Rapport intermédiaire du projet sur la résolution exacte du SSCFLP et ses simplifications

Jocelin Cailloux

November 27, 2016

1 Les problèmes

Le Facility Location Problem est un problème qui consiste à chercher où installer un certain type de facilités afin d'en minimiser le coût d'installation et de maintenance. Il existe une variante du problème pour laquelle on cherche à couvrir un maximum de personnes à moindre coût. Ici, le problème est simplifié, on doit couvrir tout le monde et on cherche le coût minimum. De plus, les facilités ne peuvent pas être installées à n'impote quel emplacement mais sur une liste d'emplacements définie, ce qui rend le problème discret.

Pour les modélisations, nous utilisons les variables suivantes :

- W_j la quantité nécessitée par la demande j
- S_i la quantité maximale que peut fournir la facilité i
- F_j le coût de construction d'une facilité à l'emplacement j
- C_{ij} le coût de liaison entre la demande i et la facilité j
- $X_i = \begin{cases} 1, & \text{si une facilité est installée à l'emplacement i} \\ 0, & \text{sinon} \end{cases}$
- $Y_{ij} = \begin{cases} 1, & \text{si la demande j est assignée à la facilité i} \\ 0, & \text{sinon} \end{cases}$

1.1 UFLP

La variante UFLP du problème est la plus simple traitée ici. Dans cette variante, les facilités peuvent satisfaire tous les besoins d'autant de demandes que nécessaire.

Nous obtenons donc la modélisation suivante :

$$min(z) = \sum_{i \in I}(X_i * F_i + \sum_{j \in J}(Y_{ij} * C_{ij})) \tag{1}$$

Sous la contrainte suivante :

- Une demande doit être liée à une et une seule facilité

$$\sum_{i \in I} X_i * Y_{ij} = 1, \quad j \in J \tag{2}$$

1.2 CFLP

La vriante CFLP est la plus permissive. Dans cette variante, une demande peut-être complétée par plusieurs facilitées.

Pour ce faire, la variable Y_{ij} ne stocke plus 0 ou 1 mais une proportion (entre 0 et 1) de la demande W_j. De plus, nous n'avons plus la contrainte limitant à 1 le nombre de facilités reliées. Cette contrainte est remplacée par la contrainte vérifiant que la demande est entièrement complétée. La fonction objectif est la même.

Nous optenons donc la modélisation suivante :

$$min(z) = \sum_{i \in I}(X_i * F_i + \sum_{j \in J}(Y_{ij} * C_{ij})) \tag{3}$$

- Une demande doit être entièrement complétée par les facilités auxquelles elle est reliée

$$\sum_{i \in I} X_i * Y_{ij} = 1, \quad j \in J \tag{4}$$

- Une facilité ne peut pas fournir plus que sa capacité maximale

$$\sum_{j \in J} X_i * Y_{ij} * W_j \leq S_i, \quad i \in I \tag{5}$$

1.3 SSCFLP

Cette variante est la plus compliquée des variantes traitées ici. Dans cette variante, une demande doit être liée à une et une seule facilité. Cependant, contrairement à la variante UFLP, chaque facilité à une quantité maximale qu'elle peut fournir.

Pour ce faire, nous reprenons les variables telles qu'elles sont décrites au début du chapitre. Nous n'avons plus besoin de vérifier qu'une demande est entièrement complétée par les facilités auxquelles elle est reliée. En effet, il n'y a qu'une seule facilité liée à une demande, nous savons donc que la facilité complète toute la demande. Il faut Toutefois vérifier que la facilité a la capacité de fournir toutes les demandes auxquelles elle est liée. La fonction objectif reste la même.

Nous optenons donc la modélisation suivante :

$$min(z) = \sum_{i \in I} (X_i * F_i + \sum_{j \in J} (Y_{ij} * C_{ij})) \tag{6}$$

- Une demande doit être liée à une et une seule facilité

$$\sum_{i \in I} X_i * Y_{ij} = 1, \quad j \in J \tag{7}$$

- Une facilité ne peut pas fournir plus que sa capacité maximale

$$\sum_{j \in J} X_i * Y_{ij} * W_j \leq S_i, \quad i \in I \tag{8}$$

2 Les instances

Les instances que nous utiliserons au cours de ce projet ont été trouvé à l'adresse suivante : `http://www-eio.upc.es/~elena/sscplp/index.html`

Nous utiliserons 4 petites instances de taille 20X10, les instances p1, p3, p4 et p5. Ces petites instances permettront de tester les algorithmes. Ils permettront de dérouler les algorithmes de manière relativement simple et ainsi aideront au débogage. Ils permettront ainsi aussi de tester et comparer certaines astuces qui pourront être tentées pour améliorer l'efficacité de la résolution. Les instances p1, p3, p4 et p5 semblent pour cela correspondre à une bonne combinaison d'instances. En effet, elles ont toutes des caractéristiques particulières. p1 possède des coûts de fabrication des facilités relativement faibles, p3 possède des capacités de facilités plutôt faibles, p4 possède des coûts de fabrication élevés et p5 a tendance à avoir des facilités dont les capacités sont elevées.

Ensuite, nous utiliserons des instances un peu plus grande, 30X15, ces instances permettront d'observer comment ralentit l'algorithme en fonction de la taille des données et d'améliorer ce qui prend du temps. Afin d'avoir des comparaisons plus pertinentes, nous devrions peut-être utiliser des instances dont les capacités des facilités sont plus ou moins grands. Je pense en effet que cette caractéristique d'une instance est intéressante à étudier. C'est pourquoi nous utiliserons les instances p11 et p14 qui ont respectivement des capacités faibles et élevées.

