépend pas de la quantité transportée.

3 FORMULATION DU PROBLEME

Nous introduisons un modèle de programmation linéaire en nombres entiers pour le problème. On commence par présenter les variables utilisées.

- $g_{ij} \in \{0, \dots, d_j\}$: indique la proportion de la demande d_j satisfaite par le magasin i ,
- $z_i \in \{0, 1\}$: indique si le magasin i est sélectionné.

$$\min \sum_{i \in M} f_i z_i + \sum_{i \in M} \sum_{j \in P_i} c_{ij} g_{ij} \quad (1)$$

$$\sum_{i \in M: j \in P_i} g_{ij} = d_j, \quad \forall j \in D \quad (2)$$

$$g_{ij} \leq u_{ij} z_i, \quad \forall i \in M, \forall j \in P_i \quad (3)$$

$$g_{ij} \in \{0, \dots, d_j\}, \quad \forall i \in M, \forall j \in P_i \quad (4)$$

$$z_i \in \{0, 1\}, \quad \forall i \in M \quad (5)$$

Les contraintes (2) font en sorte que chaque commande soit satisfaite. Les contraintes (3) garantis-

sent la faisabilité de la solution en tenant compte des quantités disponibles dans chaque magasin. Cette contrainte nous assure également que lorsque un produit $j \in D$ est transféré du magasin $i \in M$ alors le magasin i doit être sélectionné. La fonction objectif (1) minimise le coût total composé de coûts fixes qui correspondent aux magasins sélectionnés et des coûts de transport.

4 CAS DE LA CAPACITÉ ILLIMITÉE

Dans des applications réelles, un magasin peut être spécialisé dans une catégorie de produits et ainsi disposer d'une grande quantité de produits de cette catégorie. Nous présentons dans cette section une heuristique avec garantie de performance pour le problème dans lequel les produits sont supposés disponibles en quantité illimitée dans les magasins. Ainsi la quantité u_{ij} de produits $j \in D$ disponibles dans le magasin P_i vérifie $u_{ij} = +\infty$ si $j \in P_i$, 0 sinon. Dans ce cas particulier et afin de formuler le problème, nous utilisons les variables suivantes:

- $x_{ij} \in \{0, 1\}$: indique si commande j est satisfaite par le magasin i ,
- $y_i \in \{0, 1\}$: indique si le magasin i est sélectionné.

Il faut noter que la quantité transportée pour chaque produit n'est pas prise en compte. En effet, lorsqu'un produit est disponible dans un magasin, celui-ci est disponible en quantité illimitée. Par conséquent, dans une solution optimale, une commande $j \in D$ est satisfaite par un seul magasin $i \in M$ tel que $j \in P_i$.

$$\min \sum_{i \in M} f_i y_i + \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x_{ij} \quad (6)$$

$$\sum_{i \in M: j \in P_i} x_{ij} = 1, \quad \forall j \in D \quad (7)$$

$$x_{ij} \leq y_i, \quad \forall i \in M, \forall j \in D \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in M, \forall j \in D \quad (9)$$

$$y_i \in \{0, 1\}, \quad \forall i \in M \quad (10)$$

4.1 Méthode de résolution

La résolution du problème étant NP-difficile, nous utilisons la programmation linéaire afin d'approcher la solution optimale du problème à variables entières présenté ci-dessus. L'algorithme suivant retourne une solution réalisable au problème et garantit un ratio de performance.

1. Résoudre la relaxation linéaire de la formulation présentée ci-dessus (contraintes (6) - (10)). Soit (x', y') la solution (fractionnaire) obtenue.

2. En utilisant les techniques de filtrage (“*filtering*”), trouver une nouvelle solution fractionnaire. Soit (x'', y'') la solution obtenue (voir section 4.1.1).
3. Arrondir la solution fractionnaire (x'', y'') afin d’obtenir une solution entière notée x''', y''' . A cet effet, nous avons utilisé le problème de multicouverture par multiensembles (*Multiset Multicover*) (voir section 4.1.2).

