

Bin packing avec recours

1. Bin packing avec plusieurs types

Les problèmes de bin-packing consistent à calculer le nombre de conteneurs nécessaires au rangement d'un certain nombre d'objets donnés. Si les poids des objets sont arbitraires, le problème est difficile à résoudre. Cependant, certaines variantes du problème sont faciles. Par exemple, supposons que les objets ont tous la même taille, et qu'il existe deux types de conteneurs et d'objets:

- le premier type de conteneur peut contenir uniquement le premier type d'objet,
- le second type de conteneur peut contenir uniquement le second type d'objet,
- le troisième type de conteneur peut contenir les deux types d'objets.

Ce problème peut représenter le transport de deux types d'aliments, certains nécessitant de conteneurs réfrigérés, plus chers que les conteneurs classiques. La version simplifiée ci-dessous peut être résolue en temps polynomial.

2. Demande incertaine

Supposons que le nombre d'objets de chaque type ne soit pas connu au moment où l'on doit commander les conteneurs. Le problème devient alors un problème à plusieurs niveaux de décision:

1. on doit choisir le nombre de conteneurs de chaque type,
2. le nombre d'objets de chaque type est connu,
3. il faut affecter les objets aux conteneurs, en espérant qu'ils tiennent dans les conteneurs choisis au premier niveau.

Ce problème est un exemple très simple de problème d'optimisation robuste ajustable. Si le nombre de scénarios est constant, on peut résoudre le problème en temps polynomial grâce à une formulation PLNE avec un nombre constant de variables et de contraintes, en utilisant un résultat bien connu.

3. Objectifs du stage

Le premier objectif est de trouver un algorithme polynomial qui décrit comment calculer les nombres de conteneurs de chaque type nécessaires. Plutôt que d'utiliser le résultat abstrait mentionné précédemment, nous aimerions un algorithme intelligible et combinatoire. Les objectifs suivants pourront emprunter deux directions complémentaires, suivant la volonté de l'étudiant :

- étudier des généralisations simples du problème précédent (plus grand nombre de scénarios, poids différents pour les objets, ...).
- coder un algorithme pour résoudre le problème dans le cas général

4. Compétences requises

De bonnes compétences en algorithmique, éventuellement en algorithmes d'approximation. Selon la piste choisie par l'étudiant, de bonnes connaissances en programmation mathématique en nombres entiers et en optimisation robuste pourraient être intéressantes également.

5. Détails pratique

Le stage durera 6 mois, aura lieu au LIRMM (Montpellier), et sera encadré par Michael Poss.

6. Pièces à fournir

CV + lettre de motivation + classements