LA LOCALISATION DES ENTREPOTS

Léon KAUFMAN
Vrije Universiteit te Brussel

1. Introduction.

1.1. Localisation des entrepôts.

Le problème de l'établissement d'un réseau d'entrepôts se pose lorsqu'une entreprise doit satisfaire les demandes d'un grand nombre de clients géographiquement dispersés. L'entreprise peut expédier les produits demandés directement des usines aux clients ou bien utiliser des entrepôts. Dans ce dernier cas, des expéditions en gros peuvent être faites des usines aux entrepôts, par des moyens de transport économiques (par exemple par chemin de fer ou par gros camions). L'usage d'entrepôts permet donc de diminuer les coûts de transport, mais entraîne des dépenses d'établissement et de fonctionnement.

Le problème est de déterminer le nombre, les tailles et les emplacements des entrepôts du réseau projeté de manière à minimiser les frais d'entreposage et de transport.

1.2. Localisation des centres de service.

On rencontre dans le domaine public des problèmes de localisation de centres de service (par exemple écoles ou hôpitaux) semblables au problème de la localisation des entrepôts; les coûts de transport sont généralement à charge des usagers, l'objectif que l'on cherche à maximiser est une fonction de leur satisfaction, exprimée par exemple en termes monétaires ou en termes de kilomètres parcourus.

Le budget et par conséquent le nombre maximum de centres de service sont souvent décidés à priori par l'autorité publique et non par les planificateurs.

Le problème est de déterminer les emplacements et les tailles d'un nombre fixé de centres de service de manière à maximiser l'utilité des usagers.

1.3. Méthodes de résolution.

Depuis le XIX[e] siècle existent des modèles pour la localisation d'un centre de service sur un plan : ces modèles utilisent des techniques de calcul différentiel.

Le problème de la localisation simultanée de plusieurs centres de service mène rapidement à des systèmes d'équations fort complexes. De plus, il est clair qu'un centre ne peut en pratique être localisé n'importe où.

Grâce aux techniques de la recherche opérationnelle, des modèles plus réalistes et permettant d'aborder des problèmes plus complexes ont été élaborés depuis une dizaine d'années. Dans ces modèles, on recherche la localisation optimale d'un nombre fixé ou même inconnu de services parmi un ensemble fini d'emplacements préalablement choisis.

Des procédures d'optimisation par séparation permettent d'obtenir la solution optimale pour ces problèmes. Depuis la parution, en 1965, de l'algorithme de EFROYMSON et RAY [7], ces modèles ont été utilisés en pratique par de nombreuses entreprises pour résoudre leurs problèmes de localisation.

Comme les modèles ne donnent qu'une approximation de la situation réelle, les résultats obtenus doivent être considérés aussi comme approchant seulement l'optimum réel.

1.4. Problèmes étudiés.

Dans la suite de cet article, on étudie principalement un modèle pour la localisation des entrepôts dû à M. L. BALINSKI [2]. On expose l'algorithme exact d'Efroymson et Ray et six algorithmes heuristiques. Ensuite, trois séries d'expériences permettent de comparer les performances de ces algorithmes.

Enfin quelques extensions du modèle de Balinski sont brièvement discutées.

2. Un modèle mathématique.

2.1. Enoncé du modèle de Balinski.

Soient m emplacements, repérés par l'indice i, où peuvent être établis des entrepôts.

L'établissement d'un entrepôt est noté à l'aide d'une variable indicatrice y_i :

$$y_i \in \{0, 1\} \qquad (i = 1, 2, ..., m) \tag{1}$$

y_i prend la valeur 1 si un entrepôt est établi à l'emplacement i et sinon la valeur 0.

Soient n clients, repérés par l'indice j, desquels émanent des demandes d_j . S'il existe un entrepôt en i, la demande d_j peut être satisfaite à partir de l'entrepôt i dans la proportion x_{ij} ; s'il n'y a pas d'entrepôt en i, aucune demande ne peut être satisfaite à partir de i et $x_{ij} = 0$ pour tout j :

$$0 \leqslant x_{ij} \leqslant y_i \leqslant 1 \qquad (i = 1, 2, ..., m), (j = 1, 2, ..., n) \tag{2}$$

Toutes les demandes doivent être satisfaites.

$$\sum_i x_{ij} = 1 \qquad (i = 1, 2, ..., m), (j = 1, 2, ..., n) \tag{3}$$

Les données numériques sont les coûts fixes f_i pour l'établissement d'un entrepôt à l'emplacement i et les coûts c_{ij} pour satisfaire la demande d_j à partir de l'entrepôt établi en i.