A noter que dans (Ravi & Sinha 2010), les auteurs étudient une extension du problème de localisation d’entrepôts et utilisent le problème de couverture par ensembles (*Set Cover*) dans le même but. Enfin, nous obtenons un ratio d’approximation égal à $\log k + 2$, où k est le nombre maximum de produit dans tout les magasins.

4.1.1 Filtrage

Etant donné la solution optimale de la relaxation du problème noté (x', y') , cette étape consiste à maintenir un sous ensemble de variables non nulles x' dans une valeur positive et de mettre les autres variables à zero. Nous reprenons dans cette partie le raisonnement utilisé par (Ravi & Sinha 2010, Shmoys et al. 1997, Lin & Vitter 1992). En effet, le problème que nous traitons constitue un cas particulier du problème traité dans (Ravi & Sinha 2010). Ainsi, la solution (x'', y'') est atteinte en suivant les étapes suivantes.

- Soit X_j l’ensemble des magasins $i \in M$ tels que $x'_{ij} > 0$ soient triés par ordre croissant des coûts c_{ij} .
- Soit $0 < \alpha < 1$
- Etant donné α , trouver pour tout $j \in D$ le plus petit coût $c_j(\alpha)$ tel que $\sum_{i \in X_j: c_{ij} \leq c_j(\alpha)} x'_{ij} \geq \alpha$
- Soit pour tout $j \in D$, $\alpha_j = \sum_{i \in X_j: c_{ij} \leq c_j(\alpha)} x'_{ij}$
- Créer une nouvelle solution fractionnaire (x'', y'') où x'' et y'' prennent les valeurs suivantes :

- $x''_{ij} = x'_{ij}/\alpha_j$ si $c_{ij} \leq c_j(\alpha)$, 0 sinon.
- $y''_i = \min\{1, y'_i/\alpha\}$

Lemme 1. La solution (x'', y'') est réalisable pour la relaxation du problème défini par les contraintes (6 - 10) et pour tout $j \in D$, $c_j(\alpha) \leq \frac{1}{1-\alpha} \sum_{i \in M} c_{ij} x'_{ij}$.
Preuve. Pour tout $j \in D$, on a.

$$\begin{aligned} \sum_{i \in M} c_{ij} x'_{ij} &\geq \sum_{i \in M: c_{ij} \geq c_j(\alpha)} c_{ij} x'_{ij} \\ &\geq c_j(\alpha) \sum_{i \in M: c_{ij} \geq c_j(\alpha)} x'_{ij} \end{aligned} \quad (11)$$

D’après la définition de $c_j(\alpha)$,

$$\sum_{i \in M: c_{ij} < c_j(\alpha)} x'_{ij} \leq \alpha$$

Alors,

$$\sum_{i \in M} x'_{ij} = 1 \Rightarrow \sum_{i \in M: c_{ij} \geq c_j(\alpha)} x'_{ij} \geq 1 - \alpha$$

En remplaçant dans (11), nous obtenons $c_j(\alpha) \leq \frac{1}{1-\alpha} \sum_{i \in M} c_{ij} x'_{ij}$. \square

Noter qu’un résultat similaire au Lemme exposé est proposé dans (Shmoys et al. 1997).

4.1.2 Arrondi

Afin d’obtenir une solution entière réalisable au problème défini par les contraintes 6–10, Nous arrondissons dans cette section la solution (x'', y'') . De manière similaire à (Ravi & Sinha 2010), nous utilisons le problème de couverture par ensembles qui consiste à trouver le sous-ensemble de magasins permettant de couvrir la demande et dont le coût total d’ouverture est minimal. Ainsi, la solution entière obtenue notée (x''', y''') est obtenue en suivant les étapes suivantes.

Nous commençons par construire une instance du problème de couverture de l’ensemble des demandes $D = 1, \dots, n$ par les ensembles $\{s_i \mid 1 \leq i \leq t\}$ de produits que peuvent fournir chaque magasin i .

On utilise les notations suivantes :

- Un ensemble de magasins $S = \{i \in M : y''_i > 0\}$
- L’ensemble des produits disponibles dans chaque magasin : $s_i = \{j \in P_i : x''_{ij} > 0\}$
- Un coût f_i pour chaque magasin $i \in S$
- Soit k une borne supérieure du nombre de produits disponibles dans chaque magasin, i.e., $k = \max_{i \in S} \{|s_i|\}$. Clairement, k est borné par n .

