ENSEIRB-MATMECA

Filière informatique, 2ème année

Promotion 2017

Projet S7: Rapport de projet : Recherche Opérationnelle

Auteur :
TRECU Damien

Encadrant :
DETIENNE Boris

16 décembre 2015

1 Introduction

Dans le cadre de notre formation au département informatique de l'ENSEIRB-MATMECA, au sein du module IF227 de recherche opérationnelle, il nous a été demander de réaliser un projet mettant en application les connaissances théoriques acquises durant les cours et les TD.

Le projet consiste en la modélisation d'un problème concernant la collecte du café en Colombie auprès des producteurs locaux. La filière est organisée de la manière suivante : les agriculteurs vendent leur production à des centres d'achat dans des lieux proche de leur exploitation. Une fois le café collecté, les coopératives doivent transférer la production vers des dépôts appartenant à la fédération nationale des producteurs de café.

Néanmoins, avec la baisse du cours du café, le réseau de collecte du café doit être remanié afin de réduire les coûts liés à l'infrastructure et son maintien. L'objectif principal du projet est donc de réaliser un modèle permettant de minimiser ces coûts, afin de pouvoir déterminer quels dépôts doivent être maintenus ou non, et quels centres d'achat seront gérés par tel ou tel dépôt. D'autant que les producteurs veulent que leur production soit vendue à un dépôt proche de leur exploitation.

2 Modélisation du problème (Question 1)

2.1 Données

Pour la réalisation de notre modèle, plusieurs informations vont être nécessaires pour mettre en place les différents contraintes qui vont régir notre objectif. Ces données sont les suivantes :

— I : l'ensemble des dépôts actuellement en fonctionnement dans le réseau
— J : l'ensemble des centres d'achat coopératifs présents
— $\forall i \in I$, f_i représente le coût fixe d'exploitation du dépôt i
— $\forall j \in J$, d_j représente le volume de production du centre d'achat j
— $\forall i \in I$, $\forall j \in J$, c_{ij} représente le coût de collecte de la production du centre j par le dépôt i
— $\forall i \in I$, $\forall j \in J$, h_{ij} représente la distance entre le dépôt i et le centre d'achat j
— D_{max} : une constante entière représentant la distance maximale pour laquelle on considère qu'un dépôt et un centre d'achat sont géographiquement proches
— v : une constante flottante représentant la proportion minimale de la production totale de café qui doit être récupérée par des dépôts proches de leurs centres d'achat

2.2 Variables de décision

Pour compléter les données fournis et résoudre le problème posé, nous allons ajouter les variables de décision suivantes :

— $\forall i \in I$, m_i : variable booléenne valant 1 si le dépôt i est $m$aintenu, 0 sinon
— $\forall i \in I$, $\forall j \in J$, r_{ij} : variable booléenne valant 1 si le dépôt i est $r$elié au centre d'achat j, 0 sinon

— $\forall i \in I$, $\forall j \in J$, p_{ij} : variable booléenne valant 1 si le dépôt i est *p*roche du centre d'achat j, 0 sinon. Bien que déconseillée par l'encadrant, pour pouvoir simplifier le modèle final, il nous sera utile de l'avoir, notamment pour une contrainte particulière du modèle.

2.3 Modélisation finale

2.3.1 Objectif

Maintenant que nous avons tout ce qui est nécessaire pour réaliser notre modèle, nous allons pouvoir passer à sa réalisation. Tout d'abord, il nous faut commencer par l'objectif à atteindre, qui est ici une minimisation des coûts du réseau de collecte. Ces derniers sont en fait la somme des coûts liés au maintien des dépôts en service et des coûts de collecte entre les dépôts et leurs centres d'achats. On obtient donc aisément l'objectif suivant :

$$\text{Minimiser} \quad (\sum_{i \in I} f_i \times m_i) + (\sum_{i \in I} \sum_{j \in J} r_{ij} \times c_{ij})$$