Le but est de minimiser, sous les contraintes (1), (2) et (3), la fonction économique :

$$z = \sum_i \sum_j c_{ij}\, x_{ij} + \sum_i f_i\, y_i \qquad (i = 1, 2, ..., m), (j = 1, 2, ..., n) \tag{4}$$

Dans ce modèle, appelé M1 dans la suite, on fait l'hypothèse que les capacités des entrepôts sont illimitées (simple warehouse location problem). Dans la solution optimale, la demande de chaque client j sera entièrement satisfaite à partir de l'entrepôt k tel que :

$$c_{kj} = \min_i \{c_{ij} \mid y_i = 1\} \tag{5}$$

Si les coûts c_{ij} sont des fonctions linéaires des distances entre i et j, l'équation (5) signifie que la demande de chaque client est satisfaite à partir de l'entrepôt le plus proche. Il existe toujours une solution optimale telle qu'une variable x_{ij} par colonne prenne la valeur 1 et les autres variables la valeur 0 (en cas d'égalité de certains coefficients c_{ij}, d'autres types de solution existent également).

Le modèle mathématique M1 décrit par les équations (1), (2), (3) et (4) ci-dessus a été utilisé par Efroymson et Ray, par K. Spielberg [19], [20] et par divers autres auteurs [1], [10], [21], [22]. On peut substituer les termes généraux de « source » et « puits » à « entrepôts » et « client » dans l'énoncé. Cette terminologie permet d'appliquer le même modèle au problème de la détermination des emplacements et des tailles optimales d'un réseau d'usines pour minimiser les frais de production et de distribution.

2.2. Un algorithme exact.

Le problème décrit par le modèle M1 est un programme linéaire mixte et ne peut être résolu par l'algorithme du simplexe car les variables y_i pourraient prendre des valeurs fractionnaires dans la solution optimale. Les algorithmes exacts qui se sont montrés efficaces pour résoudre ce problème sont tous des procédures d'optimisation par séparation.

L'algorithme LOCA 1, utilisé dans les expériences décrites ci-dessous, est une version modifiée de celui de Efroymson et Ray; LOCA 1 est une procédure d'optimisation par séparation et évaluation séquentielle (P.S.E.S.) et non par séparation et évaluation progressive (P.S.E.P.). Ceci permet de diminuer fortement la capacité de mémoire requise et d'éviter des sous-routines de classement lors de la résolution sur ordinateur. La formulation de Efroymson et Ray est légèrement différente de celle employée par Balinski; les contraintes :

$$x_{ij} \leqslant y_i \tag{6}$$

sont remplacées par les contraintes :

$$\sum_j x_{ij} \leqslant n_i \, y_i \tag{7}$$

où n_i est le nombre de clients plus proches de i que d'un autre entrepôt ouvert. A l'étape initiale $n_i = n$ pour tout i ,mais lorsque des variables y_i sont fixées à 1, les n_i diminuent rapidement.

Le principe de séparation utilisé dans l'algorithme consiste à choisir une variable y_i à mettre soit à 0 soit à 1. A une itération K, on désigne par :

K_0 : l'ensemble des indices des variables Y_i mises à 0

K_1 : l'ensemble des indices des variables Y_i mises à 1 et

K_2 : l'ensemble des indices des variables Y_i libres, c'est-à-dire non encore fixées à 0 ou 1.