L’instance étant construite, soit h_i une variable telle que $h_i = 1$ si le magasin $i \in S$ est sélectionné, égale à 0 sinon. La formulation du problème de couverture par ensembles est ainsi définie comme suit.

$$\min \sum_{i \in S} f_i h_i \quad (12)$$

$$\sum_{i \in S: j \in s_i} h_i \geq 1, \quad \forall j \in D \quad (13)$$

$$h_i \in \{0, 1\}, \quad \forall i \in S, \forall j \in s_i \quad (14)$$

Les contraintes (13) garantissent que chaque commande $j \in D$ est affectée à au moins un magasin

et l'objectif est de minimiser coût total des magasins sélectionnés. Ainsi ce problème de couverture par ensembles peut être résolu avec un ratio d'au plus $\log(k+1)$ (voir (Vazirani 2001)). A noter que cette valeur représente le ratio entre la solution optimale et la solution relaxée du problème de couverture par ensembles. Etant donné une couverture par ensembles (solution des contraintes (12–14)), nous construisons la solution (x''', y''') de la façon suivante :

- La commande $j \in D$ est affectée au magasin $i' \in S$ tel que $i' = \operatorname{argmin}_{i \in S} \{c_{ij} : z_i = 1, j \in s_i\}$.
- $x''_{ij} = 1$ si $i = i'$ et 0 sinon.
- $y'''_i = 1$ si $z_i = 1$ et 0 sinon.

Lemme 2. La solution entière (x''', y''') est réalisable pour le problème défini par les contraintes (6 - 10).

Preuve. Nous pouvons voir que pour chaque produit $j \in D$, il existe un magasin $i \in M$ pour lequel la variable $x''_{ij} > 0$ car $\sum_{i \in M: j \in P_i} x''_{ij} = 1$. Ainsi, chaque produit est disponible dans au moins un magasin s_i . Par conséquent, la solution (x''', y''') est réalisable car chaque commande est affectée à exactement un magasin. \square

4.1.3 Facteur d'approximation

Dans cette section, un facteur d'approximation est calculé. Celui-ci représente le quotient entre les valeurs de l'objectif correspondant aux solutions (x''', y''') et (x', y') . Afin de calculer ce facteur d'approximation, nous calculons séparément le facteur d'approximation sur la partie de l'objectif qui correspond aux coûts de transport et celui correspondant ou coûts des produits. Ainsi nous obtenons.

- Facteur d'approximation sur la partie de l'objectif qui correspond aux coûts de transport dans la fonction objectif :

$$\sum_{i \in M} f_i y'_i \leq \frac{\log k + 1}{\alpha} \sum_{i \in M} f_i y''_i$$

Preuve. Etant donnée l'instance utilisée pour le problème de couverture par ensembles et le fait que le facteur d'approximation pour le problème de couverture par ensembles est au plus égal à $\log(k+1)$, on a.

$$\sum_{i \in M} f_i y''_i \leq (\log k + 1) \sum_{i \in M} f_i y'_i$$

Du fait que $y''_i = \min\{1, y'_i/\alpha\}$, on a.

$$\sum_{i \in M} f_i y''_i \leq \frac{1}{\alpha} \sum_{i \in M} f_i y'_i$$

Alors.

$$\sum_{i \in M} f_i y'''_i \leq \frac{\log k + 1}{\alpha} \sum_{i \in M} f_i y'_i$$

\square

- Facteur d'approximation sur la partie de l'objectif qui correspond aux coûts des produits dans la fonction objectif :

$$\sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'''_{ij} \leq \frac{1}{1-\alpha} \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'_{ij}$$

Preuve. Etant donné que pour tout $j \in D$, $\sum_{i \in M} c_{ij} x'''_{ij} \leq c_j(\alpha)$, on a.

$$\begin{aligned} \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'''_{ij} &\leq \sum_{j \in D} d_j c_j(\alpha) \\ &\leq \frac{1}{1-\alpha} \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'_{ij} \end{aligned}$$