2.3.2 Contraintes

Grâce à l'analyse de l'énoncé du problème, et à une analogie avec les problèmes classiques de localisation *(facility location problems)*, nous obtenons les contraintes suivantes :

$$\begin{gathered}
\forall j \in J, \ \sum_{i \in I} r_{ij} = 1 \\
\forall i \in I, \forall j \in J, \ \ r_{ij} \leq m_i \\
\forall i \in I, \forall j \in J, \ \ r_{ij} \geq 0 \\
\forall i \in I, \forall j \in J, \ \ p_{ij} \times h_{ij} \leq D_{max} \\
\sum_{i \in I} \sum_{j \in J} p_{ij} \times r_{ij} \times d_j \geq v \times \sum_{j \in J} d_j
\end{gathered}$$

Explication des contraintes ci-dessus :

— La première contrainte stipule qu'il ne doit y avoir qu'un seul et unique dépôt relié à un centre d'achat donné

— La deuxième contrainte stipule qu'on ne peut pas relier un centre d'achat à un dépôt si ce dernier n'est pas maintenu

— La troisième contrainte stipule qu'un centre d'achat peut être relié ou non à un dépôt (ceci vient en complément de la deuxième)

— La quatrième contrainte stipule qu'un dépôt est considéré proche d'un centre d'achat si la distance les séparant est inférieure à la borne limite D_{max}

— La dernière contrainte stipule qu'une partie de la production totale de tous les centres d'achat doit être collectée par des dépôts qui sont proches de leurs centres d'achat

Pour définir cette dernière contrainte difficile, il faut calculer en fait la somme de la production des centres d'achat reliés à des dépôts qui sont proches de ceux-ci. Puis on calcule cette somme pour tous les dépôts présents dans le réseau, et on vérifie qu'elle est supérieure à la proportion de la production totale de tous les centres d'achat.

3 Heuristiques de résolution (Question 3)

Maintenant que nous avons réussi à modéliser le problème, il nous a été demandé de réfléchir à deux algorithmes calculant une solution approchée de la solution finale du problème énoncé. Ceci fait, nous pourrons comparer les résultats obtenus avec celui d'IBM ILOG OPL.

Néanmoins, de par les problèmes d'organisation rencontrés pour le rendu de ce rapport, nous n'en présenterons qu'un seul ici, et le deuxième sera recherché et développé d'ici le rendu final du projet avec l'implémentation des deux propositions.

3.1 Algorithme d'approximation

Les parties les plus simples de l'algorithme imaginé seront détaillé de manière correcte, tandis que les parties qui seront plus spécifiques à l'implémentation finale en machine seront écrits de manière textuelle et générale.

```
Objectif
    Minimiser (∑_i ∈ I f_i × m_i) + (∑_i ∈ I ∑_j ∈ J r_ij × c_ij)
Initialisation
    m ← créerTableau(I, FAUX)
    r ← créerMatrice(I, J, FAUX)
    p ← créerMatrice(I, J, FAUX)
    tout_relié ← créerTableau(J, FAUX)
    total_production ← 0

Procédure
    Pour Tout i de 1 à I Faire
        Pour Tout j de 1 à J Faire
            Si h[i][j] ≤ D_max Alors
                p[i][j] ← VRAI
    Pour Tout j de 1 à J Faire
        total_production ← total_production + d[j]

    Tant que (NON toutAVrai(tout_relié)) Faire
        Pour Chaque centre d'achat j de J non relié à un dépôt Faire
            Si le coût pour rattacher j à un dépôt ouvert
                est inférieur au coût pour ouvrir un dépôt
                fermé et y rattacher j
                Alors
```

```
                i ← le dépôt optimal
                r[i][j] ← VRAI
        Sinon
            i ← le dépôt dont la somme du coût
                d'ouverture et du rattachement de j à
                ce dépôt est minimal
            m[i] ← VRAI
            r[i][j] ← VRAI

Vérifier que la contrainte concernant la proportion v est respectée
Si ce n'est pas le cas Alors
    Effectuer les changements nécessaires dans m et r pour y parvenir
    → changement de rattachements entre les coopératives et
      les dépôts déjà ouverts dans m
    → ouverture de dépôts si réellement nécessaire

Retourner m,r
```