

Projet de Recherche Opérationnelle

Localisation d'entrepôts

Le projet porte sur deux problèmes de localisation d'entrepôts, qui diffèrent en fait par la présence (ou l'absence) de limites sur la capacité des entrepôts. Reprenons les données de l'exercice 5 du TD :

- La capitale, Lordaeron (qui a besoin de 500 unités de nourriture)
- La citadelle de Dalaran (besoin : 350 unités)
- La cité de Stratholme (besoin : 300 unités)
- La forteresse de Mardenholde (besoin : 400 unités)
- La forteresse de Durnholde (besoin : 350 unités)
- La cité de Caer Darrow (besoin : 250 unités)

Les entrepôts peuvent être placés à Brill, Hearthglen, Andorhal et Strahnbrad. Les coûts de constructions associés sont respectivement de 200, 250, 300 et 350 pièces d'or. Les coûts de transports entre les lieux possibles pour les entrepôts et les cités à desservir sont les suivants (en pièces d'or). Dans un cas, la capacité des entrepôts est infinie, dans le second cas, elle est de 800.

	Brill	Hearthglen	Andorhal	Strahnbrad
Lordaeron	100	200	250	350
Dalaran	500	450	350	250
Stratholme	125	75	125	300
Mardenholde	175	50	175	300
Durnholde	300	300	200	100
Caer Darrow	350	350	250	150

Typiquement, le problème est de placer les entrepôts, de manière à minimiser les coûts, tout en satisfaisant les différentes contraintes (demande, capacité).

Documents à réaliser

La bonne réalisation d'un logiciel ne peut se faire en se jetant directement dans le code sans comprendre ce que l'on fait. Comme le projet est assez "léger", vous n'aurez pas à fournir autant de volume de documentation qu'un projet à plein temps, mais cela reste impératif.

Document technique : ce document décrit plus en détail le sujet traité. Il doit présenter un modèle mathématique commenté. Pour les métaheuristiques, il doit présenter les voisinages choisis, le(s) type(s) de mouvement(s) ainsi que les réglages nécessaires.

Document organique : ce document présente l'ossature du logiciel de résolution du problème. Il décrit de manière détaillée les différentes classes (C++, Java, autre) du programme (variables, méthodes, données,...) et leurs rôles dans le logiciel.

Document utilisateur : ce document décrit le mode d'emploi du logiciel.

Document de synthèse des résultats : il constitue la deuxième partie du document technique. Il doit présenter les résultats détaillés et commentés des différentes approches et les comparaisons entre les différentes approches.

Ces documents feront 4 ou 5 pages¹, et seront à rendre en temps et en heure (voir planning plus bas).

¹Vous avez le droit de faire plus si cela reste raisonnable.

Modèle mathématique

Il s'agit du modèle vu en TD pour l'exercice 5. Une écriture générale de ce modèle est :

$$\begin{aligned} \min z &= \sum_{i=1}^m f_i y_i + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.c.} \quad & \sum_{i=1}^m x_{ij} \geq d_j \\ & \sum_{j=1}^n x_{ij} \leq s_i y_i \\ & 0 \leq x_{ij} \leq s_i \\ & 0 \leq x_{ij} \leq d_j \\ & x_{ij} \in \mathbb{R}^+, y_i \in \{0, 1\} \end{aligned}$$

Avec :

x_{ij} le nombre d'unités transportées de l'entrepôt i vers le point de demande j

$$y_i = \begin{cases} 1 & \text{si l'entrepôt } i \text{ est utilisé} \\ 0 & \text{sinon} \end{cases}$$

d_j la demande au point j

s_i la capacité de l'entrepôt au point i

f_i le coût d'installation d'un entrepôt au point i

c_{ij} le coût d'acheminement de l'entrepôt i vers le point de demande j

Dans le cas où les entrepôts ont une capacité "infinie", le problème s'écrit :

$$\begin{aligned} \min z &= \sum_{i=1}^m f_i y_i + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.c.} \quad & \sum_{i=1}^m x_{ij} \geq d_j \\ & \sum_{j=1}^n x_{ij} \leq S y_i \\ & 0 \leq x_{ij} \leq d_j \\ & x_{ij} \in \mathbb{R}^+, y_i \in \{0, 1\} \end{aligned}$$

Avec $S = \sum_{j=1}^n d_j$.

Approches

Le but de ce projet est de résoudre ces deux problèmes à l'aide de différentes méthodes (exactes, heuristiques, métaheuristiques), et de les comparer. De manière non exhaustive :

- Énumération (1)
- Branch & Bound, Branch & Cut (2)
- Algorithme glouton (3)
- Algorithme d'échange (4)
- Recherche tabou (5)
- Recuit simulé (6)
- Recherche à voisinage variable (7)
- Algorithme génétique (8)
- etc...

Vous aurez à tester les approches 1 à 7. Les autres méthodes (8, par exemple) sont optionnelles et viendront bonifier votre note.

Outils

Vous avez ici une assez grande autonomie concernant les outils que vous pouvez utiliser, sachant toutefois que vous devez utiliser un langage orienté objet. Pour les différentes méthodes que vous choisirez de tester, vous pourrez faire appel à des programmes déjà existants pour faire une partie du travail. Notamment, pour du Branch & Bound, à Cplex, ou pour du Branch & Cut, à Coin-Cbc. Idem pour l'interface et l'affichage, il y a des bibliothèques faites pour afficher des graphes, servez-vous en.

Votre programme devra être capable de lire un fichier fourni dans le format décrit en annexe. L'utilisateur aura ensuite le choix d'utiliser telle ou telle méthode de résolution, avec la possibilité de modifier les paramètres pertinents, puis une fois l'algorithme lancé et terminé, aura à sa disposition le résultat obtenu (le coût), ainsi qu'un "joli graphique" montrant les emplacements, les capacités des entrepôts le cas échéant, et les flux retenus. Enfin, la possibilité d'exporter les résultats est attendue.

Données

A petite échelle, vous vous servirez des données de l'exercice 5 du TD, ou d'exemples que vous inventerez. Une fois les documents techniques et organiques rendus, des jeux de données supplémentaires seront mis à votre disposition.

Planning

Vous travaillerez en binômes pour ce projet.

- 26 Novembre : La liste des binômes est finalisée.
- 10 Décembre : Les documents techniques et organiques sont rendus. Distribution des jeux de données.
- 3 Février : Le manuel utilisateur est rendu.
- 24 Février : Tous les documents sont rendus, ainsi que la version finale des logiciels.
- 11 Mars : Soutenances

Annexe : format des jeux de données

Les instances auront le format suivant :

Nombre d'emplacements potentiels pour les entrepôts (m), nombre de clients (n)

Pour chaque emplacement potentiel d'entrepôt i ($i=1,\dots,m$) :

capacité (s_i), coût fixe (f_i)

Pour chaque client j ($j=1,\dots,n$) :

demande (d_j), coût pour allouer toute la demande de j à l'entrepôt i ($i=1,\dots,m$)