\square

- Alors pour la fonction objectif complète nous obtenons le facteur d'approximation suivant.

$$\begin{aligned} \sum_{i \in M} f_i y'''_i + \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'''_{ij} &\leq \frac{\log k + 1}{\alpha} \sum_{i \in M} f_i y'_i + \frac{1}{1-\alpha} \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'_{ij} \\ &\leq \max\left\{\frac{\log k + 1}{\alpha}, \frac{1}{1-\alpha}\right\} \\ &\quad \left(\sum_{i \in M} f_i y'_i + \sum_{i \in M} \sum_{j \in D} d_j c_{ij} x'_{ij}\right) \end{aligned}$$

Ainsi, en posant $\alpha = \frac{\log k + 1}{\log k + 2}$ nous obtenons un facteur d'approximation égal à $\log k + 2$.

On peut noter que ce facteur d'approximation est inférieur à celui présenté dans (Ravi & Sinha 2010) (égal à $\log k + 4$). C'est dû au fait que le problème que nous avons traité peut être considéré comme étant un cas particulier du problème étudié par ces auteurs.

5 CAS DE LA CAPACITÉ LIMITÉE

Nous proposons dans cette section une heuristique avec garantie de performance pour le cas où les produits sont disponibles dans les magasins en quantités limités. La programmation linéaire est également utilisée afin d'approcher la solution optimale du problème à variables entières. Notons que du fait des capacités sur les magasins, il est impossible de trouver un ratio de performance basé sur la relaxation linéaire du problème (voir (Shmoys et al. 1997) pour plus de détails). Par conséquent, nous proposons une (ρ, ρ') -approximation qui trouve en temps polynomial une solution avec un facteur ρ du coût de l'optimum mais où les capacités sont étendues d'un facteur ρ' .

Un aperçu de l'algorithme utilisé est présenté dans ce qui suit.

5.1 Méthode de résolution

L'algorithme suivant est utilisé afin de résoudre le problème pour le cas où les quantités disponibles dans chaque magasin sont limitées. Cet algorithme est similaire à celui utilisé pour le cas où les capacités sont considérées illimitées.

1. Résoudre la relaxation linéaire du problème défini par les contraintes (1) - (5). Soit (g', z') la solution (fractionnaire) obtenue (voir section 5.1.1)..
2. Trouver une nouvelle solution fractionnaire. Soit (g'', z'') la solution obtenue (voir section 5.1.2).
3. Trouver une solution réalisable (entière). Soit (g''', z''') la solution obtenue (voir section 5.1.3).

Finalement, nous obtenons un facteur d'approximation égal à $(\log k' + 2, \frac{\log k' + 2}{\log k' + 1})$ où k' est le nombre maximum de produits dans tout les magasins.

5.1.1 Formulation du problème relaxé

Nous présentons un modèle de programmation linéaire pour le problème à variables relaxées. Les variables suivantes utilisées.

- $g'_{ij} \in [0, 1]$: indique la proportion de la demande d_j satisfaite par le magasin i ,
- $z'_i \in [0, 1]$: indique si le magasin i est sélectionné.

$$\min \sum_{i \in M} f_i z'_i + \sum_{i \in M} \sum_{j \in D} d_j c_{ij} g'_{ij} \quad (15)$$

$$\sum_{i \in M} g'_{ij} = 1, \quad \forall j \in D \quad (16)$$

$$d_j g'_{ij} \leq u_{ij} z'_i, \quad \forall i \in M, \forall j \in D \quad (17)$$

$$0 \leq g'_{ij} \leq 1, \quad \forall i \in M, \forall j \in D \quad (18)$$

$$0 \leq z'_i \leq 1, \quad \forall i \in M \quad (19)$$

Cette formulation représente une relaxation du la formulation définie par les contraintes (1)–(5)). En effet, un remplacement de g_{ij} par $d_j g'_{ij}$, $\forall i \in M, \forall j \in D$ rend les deux formulations identiques.

On peut remarquer qu'en se basant sur la programmation linéaire, il est impossible de trouver un algorithme d'approximation pour le problème considéré à capacité limitée. nous invitons les lecteurs à se référer au papier de (Shmoys et al. 1997) pour plus de détails.

