

Projet de Complexité

Maël Audren de Kerdrel - Garance Vallat

04.01.15

1 Étude du problème et de sa complexité algorithmique

Le problème de ce projet est appelé problème Bin Packing, avec la particularité d'être en deux dimensions. Dans la partie suivante, on étudiera sa complexité.

Il s'agit d'un problème d'optimisation combinatoire, qui a de nombreuses applications pratiques, notamment pour le transport de matériau.

Complexité algorithmique

Ce problème est similaire au problème de Bin Packing, qui est NP-complet. Ce problème est un problème de Bin Packing à 2 dimensions, en fait une généralisation.

On peut commencer par réduire le problème Partition à Bin Packing, afin de montrer la NP-difficulté.

Voici une présentation du problème sous forme de décision :

DONNÉES :

Ensemble fini d'éléments U ($taille \in N$), une taille $B \in N$ et un nombre $k \in N$.

QUESTION :

Peut-on partitionner U en U_1, U_2, \dots, U_k sans que la somme des tailles des éléments des U_i dépasse B ?

Rappel du problème Partition sous forme de décision

DONNÉES :

Un ensemble fini d'entiers non négatifs A

QUESTION :

Est-ce qu'il existe une partition en A en deux ensembles A' et A'' telle que la somme des éléments de A' soit égale à la somme des éléments de A'' ?

La transformation à effectuer en temps polynomial pour réduire Partition à Bin Packing est la suivante :

Si B est la moitié de la somme des tailles des éléments et $k = 2$, alors un Bin

Packing existe si et seulement si un Partition existe.

Problème de Bin Packing 2D sous forme de décision :

DONNÉES :

Un ensemble fini d'éléments U avec une largeur $L \in N$ et une hauteur $H \in N$. Une dimension de boîte B avec une largeur $J \in N$ et une hauteur $K \in N$, un nombre n .

QUESTION :

Peut-on partitionner U en U_1, U_2, \dots, U_k sans que la somme des longueurs et largeurs des éléments des U_i dépasse la longueur et largeur de B ?

La transformation en temps polynomial pour réduire Bin Packing à Bin Packing 2D est la suivante :

La taille des éléments U = la hauteur H de chaque U . La largeur de chaque $U = 1$. La taille B = la hauteur K de la boîte, et sa largeur $J = 1$ et enfin $n = k$.

Dans ces conditions, on a un Bin Packing si et seulement si on a un Bin Packing 2D. En effet, un Bin packing est un Bin Packing 2D lorsque les tailles de chaque élément U correspondent à la longueur de chaque élément du Bin packing 2D, et la largeur des éléments du Bin Packing 2D est la largeur de la boîte du 2D, soit 1. La longueur de la boîte est la taille B du Bin Packing 1 Dimension.

Il n'existe donc pas d'algorithme exact pour résoudre en temps polynomial ce problème. Cependant, on peut trouver des algorithmes d'approximation.

2 Présentation de notre solution

Algorithme choisi

Nous avons choisi d'utiliser une variante de l'algorithme *First Fit Decreasing*, en se servant d'"étagère" pour remplir les boîtes au fur et à mesure, tout en pouvant y revenir.

Pour cet algorithme, on effectue d'abord une série d'initialisations : On enregistre les dimensions de la boîte, afin de pouvoir librement y revenir, on établit le nombre de boîtes à 0 pour commencer, on crée une structure où on maintiendra une liste des étagères "ouvertes", et on en crée une première, dans la première boîte.

Enfin, on trie de façon décroissante la liste des rectangles, en fonction de leur largeur uniquement.

L'algorithme en lui-même consiste en tester, pour chaque rectangle, s'il entre dans un étage actuellement ouvert. Un rectangle peut entrer dans un étage s'il reste en largeur un espace suffisant pour qu'il s'y insère, et que la hauteur maximum de l'étage le permet aussi.

La hauteur maximum de l'étage est calculée ainsi : si l'étage est le "dernier" au sens du plus haut, de la boîte, sa hauteur actuelle peut être augmentée d'autant que de toute la hauteur restante de la boîte si besoin est.

Références

- [1] Jukka Jylänki, *A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing*. 2010.