European Journal of Operational Research 000 (2018) 1–19

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Stabilized branch-and-price algorithms for vector packing problems

Katrin Heßler*, Timo Gschwind, Stefan Irnich

Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9 Mainz D-55128, Germany

ARTICLE INFO

Article history:
Received 9 August 2017
Accepted 27 April 2018
Available online xxx

Keywords:
Cutting
Vector packing
Shortest path problem with resource constraints
Dual-optimal inequalities
Stabilization

ABSTRACT

This paper considers packing and cutting problems in which a packing/cutting pattern is constrained independently in two or more dimensions. Examples are restrictions with respect to weight, length, and value. We present branch-and-price algorithms to solve these vector packing problems (VPPs) exactly. The underlying column-generation procedure uses an extended master program that is stabilized by (deep) dual-optimal inequalities. While some inequalities are added to the master program right from the beginning (static version), other violated dual-optimal inequalities are added dynamically. The column-generation subproblem is a multidimensional knapsack problem, either binary, bounded, or unbounded depending on the specific master problem formulation. Its fast resolution is decisive for the overall performance of the branch-and-price algorithm. In order to provide a generic but still efficient solution approach for the subproblem, we formulate it as a shortest path problem with resource constraints (SPPRC), yielding the following advantages: (i) Violated dual-optimal inequalities can be identified as a by-product of the SPPRC labeling approach and thus be added dynamically; (ii) branching decisions can be implemented into the subproblem without deteriorating its resolution process; and (iii) larger instances of higher-dimensional VPPs can be tackled with branch-and-price for the first time. Extensive computational results show that our branch-and-price algorithms are capable of solving VPP benchmark instances effectively.

1. Introduction

In this paper, we analyze different covering formulations and column-generation-based solution approaches for packing and cutting problems with two or more dimensions. While the 1-dimensional case includes the classical *bin packing problem* (BPP) and *cutting stock problem* (CSP) that are both well studied and recently surveyed by Delorme, Iori, and Martello (2016), we focus on the multidimensional case where packings/cutting patterns are constrained independently in all dimensions. Examples are restrictions with respect to weight, length, and value. In the following, we refer to these problems as *vector packing problems* (VPPs). The literature uses different names for VPPs such as p-dimensional vector (bin) packing (Brandão & Pedroso, 2016; Buljubašić & Vasquez, 2016; Spieksma, 1994), vector bin packing (Panigrahy, Talwar, Uyeda, & Wieder, 2011), or (for two dimensions) two-constraint bin packing (Monaci & Toth, 2006). In contrast to VPPs, there is another family of multidimensional packing problems where dimensions are not independent, e.g., when packing rectangles (Huang & Korf, 2012) or 3-dimensional items (Martello, Pisinger, & Vigo, 2000). These problems are not addressed here.

VPPs have various practical applications. For example, consider a logistics company that has to transport items with different lengths and weights. The smallest amount of vehicles possible should be used for transportation. How can the items be packed into the vehicles taking into account the length of the vehicles and the maximum loading weight? Another example is the static resource allocation problem. Given a set of servers with known capacities and a set of services with known demands the aim is to minimize the number of required servers (Panigrahy et al., 2011).

In the literature, several heuristic and exact methods have been introduced to solve VPPs. The first exact approach was proposed by Spieksma (1994). The author incorporated lower bounds into a branch-and-bound algorithm and, additionally, introduced a heuristic based on the first-fit decreasing heuristic. The approach was improved by Caprara and Toth (2001) who exploited on the one hand lower bounds combined with heuristics and on the other hand a branch-and-bound algorithm to find exact solutions for VPPs. Alves, Valério de Carvalho, Clautiaux, and Rietz (2014) further enhanced the approach by calculating lower bounds based on dual-feasible functions. Recently, Brandão and Pedroso (2016) adapted

* Corresponding author.
E-mail addresses: khessler@uni-mainz.de (K. Heßler), gschwind@uni-mainz.de (T. Gschwind), irnich@uni-mainz.de (S. Irnich).

https://doi.org/10.1016/j.ejor.2018.04.047

the arc-flow formulation with side constraints (Valério de Carvalho, 1999) to packing problems including VPPs. The solution was accelerated by means of graph compression to reduce symmetry and to combine sub-graphs. In contrast, Hu, Zhu, Qin, and Lim (2017) used the covering formulation of Gilmore and Gomory (1961) and introduced a branching strategy based on dominance relations between cutting patterns. Besides exact approaches, heuristics have been applied to solve larger instances. Buljubašić and Vasquez (2016) and Vasquez and Buljubašić (2018) used consistent neighborhood search to find heuristic solutions. Variants of the first-fit decreasing algorithm were presented by Panigrahy et al. (2011), yielding good heuristic VPP solutions.