5.1.2 Filtrage

Cette étape est semblable à celle présentée en section 4.1.1 pour le cas où les quantités dans les magasins sont considérés limitées. Ainsi, les différentes étapes utilisées afin d'obtenir la nouvelle solution fractionnaire (g'', z'') ne sont pas exposées. Nous rappelons cependant certaines valeurs.

- $g''_{ij} = g'_{ij}/\alpha_j$ if $c_{ij} \leq c_j(\alpha)$ et 0 sinon.
- $z''_i = \min\{1, y'_i/\alpha\}$
- $\forall j \in D, c_j(\alpha) \leq \frac{1}{1-\alpha} \sum_{i \in M} c_{ij} x'_{ij}$.

La solution ainsi obtenue notée (g'', z'') n'est pas forcément réalisable pour les contraintes (1–4). En effet, soit une solution dans laquelle il existe $i \in M$ et $j \in D$ tel que $d_j g'_{ij} = u_{ij}$. Dans ce cas, $z'_i = 1$. En remplaçant (g', z') par (g'', z'') , cette contrainte n'est pas respectée (cette contrainte correspond à la contrainte (17) dans la formulation relaxée). Nous étendons par conséquent la quantité de produits disponibles dans chaque magasin afin d'avoir une solution (g'', z'') réalisable. Soit u'_{ij} la nouvelle quantité de produit j disponible dans le magasin i telle que $u'_{ij} = \frac{1}{\alpha} u_{ij}$.

5.1.3 Arrondi

Dans cette section, nous arrondissons la solution (g'', z'') afin d'obtenir une solution entière notée (g''', z''') qui soit réalisable sur les contraintes (1) - (5)). A cet effet, nous utilisons le problème de multicouverture par multiensembles (*Multiset Multicover*) comme sous problème du cas que nous traitons.

Problème de multicouverture par multiensembles : Ce problème est une généralisation du problème de couverture par ensembles dans lequel chaque élément e est associé à un nombre de couvertures r_e indiquant le nombre de fois ou celui-ci doit être couvert par toute couverture réalisable. En plus du nombre de couvertures, chaque élément apparaît dans un ensemble S avec une multiplicité de $m(S, e)$.

La solution entière (g''', z''') est alors obtenue par les étapes suivantes.

Nous commençons par construire une instance du problème de multicouverture par multiensembles de la manière suivante :

- Un ensemble de n membres $D = 1, \dots, n$ représentant les demandes
- Une demande de couverture $r_j = d_j$ pour chaque produit $j \in D$

- Un ensemble de magasins $S = \{i : i \in M, z_i'' > 0\}$
- L'ensemble des produits disponibles dans chaque magasin : $s_i = \{j \in P_i : x_{ij}'' > 0\}$
- La multiplicité $m(s_i, j) = u'_{ij}$ pour chaque produit $j \in s_i$ qui correspond au quantités disponibles
- Un coût f_i pour chaque magasin $i \in S$
- Soit k' une borne supérieure du nombre de produits disponibles dans chaque magasin, i.e., $k' = \max_{i \in S} \{\sum_{j \in s_i} u'_{ij}\}$.

Etant donné l'instance, on note par h'_i une variable qui vaut 1 si le magasin $i \in S$ est sélectionné et 0 sinon. Le problème de multicouverture par multi-ensembles peut être formulé comme suit.

$$\begin{aligned} \min \sum_{i \in S} f_i h'_i \\ \sum_{i \in S: j \in s_i} m(s_i, j) h'_i \geq r_j, \quad \forall j \in D \quad (20) \\ h'_i \in \{0, 1\}, \quad \forall i \in S, \forall j \in s_i \end{aligned}$$

Les contraintes (20) garantissent la satisfaction des demandes et l'objectif est de minimiser le coût total induit par les magasins sélectionnés. Ce problème peut être résolu en temps polynomial avec un ratio d'au plus $\log(k+1)$ par rapport à la solution relaxée (voir (Rajagopalan & Vazirani 1998) pour plus de détails). A noter que ce ratio d'approximation est logarithmique sur la taille du plus grand ensemble et non sur n . Dans notre cas, la taille du plus grand ensemble correspond au nombre maximum de produits disponibles dans un magasin.

