Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem

François Clautiaux*, Ruslan Sadykov, François Vanderbeck, Quentin Viaud

IMB, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France
INRIA Bordeaux - Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence, France

A R T I C L E I N F O

Article history:
Received 5 July 2017
Received in revised form 1 December 2017
Accepted 13 February 2018
Available online xxxx

Keywords:
Cutting and packing
Dynamic programming
Lagrangian filtering
Reduced-cost fixing

A B S T R A C T

The two-dimensional knapsack problem consists in packing a set of small rectangular items into a given large rectangle while maximizing the total reward associated with selected items. We restrict our attention to packings that emanate from a k-stage guillotine-cut process. We introduce a generic model where a knapsack solution is represented by a flow in a directed acyclic hypergraph. This hypergraph model derives from a forward labelling dynamic programming recursion that enumerates all non-dominated feasible cutting patterns. To reduce the hypergraph size, we make use of further dominance rules and a filtering procedure based on Lagrangian reduced costs fixing of hyperarcs. Our hypergraph model is (incrementally) extended to account for explicit bounds on the number of copies of each item. Our exact forward labelling algorithm is used to solve the max-cost flow model in the base hyper-graph with side constraints to model production bounds. Results of numerical comparison against existing approaches are reported on instances from the literature and on datasets derived from a real-world application.

1. Introduction

The two-dimensional rectangular knapsack problem (2KP) consists in packing or cutting a set of small rectangles, each with a given profit, into a given rectangular sheet in order to maximize the total profit associated with cut pieces. Industrial applications arise when paper, glass, steel, or any other material has to be cut from large pieces of raw material. More formally, we assume a set of small rectangular items $\mathcal{I}$ and a rectangular piece of stock material (stock sheet/plate) of width W and height H. Each item $i \in \mathcal{I}$ has a profit p_i, a width w_i, a height h_i and has a maximum production demand d_i. The problem is to cut items orthogonally from the initial stock sheet in such a way that items do not overlap and the total profit of the

* Corresponding author at: IMB, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France.
E-mail addresses: francois.clautiaux@u-bordeaux.fr (F. Clautiaux), ruslan.sadykov@inria.fr (R. Sadykov), francois.vanderbeck@u-bordeaux.fr (F. Vanderbeck), quentin.viaud@u-bordeaux.fr (Q. Viaud).

https://doi.org/10.1016/j.disopt.2018.02.003

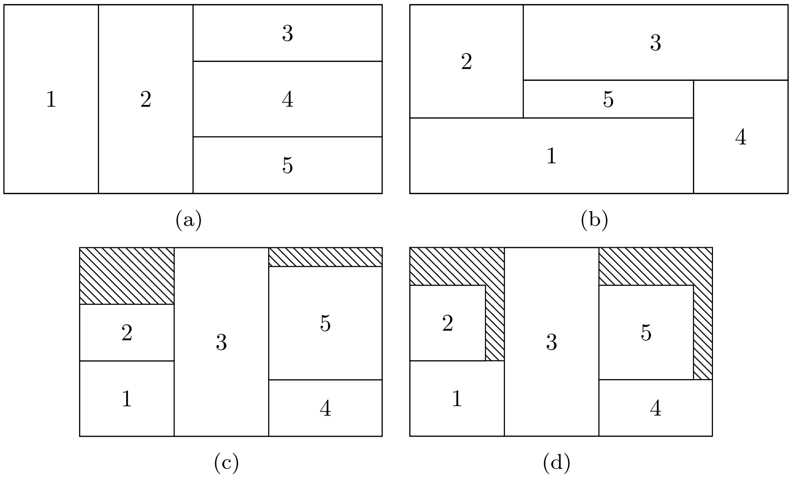

Fig. 1. Guillotine (a) and non guillotine (b) cutting patterns. Exact (c) and non-exact (d) 2-stage guillotine cutting patterns. In configuration (d), an extra cut (trimming) has to be performed to obtain items 2 and 5.

cut items is maximum. This problem is called Single Large Object Placement Problem (SLOPP) according to the typology of Wäscher et al. [1].

The literature reports on several 2KP variants emanating from glass, paper and steel industries as outlined by Gilmore and Gomory [2] and Vanderbeck [3]. The most common of these variants is to perform *"guillotine cuts"* on the stock sheet: cuts go from one edge of the stock sheet to the opposite edge and have to be parallel to an edge of the stock piece (see Fig. 1 for an illustration). Such cut divides an initial plate into two sub-plates or strips. Another typical constraint is to limit the number of cuts needed to produce an item. A problem is called *"k-stage"* when an item has to be cut using at most k successive guillotine cuts. If there are no stage restrictions, the problem is said to be *"any-stage"*. At the last cutting stage of a *"k-stage"* problem variant, if an additional cut is allowed only to separate an item from a waste area, the problem is said to be *"with trimming"* or *"non-exact"* (see Fig. 1). If such extra cut is not allowed, the problem is *"exact"*. To extract items, cuts of different length are performed on the initial sheet. Possible values for cut lengths are in the range $\{1,\ldots,W\}$ and $\{1,\ldots,H\}$. An additional constraint is to restrict the set of possible lengths for a cut. A cut is *"restricted"* if its length is equal to the height h_i or the width w_i of some item $i \in \mathcal{I}$, and item i is immediately cut to initialize one of the two obtained sub-plates. Finally, when there are no upper bounds on the number of times the items can be cut (*i.e.* $d_i = +\infty, \forall i \in \mathcal{I}$), the 2KP is *"unbounded"*, otherwise it is *"bounded"*. Item *"rotation"* can be permitted, by which we mean permuting the value of height and width. For more clarity the following notations are used to characterize the versions of the problem considered:

- C (resp. U) indicates that the demand of each item is bounded (resp. unbounded)
- NR means that cut lengths are non-restricted and non-exact, NRE that cuts are non-restricted and exact, R that cuts are restricted non-exact and RE means restricted exact
- k is an integer which corresponds to the maximum number of stage, while ∞ is associated to the any-stage variant
- f (resp. r) does not permit item rotation (resp. permit item rotation).

For example, using notation C-2KP-RE-4,f (resp. U-2KP-R-∞,r) refers to the bounded restricted exact 4-stage problem variant without item rotation (resp. the unbounded restricted non-exact any-stage problem with item rotation).

The two dimensional knapsack problem is the subject of a large number of scientific papers. Pioneering work goes back to Gilmore and Gomory [2], where a dynamic program for C-2KP-NR-2,f is proposed.

Please cite this article in press as: F. Clautiaux, et al., Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem, Discrete Optimization (2018), https://doi.org/10.1016/j.disopt.2018.02.003.