Le problème à une itération K se réduit à :

$$\min \left(\sum_{i \in K_2} f_i \, y_i + \sum_{i \in K_1} f_i + \sum_{i \in K_1 \cup K_2} \sum_j c_{ij} \, x_{ij} \right) \tag{8}$$

$$\sum_{i \in K_1 \cup K_2} x_{ij} = 1 \qquad j = 1, \ldots, n \tag{9}$$

$$\sum_{j=1}^{n} x_{ij} \leqslant n_i \, y_i \qquad i \in K_2 \tag{10}$$

$$0 \leqslant x_{ij} \qquad j = 1, \ldots, n, \; i \in K_2 \tag{11}$$

$$y_i \in \{0, 1\} \qquad i \in K_2 \tag{12}$$

LOCA 1 utilise un test d'optimalité et un test d'optimalité conditionnelle.

La valeur de la fonction d'évaluation employée pour le test d'optimalité à chaque itération est la solution du problème obtenu en remplaçant les contraintes (12) d'intégralité des variables y_i par :

$$0 \leqslant y_i \leqslant 1 \qquad i \in K_2 \qquad (12\text{ bis})$$

Cette valeur est facilement obtenue grâce au théorème suivant :

Théorème : Dans la solution optimale de (8), (9), (10), (11) et (12 bis) :

$$y_i = \frac{1}{n_i} \sum_j x_{ij} \qquad i \in K_2 \qquad (13)$$

Démonstration : La démonstration se fait par l'absurde. Supposons que dans la solution optimale existe un indice i appartenant à l'ensemble K_2 tel que :

$$y_i > \frac{1}{n_i} \sum_j x_{ij} \qquad (14)$$

la valeur de y_i peut être diminuée jusqu'à ce qu'on obtienne l'égalité tout en laissant les autres x_{ij} et y_i inchangés. Comme on suppose que les f_i sont strictement positifs, ceci diminue la valeur de la fonction économique; ce qui contredit l'optimalité. CQFD.

Pour obtenir la fonction d'évaluation, on remplace y_i par $\sum_j \frac{x_{ij}}{n_i}$ dans la fonction économique, ce qui donne le problème suivant :

$$\min \left(\sum_{i \in K_1} f_i + \sum_{i \in K_2} \sum_j \frac{f_i \, x_{ij}}{n_i} + \sum_{i \in K_1 \cup K_2} \sum_j c_{ij} \, x_{ij} \right) \qquad (15)$$

$$\sum_{i \in K_1 \cup K_2} x_{ij} = 1 \qquad j = 1, ..., n \qquad (16)$$

$$0 \leqslant x_{ij} \qquad i \in K_1 \cup K_2 , \; j = 1, ..., n \qquad (17)$$

En posant :

$$g_k = f_k \text{ pour } k \in K_2 \qquad (18)$$

$$g_k = 0 \text{ pour } k \in K_1$$

le problème devient :

$$\sum_{i \in K_1} f_i + \min \left(\sum_j \sum_{i \in K_1 \cup K_2} \left[\left(c_{ij} + \frac{g_i}{n_i} \right) x_{ij} \right] \right) \qquad (19)$$

$$\sum_{i \in K_1 \cup K_2} x_{ij} = 1 \qquad j = 1, ..., n \qquad (20)$$

$$0 \leqslant x_{ij} \qquad i \in K_1 \cup K_2 , \; j = 1, ..., n \qquad (21)$$

La solution optimale est donnée par :

$$x_{ij} = 1 \quad \text{si } c_{ij} + \frac{g_i}{n_i} = \min_{k \in K_1 \cup K_2} \left(c_{kj} + \frac{g_k}{n_k}\right) \tag{22}$$

$$x_{ij} = 0 \tag{23}$$

sinon on calcule, pour le test d'optimalité, la valeur de la fonction d'évaluation et on la compare avec la valeur de la meilleure solution déjà obtenue; deux cas sont possibles :

a) la fonction d'évaluation est plus grande :
alors l'ensemble ne contient pas de solution de coût inférieur à celui de la meilleure solution déjà obtenue et il est sondé;

b) la fonction d'évaluation est plus petite :
- K_2 est vide : l'ensemble comprend un seul élément et on a obtenu une solution dont la valeur est moindre que celle de la meilleure solution dejà obtenue par l'algorithme;
- K_2 n'est pas vide : l'ensemble n'est pas sondé et on continue avec les tests suivants.

Un test d'optimalité conditionnelle permet de voir si certaines variables doivent être mises à 0 pour que le sous-ensemble de solutions candidates considéré puisse contenir la solution optimale.