Etant donné une solution au problème de multicouverture par multiensembles présenté ci-dessus, la solution entière (g''', z''') est construite à l'aide de l'algorithme suivant.

- $z_i''' = 1$ si $h'_i = 1$ et 0 sinon.
- Pour chaque commande $j \in D$,
 - Soit $R_j = \{i : h'_i = 1, j \in s_i\}$.
 - Soit $d'_j = d_j$,
 - Pour $l = 1$ à $|R_j|$, faire
 - * Si $d'_j \leq 0$,
 - $x'''_{lj} = \min\{d'_j, u'_{lj}\}/d_j$,
 - $d'_j = d_j - u'_{lj}$
 - * Sinon $x'''_{lj} = 0$

Il faut noter qu'un autre algorithme que celui présenté ci-dessus ne peut améliorer le facteur

d'approximation théorique. Enfin, la solution (g''', z''') est réalisable pour le problème défini par les contraintes (15-19). Soit $\forall i \in M, \forall j \in D, g_{ij} = d_j g'''_{ij}$ et $\forall i \in M, z_i = z_i'''$. La solution (g, z) est réalisable pour le problème défini par les contraintes (1-5).

5.1.4 Facteur d'approximation

Le facteur d'approximation est obtenu de manière similaire à celle utilisée pour le cas où les quantités de produits disponibles dans les magasins sont illimitées. Ainsi, pour $\alpha = \frac{\log k+1}{\log k+2}$, nous obtenons un facteur d'approximation égal à $(\log k + 2, \frac{\log k+2}{\log k+1})$.

6 Conclusion

Nous avons proposé des algorithmes d'approximation avec garantie de performance pour deux cas particuliers d'un problème d'affectation d'une commande multiproduit multifournisseur dans le cadre vente en ligne. Par la suite nous proposerons une méthode exacte et des heuristiques basées sur la programmation linéaire en nombres entiers pour la résolution pratique du cas général. Ces travaux sont réalisés dans le cadre du projet "One Stock Performance", financé par le programme "Easynov" (FEDER / REGION Midi-Pyrénées Languedoc Roussillon) en partenariat avec la société DEVATICS.

References

- An, H.-C., Singh, M. & Svensson, O. (2017). Lp-based algorithms for capacitated facility location, *SIAM Journal on Computing* **46**(1): 272–306.
- Chudak, F. A. (1998). *Improved Approximation Algorithms for Uncapacitated Facility Location*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 180–194.
- Daskin, M. S., Snyder, L. V. & Berger, R. T. (2005). *Facility Location in Supply Chain Design*, Springer US, Boston, MA, pp. 39–65.
- Guha, S. & Khuller, S. (1999). Greedy strikes back: Improved facility location algorithms, *Journal of Algorithms* **31**(1): 228 – 248.
- Lin, J.-H. & Vitter, J. S. (1992). e-approximations with minimum packing constraint violation (extended abstract), *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, ACM, New York, NY, USA, pp. 771–782.
- Melo, M., Nickel, S. & da Gama, F. S. (2009). Facility location and supply chain management a re-

view, *European Journal of Operational Research* **196**(2): 401 – 412.

Rajagopalan, S. & Vazirani, V. V. (1998). Primal-dual rnc approximation algorithms for set cover and covering integer programs, *SIAM Journal on Computing* **28**(2): 525–540.

Ravi, R. & Sinha, A. (2010). Approximation algorithms for multicommodity facility location problems, *SIAM Journal on Discrete Mathematics* **24**(2): 538–551.

Shmoys, D. B., Tardos, E. & Aardal, K. (1997). Approximation algorithms for facility location problems (extended abstract), *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, ACM, New York, NY, USA, pp. 265–274.

Vazirani, V. V. (2001). *Approximation Algorithms*, Springer-Verlag New York, Inc., New York, NY, USA.

Williamson, D. P. & Shmoys, D. B. (2011). *The design of approximation algorithms*, Cambridge university press.