

INFO-F310 - ALGORITHMIQUE ET RECHERCHE OPÉRATIONNELLE

Renaud Chicoisne
rchicois@ulb.ac.be

Jérôme De Boeck
Jerome.De.Boeck@ulb.be

1. Problème

Le problème de bin packing est un problème combinatoire intéressant avec des applications industrielles, dans lequel un ensemble de produits doit être emballé dans un ensemble de boîtes. Les produits sont classés par type T , chaque type de produit $t \in T$ a une taille $s_t > 0$. Pour chaque type d'objet $t \in T$ il y a n_t objets à emballer. Il y a donc un total de $n = \sum_{t \in T} n_t$ objets à emballer. Les boîtes ont toutes une capacité identique $c > 0$, nous considérons que $c \geq s_t$ pour tout $t \in T$. Le but est de trouver un rangement qui utilise le moins de boîtes possibles.

Nous considérons certaines contraintes additionnelles dues aux types de produits à emballer :

- *Nbr max d'objets* : Certains type d'objets $t \in T$ peuvent apparaitre dans au plus b_t boîtes. Par exemple si $b_t = 4$, le produit de type t ne pourra pas être placé dans plus de 4 boîtes.
- *Incompatibilités* : Certains objets ne peuvent être placés dans des boîtes identiques. Soit $\mathcal{I} = \{I_1, I_2, \dots, I_{|\mathcal{I}|}\}$ l'ensemble des incompatibilités. Un élément I_n de \mathcal{I} est un ensemble d'objets incompatibles, il y a donc $|\mathcal{I}|$ ensembles incompatibles. Par exemple si $(2, 4, 5) \in \mathcal{I}$, les objets de type 2, 4 et 5 doivent être dans des boîtes différentes.

2. Instance

Un répertoire contenant des instances `instance_n_t_i_id.txt` du problème vous est fourni, `n`, `t`, `i` et `id` représentent le nombre total d'objets, le nombre type d'objet, le nombre d'incompatibilité et le numéro de l'instance de cette taille. Pour chaque taille, trois instances sont fournies. Chaque fichier est au format suivant :

```
nb_objet_total |T| |I|
c
s_1 n_1 b_1
...
s_|T| n_|T| b_|T|
I_1
...
I_|I|
```

3. Documents à remettre

1. Génération de modèle :

Un script python3 nommé `generate_model.py` prenant en paramètre en ligne de commande le nom d'une instance dans le même dossier et un paramètre p égal à 0, 1 ou 2 qui génère un programme linéaire en nombre entiers de cette instance `instance_n_t_i_id.txt` au format CPLEX LP vu en TP. La valeur de p indique quelles contraintes supplémentaires considérer :

- si $p = 0$, aucune des deux contraintes additionnelles ne doivent être incluses dans le modèle,
- si $p = 1$, la contrainte *Nbr max d'objets* doit être incluses dans le modèle,

- si $p = 2$, les deux contraintes supplémentaires *Nbr max d'objets* et *Incompatibilités* sont à inclure dans le modèle.

Ce programme doit être sauvé dans un fichier `model_n_t_i_id_p.lp` où p le paramètre passé en ligne de commande. Le script appelé sur l'instance `instance_10_4_2_1.txt` via la commande

```
python3 generate_model instance_10_4_2_1.txt 2
```

doit générer un fichier `model_10_4_2_1_2.lp`. Comme utilisé en TP, le fichier doit pouvoir être résolu et sauver les résultats avec la commande

```
glpsol --lp model_10_4_2_1_2.lp -o model_10_4_2_1_2.sol
```

2. Programme linéaire :

Un fichier `model_10_4_2_1_2.sol` retourné par la fonction du point précédent sur l'instance `instance_10_4_2_1.txt` via la commande

```
python3 generate_model instance_10_4_2_1.txt 2
```

3. Solution du programme linéaire :

Un fichier `model_10_4_2_1_2.sol` qui contient la solution optimale obtenue en considérant l'instance `instance_10_4_2_1.txt` ainsi que les contraintes d'incompatibilité et de nombre maximum de boîte par type d'objet.

4. Rapport

Un rapport \LaTeX compilé au format pdf qui détaille votre formulation et ses résultats. Le rapport doit contenir au minimum :

- vos noms, prénoms et matricules,
- le système d'exploitation de la machine sur laquelle vous avez fait vos tests,
- la formulation utilisée en décrivant les différentes notations comme en TP,
- expliquer en détail pourquoi les variables, contraintes et fonction objectif du programme modélisent entièrement le problème, tâchez d'avoir une formulation de qualité. Conseils :
 - si vous utilisez des contraintes de type big-M, utilisez la plus petite constante possible en justifiant celle-ci,
 - vous pouvez estimer le nombre maximum de boîte nécessaires via une heuristique pour limiter le nombre de variables binaires, justifier cette heuristique si vous l'utilisez,
- une analyse des résultats de la résolution des différentes instances en considérant différentes valeurs pour p . Cette analyse peut comparer les objectifs et temps de résolution (attention à bien noter ceux-ci à la fin d'une résolution) pour des instances de taille croissante ainsi que l'impact des contraintes additionnelles. Vous pouvez analyser les temps de résolution moyens ainsi que l'écart-type pour les différentes tailles d'instances. Pour les instances non résolues, vous pouvez analyser si une solution faisable a été trouvée. Tâchez d'interpréter le pourquoi des résultats.

Un nombre exhaustif d'instances est fournies. Le rapport ne doit pas contenir les résultats de toutes les instances pour chaque valeur de p . Sélectionnez les instances afin d'être le plus complet dans vos résultats.

Le temps de résolution des modèles peut être limité à 10 minutes. Ajoutez l'option `--tmlim t` lorsque `glpsol` est utilisé où t est le temps limite en secondes. Vérifiez bien dans les fichiers de solutions si une solution a été trouvée et s'il est optimale ou non lorsque le temps limite est atteint.

4. Consignes de remise

Ce travail est à faire par groupe de 2 personnes. Vous êtes invité à communiquer vos groupes pour le 14 mars (un email par groupe reprenant les noms et matricules suffit). Si vous ne trouvez personne, envoyez également un email avant cette date. Si vous ne vous manifestez pas avant le 14 mars, je ne pourrai vous garantir de trouver un partenaire pour le projet.

Veillez à scrupuleusement respecter le nom des fichiers remis et la syntaxe des appels en ligne de commande. Le rapport doit être remis au format pdf.

Pour remettre votre projet sur l'UV, nous vous demandons de

- créer localement sur votre machine un répertoire de la forme `NOM1_NOM2` (exemple : `CHICOISNE_DEBOECK`, sans espace) dans lequel vous mettez les fichiers demandés (sans y inclure de fichier de données)

- compresser ce répertoire via un utilitaire d'archivage produisant un **.zip** (aucun autre format de compression n'est accepté)
- de soumettre le fichier archive **.zip**, et uniquement ce fichier, sur l'UV (une seule remise par binôme)

Le projet est à remettre pour le 18 avril 2021 à 14h sur l'UV. Tout manquement aux consignes ou retard sera sanctionné directement d'un **0/10**.