C'est le cas lorsque le gain maximum qui pourrait être réalisé sur les coûts de transport en ouvrant un entrepôt est plus petit que le coût fixe d'établissement. Pour exprimer ceci mathématiquement, on définit :

$$cm_j = \min_{k \in K_1} c_{kj} \tag{24}$$

La condition s'exprime alors :

$$\sum_j \max\,(0,\ cm_j - c_{ij}) < f_i \tag{25}$$

La sommation est faite sur les valeurs positives de $cm_j - c_{ij}$ parce que seuls les gains doivent être pris en compte.

Il n'y a pas de tests d'admissibilité car les contraintes peuvent toujours être vérifiées dès qu'un y_i est égal à 1.

La règle de choix de la variable utilisée pour la séparation consiste à mettre la variable y_i à 1 si :

$$\sum_j \max\,(0,\ cm_j - c_{ij}) - f_i = \max_{k \in K_2} \left[\sum_j \max\,(0,\ cm_j - c_{kj}) - f_k\right] \tag{26}$$

On met donc à 1 la variable qui entraîne la plus grande réduction des coûts.

2.3. Six algorithmes heuristiques.

Divers auteurs [8], [10], [11] ont proposé des algorithmes heuristiques pour le modèle M1 ou d'autres modèles proches de M1. Ces algorithmes permettent d'obtenir rapidement une bonne solution, qui n'est pas nécessairement optimale.

Six algorithmes heuristiques ont été utilisés dans les expériences. Les trois algorithmes ATTILA 1, BABEL 1 et BABILA 1 travaillent par modifications successives de composantes du vecteur y, une composante à la fois. Les valeurs des variables x_{ij} se déduisent immédiatement de celles des variables y_i.

1) ATTILA 1. Initialement on suppose qu'il existe un entrepôt en chaque emplacement ($y_i = 1, \forall i$). A la k^{me} itération, on calcule pour chaque emplacement où un entrepôt reste ouvert la variation de coût qui serait obtenue en fermant cet entrepôt. On calcule le minimum de ces variations de coût. Si ce minimum est négatif on ferme l'entrepôt correspondant et on passe à l'itération suivante. Si ce minimum est positif on a atteint un optimum local et on s'arrête.

2) BABEL 1. Initialement on suppose qu'il n'existe aucun entrepôt et que le coût de transport est un nombre fini très grand. A la k^{me} itération, on calcule pour chaque emplacement où il n'y a pas d'entrepôt la variation de coût qui serait obtenue en ouvrant un entrepôt en cet emplacement. On calcule le minimum de ces variations de coût. Si ce minimum est négatif on ouvre l'entrepôt correspondant et on passe à l'itération suivante. Si ce minimum est positif ou nul on a atteint un optimum local et on s'arrête.

3) BABILA 1. Initialement on part d'une solution quelconque, avec des entrepôts en certaines emplacements ou éventuellement en tous. A la k^{me} itération, on calcule pour chaque emplacement où un entrepôt est ouvert la variation de coût qui serait obtenue en le fermant et pour chaque emplacement où il n'y a pas d'entrepôt la variation de coût qui serait obtenue en y ouvrant un entrepôt. On calcule le minimum de ces variations de coût. Si ce minimum est négatif, ou ouvre (ou on ferme) l'entrepôt correspondant et on passe à l'itération suivante. Si ce minimum est positif ou nul on a atteint un optimum local et on s'arrête.

4) ATTILA 2, BABEL 2 et BABILA 2. Ces trois algorithmes sont des versions améliorées des algorithmes ATTILA 1, BABEL 1 et BABILA 1. Après l'obtention d'une bonne solution par une de ces méthodes, on examine si le coût peut être réduit par déplacement d'un entrepôt. A cet effet, à la k^{me} itération, on calcule pour tous les couples (i_1, i_2) d'emplacements, tel qu'il y ait un entrepôt en i_1 et pas d'entrepôt en i_2, la variation de coût qui serait obtenue en déplaçant l'entrepôt de l'emplacement i_1 à l'emplacement i_2. Si une variation de coût est négative on déplace l'entrepôt correspondant et on passe à l'itération suivante. Si toutes les variations de coût sont positives ou nulles, la solution obtenue ne peut plus être améliorée par déplacement d'un entrepôt à la fois et on s'arrête.