Nous utiliserons seulement une instance de chaque taille pour les plus grandes. Il s'agît surtout de tester le comportement de l'algorithme en augmentant de plus en plus la taille des données. Les instances p25, p32, p35 et p45 sont choisies de manière arbitraire. Elles sont respectivement de tailles 40X20, 50X20, 60X30 et 75X30.

Une fois l'algorithme stabilisé, ou pour tester le comportement dans un contexte spécifique, d'autres instances pourront être utilisées. Il est possible que ces instances choisies soient amenées à changer si les tests des premières instances indiquent que la variation de certaines caractéristiques sont plus pertinentes à tester que prévu.

3 Les temps CPU et les meilleures solutions obtenues à l'aide de GLPK

- Solution de p1 :
  - Temps de résolution : 8 minutes et 55 secondes
  - Valeur de la fonction objectif : 2014
  - Les facilités à construire sont : {1,2,3,4,6,7,8}
  - Les liaisons à faire sont :
    - (1,{1,12,14})
    - (2,{7,8,18})
    - (3,{10,11})
    - (4,{0,3,13,15})
    - (6,{6,16,17})
    - (7,{4,5})
    - (8,{2,9,19})

- Solution de p3 :
  - Temps de résolution : 15 minutes et 16 secondes
  - Valeur de la fonction objectif : 6051
  - Les facilités à construire sont : {0,2,3,4,6,8,9}
  - Les liaisons à faire sont :
    * (0,{1})
    * (2,{2,9,18})
    * (3,{0,8,13,14})
    * (4,{7,16,17})
    * (6,{6,15})
    * (8,{4,5,10,19})
    * (9,{3,11,12})

- Solution de p4 :
  - Temps de résolution : 19 secondes
  - Valeur de la fonction objectif : 7168
  - Les facilités à construire sont : {0,3,4,5,6,7,9}
  - Les liaisons à faire sont :
    * (0,{5,6,7,15,18})
    * (3,{17,19})
    * (4,{10,11,12})
    * (5,{16})
    * (6,{1,2,8})
    * (7,{0,14})
    * (9,{3,4,9,13})

- Solution de p5 :
  - Temps de résolution : 22 secondes
  - Valeur de la fonction objectif : 4551
  - Les facilités à construire sont : {0,1,3,4,5,6,8}
  - Les liaisons à faire sont :
    * (0,{0,7,16,19})
    * (1,{5,13,15})

    * (3,{10,14,17})
    * (4,{1,2,8})
    * (5,{6})
    * (6,{3,9,12,18})
    * (8,{4,11})

- Solution de p11 : Non résolu

- Solution de p14 :
  - Temps de résolution : 3 minutes et 39 secondes
  - Valeur de la fonction objectif : 5965
  - Les facilités à construire sont : {1,3,4,6,7,13,14}
  - Les liaisons à faire sont :
    * (1,{4,10,20})
    * (3,{22,23})
    * (4,{14})
    * (6,{0,26,29})
    * (7,{5,9,12,21,24,25})
    * (13,{3,13,15,16,27})
    * (14,{1,2,6,7,8,11,17,18,19,28})

- Solution de p25 :
  - Temps de résolution : 73 millisecondes
  - Valeur de la fonction objectif : 8947
  - Les facilités à construire sont : {1,3,4,5,18}
  - Les liaisons à faire sont :
    * (1,{11,15,16,20,22,26,27,29,33,34})
    * (3,{0,4,14,19,21,24,31,32,38})
    * (4,{1,8,9,13,17,37,39})
    * (5,{3,6,7,12,25,28})
    * (18,{2,5,10,18,23,30,35,36})

- Solution de p32 :
  - Temps de résolution : 9 secondes
  - Valeur de la fonction objectif : 9881
  - Les facilités à construire sont : {2,4,6,7,8}
  - Les liaisons à faire sont :
    * (2,{6,8,9,15,16,17,20,23,24,33})
    * (4,{0,3,18,19,26,39,44,45,47})
    * (6,{10,11,13,14,28,30,31,32,40,48})
    * (7,{1,22,27,29,37,38,43,46})
    * (8,{2,4,5,7,12,21,25,34,35,36,41,42,49})

- Solution de p35 :
  - Temps de résolution : 12 minutes et 38 secondes
  - Valeur de la fonction objectif : 5456
  - Les facilités à construire sont : {1,4,16,18,21,22,27,28}
  - Les liaisons à faire sont :
    * (1,{15,37,45,47,50,52})
    * (4,{1,2,4,6,13,17,34})
    * (16,{0,7,9,12,14,41,42,55,59})
    * (18,{16,19,21,25,26,32,54})
    * (21,{8,18,20,22,38,58})
    * (22,{3,23,24,27,40,43,44,48,49})
    * (27,{30,31,35,36,46,51,56})
    * (28,{5,10,11,28,29,33,39,53,57})

- Solution de p45 :
  - Temps de résolution : 3 secondes
  - Valeur de la fonction objectif : 17676
  - Les facilités à construire sont : {5,7,9,13,14,15,18,20,28}
  - Les liaisons à faire sont :
    * (5,{3,9,14,18,20,21,24,43,61,71})
    * (7,{12,52,53,55,63})
    * (9,{0,2,17,26,44,49})

- (13,{13,15,30,33,39,46,58,66,67,68})
- (14,{1,7,10,27,31,64})
- (15,{5,25,36,38,45,50,57,60,62,72})
- (18,{4,6,16,29,40,41,51,65,69,74})
- (20,{8,22,23,32,47,48,54,70})
- (28,{11,19,28,34,35,37,42,56,59,73})