

Preuves par ordinateur pour des problèmes de packing online

Antoine Lhomme¹, Nicolas Catusse¹, and Nadia Brauner¹

¹Univ. Grenoble Alpes, CNRS, G-SCOP, Grenoble, 38000 France

Dans les problèmes de packing online, les objets à placer sont reçus progressivement. Un algorithme pour ce type de problèmes doit réaliser un packing sans connaissances sur les futurs objets. Ce manque de connaissances peut rendre la solution sous-optimale. L'objectif est de construire des algorithmes de packing online qui fournissent des solutions garanties d'être proches de la solution optimale.

L'analyse se porte alors sur le pire cas d'algorithmes : on cherche à déterminer des algorithmes online optimaux au sens du pire cas.

Pour ce faire, des méthodes computationnelles peuvent être utilisées. On modélise un problème online sous forme d'un jeu à deux joueurs, l'algorithme et l'adversaire, où l'algorithme a pour but de construire un packing tandis que l'adversaire envoie des objets pour dégrader la qualité de la solution produite par l'algorithme. Ensuite, on cherche à construire des arbres qui modélisent une stratégie pour un des joueurs, et qui construisent également une preuve de borne sur la performance d'un algorithme online optimal.

Ces méthodes computationnelles ont plusieurs intérêts : premièrement, l'analyse du pire cas d'un algorithme est souvent difficile, longue, et doit traiter de nombreux cas différents. Ces méthodes computationnelles permettent de simplifier l'obtention de telles analyses. De plus, les ordinateurs peuvent traiter un très grand nombre de cas pour construire une preuve, ce qui peut permettre d'obtenir des preuves plus fines que celles obtenues « à la main ». Enfin, il est possible de construire des algorithmes online par des recherches computationnelles.

Nous verrons comment fonctionnent de telles méthodes computationnelles dans des cas particuliers, comme dans le problème du bin stretching online (introduit par Azar and Regev [2001]). Ce problème est similaire au bin packing : des objets, définis par leur taille, arrivent l'un après l'autre, et doivent être placés dans m bins. Un objet doit être placé avant que le suivant ne soit révélé. L'objectif est alors de minimiser la charge de la bin la plus remplie après avoir placé tous les objets. De plus, les objets sont garantis de rentrer dans m bins de taille unitaire.

Pendant longtemps, la seule borne inférieure connue sur la performance des algorithmes online pour ce problème était la borne de $4/3$ trouvée « à la main » ; les preuves computationnelles ont permis d'améliorer les bornes inférieures sur la performance des algorithmes (Gabay et al. [2017], Böhm and Simon [2020], Lhomme et al. [2022]), et ont également permis la construction de meilleurs algorithmes (Lieskovský [2022]). La Figure 1 donne un exemple de preuve de borne

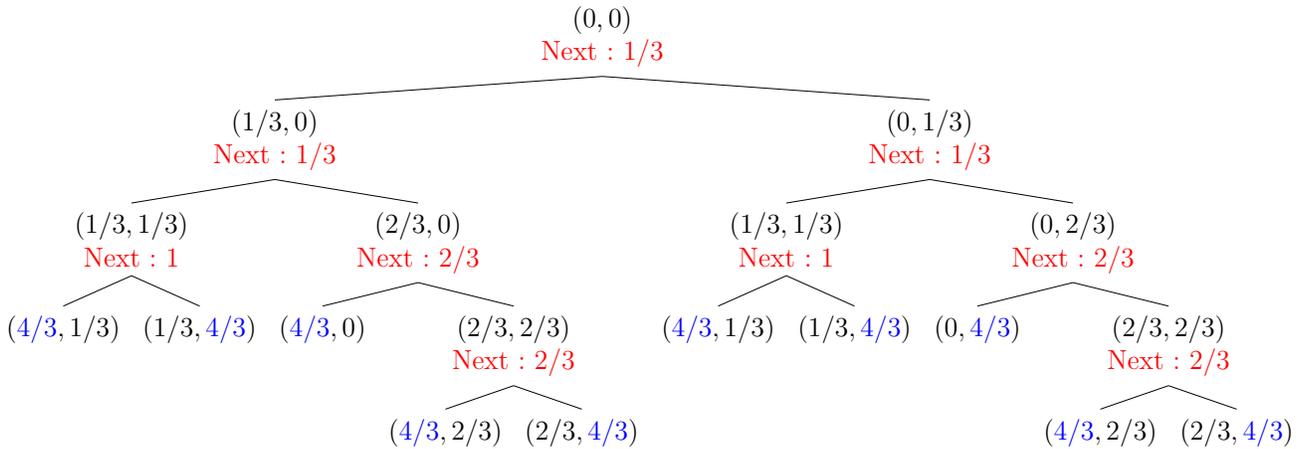


FIGURE 1 – Preuve arborescente de borne inférieure de valeur $4/3$ pour deux bins. En rouge : objet à placer ; entre parenthèse : charge de chaque bin.

inférieure construite par ordinateur pour $m = 2$ bins. Cette preuve arborescente montre que n’importe quel algorithme online, dans le pire cas, remplit un bin à au moins $4/3$, alors qu’il est possible de faire rentrer tous les objets dans deux bins de taille 1.

Références

- Yossi Azar and Oded Regev. On-line bin-stretching. *Theoretical Computer Science*, 268(1) : 17–41, 2001. ISSN 0304-3975. doi : [https://doi.org/10.1016/S0304-3975\(00\)00258-9](https://doi.org/10.1016/S0304-3975(00)00258-9). URL <https://www.sciencedirect.com/science/article/pii/S0304397500002589>. On-line Algorithms ’98.
- Martin Böhm and Bertrand Simon. Discovering and certifying lower bounds for the online bin stretching problem, 2020.
- Michaël Gabay, Nadia Brauner, and Vladimir Kotov. Improved lower bounds for the online bin stretching problem. *JOR*, 15(2) :183–199, Jun 2017. ISSN 1614-2411. doi : 10.1007/s10288-016-0330-2. URL <https://doi.org/10.1007/s10288-016-0330-2>.
- Antoine Lhomme, Olivier Romane, Nicolas Catusse, and Nadia Brauner. Online bin stretching lower bounds : Improved search of computational proofs, 2022. URL <https://arxiv.org/abs/2207.04931>.
- Matej Lieskovský. Better algorithms for online bin stretching via computer search, 2022. URL <https://arxiv.org/abs/2201.12393>.