3. Expériences.

3.1. Un modèle numérique.

Pour tester ces algorithmes, nous considérons une entreprise fictive d'ameublement. Cette entreprise possède une seule fabrique et envisage l'établissement d'un réseau d'entrepôts. Le modèle tient compte des coûts de transport de la fabrique aux entrepôts, des entrepôts aux clients et des coûts d'établissement et de fonctionnement des entrepôts.

La génération des problèmes ainsi que les résultats des deux premières séries d'expériences sont décrits plus en détail dans [9].

3.2. Première série.

Le but de cette série d'expériences est d'étudier les distributions des erreurs par rapport à la solution optimale obtenues en utilisant les algorithmes heuristiques. A cet effet 50 problèmes de même taille sont résolus par tous les algorithmes. Chaque problème est obtenu par tirage au sort de 40 des 57 villes; ces 40 villes sont considérées comme lieux de demande et comme endroits possibles pour construire des entrepôts.

Pour chaque problème et chaque algorithme, on note le temps de résolution, le coût optimal, le nombre et les emplacements des entrepôts.

3.3. Deuxième série.

Le but de cette série d'expériences est d'étudier la variation des temps de résolution sur ordinateur en fonction de la taille des problèmes. A cet effet 50 problèmes de taille variable sont résolus par l'algorithme exact et par l'algorithme heuristique ayant donné la plus petite erreur moyenne au cours de la première série d'expériences.

Les 57 villes sont considérées comme lieux de demande. Pour chaque problème, un nombre m compris entre 5 et 50 est d'abord tiré au hasard. m villes sont ensuite tirées au hasard parmi les 57 villes, comme emplacements possibles pour établir des entrepôts.

Pour chaque problème et pour les deux algorithmes, on note le temps de résolution, le coût optimal, le nombre et les emplacements des entrepôts.

3.4. Troisième série.

Le but de cette série d'expériences est d'étudier la variation de la solution optimale lorsqu'on modifie les coûts fixes d'établissement des entrepôts; on considère des problèmes avec 57 lieux de demande et un nombre d'emplacements où l'on peut construire des entrepôts variant entre 30 et 50.

Ces problèmes sont résolus par LOCA 1 et la série est recommencée pour des coûts fixes diminuant de 10 % de leur valeur, en partant de 100 % et allant jusqu'à 60 % de celle-ci. Il y a donc pour chaque problème 5 niveaux de coûts fixes.

3.5. Conclusions.

La première série d'expériences montre que tous les algorithmes heuristiques donnent de bons résultats du point de vue des erreurs : une erreur non nulle est obtenue pour moins de 40 % des problèmes avec n'importe lequel de ces algorithmes. La moyenne des erreurs positives est toujours inférieure à 0,5 % de la valeur de la solution optimale. Les meilleurs résultats sont obtenus avec l'algorithme ATTILA 2 : une solution non optimale a été obtenue pour 4 problèmes sur 50 seulement. Les algorithmes ATTILA 1 et 2 donnent de meilleurs résultats que BABEL 1 et 2 respectivement.

Le temps de calcul par ATTILA 2 est 4 fois moindre que par LOCA 1. La dispersion des temps de calcul est faible pour tous les algorithmes heuristiques et forte pour LOCA 1.

Pour analyser les résultats de la deuxième série, on a ajusté aux résultats obtenus par les 2 algorithmes une courbe d'équation :

$$T = a\, m^{b} \tag{27}$$

où T est le nombre de secondes de temps de calcul sur IBM 7040 et m le nombre d'emplacements.

On trouve pour LOCA 1 :

$$T = 0{,}01\ m^{2,64} \tag{28}$$

et pour ATTILA 2 :

$$T = 0{,}03\ m^{1,97} \tag{29}$$

La troisième série d'expériences montre que les solutions obtenues pour ce type de problèmes sont peu sensibles aux variations des coefficients f_i. Pour 4 problèmes sur 11, la même solution est obtenue pour les 5 valeurs des coûts fixes et pour 6 autres problèmes sur 11 la même solution est obtenue pour 4 valeurs sur 5 des coûts fixes.

Pour 9 problèmes sur 11, la diminution des coûts fixes se traduit uniquement par l'ouverture de nouveaux entrepôts. Dans 2 cas seulement sur 11 il y a également une fermeture.

En conclusion, quand les problèmes sont de taille moyenne, jusqu'à 80 villes environ, l'algorithme LOCA 1 est efficace. Pour des problèmes plus larges, il faut employer une méthode heuristique. Dans ce cas l'algorithme ATTILA 2 donne de bons résultats.

4. Extensions.

4.1. Un modèle avec contraintes de capacité.

Le modèle M1 suppose que les entrepôts ont des capacités illimitées; cette hypothèse est valable au cas où un réseau entièrement nouveau doit être établi; au cas où certains entrepôts existent déjà (les variables y_i correspondantes sont alors fixées à 1), il est nécessaire de tenir compte des contraintes de capacité; de même, lorsqu'on étudie un problème de localisation d'usines, il faut tenir compte des limites des capacités de production des usines existantes. Pour ce problème, la généralisation suivante du modèle de Balinski a été proposée par G. SA [18] :

$$\min \left(\sum_{i \in M} \sum_{j} c_{ij}\, x_{ij} + \sum f_i\, y_i \right) \tag{30}$$

$$\sum_{i} x_{ij} = d_j \qquad j = 1, \dots, n \tag{31}$$

$$\sum_{j} x_{ij} \leqslant y_i\, a_i \qquad i = 1, \dots, m \tag{32}$$

$$y_i \in \{0, 1\} \qquad i = 1, \dots, m \tag{33}$$

$$0 \leqslant x_{ij} \qquad i = 1, \dots, m,\ j = 1, \dots, n \tag{34}$$

où x_{ij} est la quantité de bien allant de i à j,
c_{ij} le coût unitaire de transport de i à j,
a_i la capacité de l'entrepôt projeté en i,
d_j la demande en j; les f_i et y_i s'interprètent comme dans le modèle M1.

Sâ a proposé un algorithme semblable à l'algorithme d'Efroymson et Ray mais moins efficace.

P.S. DAVIS et T.L. RAY [5] ont proposé un autre algorithme utilisant des évaluations très précises de la valeur de la fonction économique; ces évaluations sont calculées en résolvant un problème de transport généralisé, par une méthode de décomposition utilisant récursivement l'algorithme « out of kilter » de Fulkerson. La solution optimale est souvent obtenue sans séparation pour des problèmes pratiques; le calcul de l'évaluation étant long, l'algorithme ne peut être utilisé que pour des problèmes de taille petite ou moyenne.

D.H. MARKS [13], [14] ainsi que B. ROY et R. BENAYOUN [17] ont proposé des procédures d'optimisation par séparation sur un graphe qui peuvent également être utilisées pour ce problème.

4.2. Un modèle pour la localisation de centres de service.

On cherche à maximiser une fonction de l'utilité des usagers avec un budget limité.

Le modèle suivant, dû à C. REVELLE et R.W. SWAIN [15], permet de localiser p centres de services de manière à minimiser la somme des trajets des usagers :

$$\min \sum_i \sum_j a_{ij} x_{ij} \tag{35}$$

$$\sum_i x_{ij} = 1 \qquad j = 1, \ldots, n \tag{36}$$

$$\sum_j x_{ij} \leqslant n_i y_i \qquad i = 1, \ldots, m \tag{37}$$

$$\sum_i y_i = \mathrm{P} \tag{38}$$

$$y_i \in \{0, 1\} \qquad i = 1, \ldots, m \tag{39}$$

$$0 \leqslant x_{ij} \qquad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{40}$$

où a_{ij} est la distance entre i et j, P le nombre d'entrepôts à ouvrir; les x_{ij}, les n_i et les y_i s'interprètent comme dans le modèle M1.

La contrainte (38) est due au budget limité.

La solution obtenue par programmation linéaire est souvent en variables entières; si ce n'est pas le cas, on utilise une procédure d'optimisation par séparation en choisissant une variable fractionnaire et en la fixant d'une part à 0 et d'autre part à 1.

Le nombre de variables ($n \times m$) limite en pratique la taille des problèmes qu'on peut résoudre par cette méthode.

REFERENCES

[1] ATKINS, R.J. and SHRIVER, R.H. (1968) : « New Approach to Facilities Location », *Harvard Business Review* 46, 70-79.

[2] BALINSKI, M.L. (1961) : « Fixed-cost Transportation Problems », *Naval Research Logistics Quarterly,* 8, 41-54.

[3] BALINSKI, M.L. (1964) : « On Finding Integer Solutions to Linear Programs », *Mathematica-Princeton,* N.J.

[4] BAUMOL, W.J. and KUHN, M.W. (1962) : « An approximate Algorithm for the Fixed-Charges Transportation Problem », *Naval Research Logistics Quarterly,* 9, 1-15.

[5] DAVIS, P.S. and RAY, T.I. (1969) : « A Brand-Bound Algorithm for the capacitated Facilities Location Problem ». *Naval Research Logistics Quarterly* 16, 331-344.

[6] DESCAMPS, R. (1966) : « Un algorithme d'optimisation pour une classe de problèmes d'implantation » dans HERTZ, D.B. et MELESE, J., Editeurs, *Actes de la Quatrième Conférence internationale de Recherche Opérationnelle,* Wiley-Interscience, New York, 834-842.

[7] EFROYMSON, M.A. and RAY, T.L. (1966) : « A Branch and Bound Algorithm for Plant Location », *Operations Research,* 14, 361-368.

[8] FELDMAN, E., LEHRER, F.A. and RAY, T.L. (1966) : « Warehouse Location under continuous Economies of Scale ». *Management Science,* 12, 670-684.

[9] HANSEN, P. et KAUFMAN, L. (1970) : « Comparaison d'algorithmes pour le problème de la localisation des entrepôts », à paraître dans J. BRENNAN, Editor, « *Operations Research in Industrial Systems* », English University Press (1971).

[10] HENRY, G. (1968) : « Recherche d'un réseau de dépôts optimum », *Revue Française d'Informatique et de Recherche Opérationnelle,* 2, 61-70.

[11] KUEHN, A.A. and HAMBURGER, M.J. (1963) : « A Heuristic Program for Locating Warehouses ». *Management Science,* 10, 643-666.

[12] MANNE, A.S. (1964) : « Plant Location under Economies of Scale Decentralisation and Computation », *Management Science* 11, 213-235.

[13] MARKS, D.H. (1969) : « Facility Location and Routing models in Solid Waste Collection Systems », Ph. D. Thesis, The Johns Hopkins University.

[14] REVELLE, C., MARKS, D. and LIEBMAN, J.C. (1970) : « An Analysis of Private and Public Sector Location Models », *Management Science* 16, 692-707.

[15] REVELLE, C. and SWAIN, R.W. (1970) : « Central Facilities Location », *Geographical Analysis,* 2, 30-42.

[16] ROY, B. (1969) : « Procédures d'exploration par séparation et évaluation (P.S.E.P. et P.S.E.S.), *Revue Française d'Informatique et de Recherche Opérationnelle,* 3, 61-90.

[17] ROY, B. et BENAYOUN, R. (1967) : « Programmes linéaires en variables bivalentes et continues sur un graphe (le programme POLIGAMI) », *METRA,* 6, 675-710.

[18] SA, G. (1969) : « Branch and Bound and Approximate Solutions to the Capacitated Plant Location Problem », *Operations Research,* 17, 1005-1016.

[19] SPIELBERG, K. (1969) : « Algorithms for the Simple Plant-Location Problem with some Side Conditions », *Operations Research,* 17, 85-111.

[20] SPIELBERG, K. (1969) : « Plant Location with Generalized Search Origin », *Management Science,* 16, 165-178.

[21] STURM, L.B.J.M. (1969) : « On the Warehouse Location Problem », Communication at the « *1969 European Econometric Society Meeting* », Brussels.

[22] ZIMMERMAN, R. (1967) : « A Branch and Bound Algorithm for Depot Location », *METRA* 4, 661-674.