We formally define the VPP as follows: Let $D \geq 1$ be an integer specifying the dimension. A given set $\mathcal{I}$ of items needs to be packed into a minimum number of equally sized bins. Bins are characterized by capacities given by a D-dimensional vector $\mathbf{W} = (W^1, \ldots, W^D)$. Items $i \in \mathcal{I}$ are characterized by weights/sizes which are given by D-dimensional vectors $\mathbf{w}_i = (w_i^1, \ldots, w_i^D) \leq \mathbf{W}$. Dyckhoff (1990) classified cutting and packing problems according to the categories dimensionality, kind of assignment, assortment of large objects (bins), and assortment of (small) items. Hence, the BPP is classified as 1/V/I/R (1): 1-dimensional/V: a selection of objects (bins) and all items/I: identical figure/R: many items of relatively few different (non-congruent) figures). In contrast, the CSP is classified as 1/V/I/M (M: many items of many different figures). Although classified differently in Dyckhoff (1990), the BPP and the CSP are essentially the same problem as pointed out in Ben Amor and Valério de Carvalho (2005, p. 132). The relationship between BPP and CSP is the following: In the BPP, all items are modeled as *individual objects*. As a consequence, each and every item $i \in \mathcal{I}$ has a demand of $q_i = 1$ and a model must include a corresponding covering/packing constraint. In contrast, in the CSP all items of the same weight are *aggregated*. We formally introduce the set of aggregated items as follows: Let $\mathcal{I}[\mathbf{w}]$ be the equivalence classes of items with identical weight $\mathbf{w}$. Then, the set I of *aggregated items* is the coset $\mathcal{I}/\mathcal{I}[\mathbf{w}]$ that contains one representative item for each of the different weights. As a consequence, any two different aggregated items $i_1, i_2 \in I$ have different weights $\mathbf{w}_{i_1} \neq \mathbf{w}_{i_2}$. The demand of an aggregated item $i \in I$ is $q_i = |\mathcal{I}[\mathbf{w}_i]| = |\{j \in \mathcal{I} : \mathbf{w}_j = \mathbf{w}_i\}| \geq 1$, and at least some aggregated items have a demand greater than 1.

The models that we compare are all variations of the famous covering formulation of Gilmore and Gomory (1961) for CSP that is based on cutting patterns. For the VPP, a *pattern* (or *packing*) describes how a subset of the items is packed into a bin. Assuming $I = \{1, 2, \ldots, m\}$, i.e., m is the number of aggregated items, the set of feasible patterns for the VPP, with different levels of aggregation, can be defined as

$$P = \left\{ (a_1, \ldots, a_m)^\top \in \mathbb{Z}_+^m : \sum_{i=1}^m a_i w_i^d \leq W^d \text{ for all } 1 \leq d \leq D \right\}. \quad (1)$$

In order to uniquely refer to a pattern $p \in P$ we write its coefficients as $\mathbf{a}^p = (a_1^p, \ldots, a_m^p)^\top$. Using integer decision variables x_p for the number of times pattern $p \in P$ is used, the VPP can be formulated as

$$z^{\text{VPP}} = \min \sum_{p \in P} x_p \quad (2a)$$

$$(\text{VPP}) \qquad \text{s.t.} \sum_{p \in P} a_i^p x_p = q_i, \qquad i \in I \quad (2b)$$

$$x_p \geq 0 \text{ integer}, \qquad p \in P. \quad (2c)$$

The objective (2a) is the minimization of the patterns/bins that are used. Constraints (2b) ensure that all items are packed as often as their demand requires. As already mentioned by Gilmore and Gomory (1961), (2b) can be replaced by covering constraints, i.e., $\sum_{p \in P} a_i^p x_p \geq q_i$ for all $i \in I$. The domain of the decision variables is given by (2c).

The quality of the linear relaxation bound is crucial to solve mixed-integer problems. As in the BPP (i.e., $D = 1$ and $q_i = 1$ for all $i \in I$), patterns can be restricted to only have binary coefficients $\mathbf{a}^p \in \{0, 1\}^m$ when modeling individual items. While such a constraint does not impact the validity of integer VPP solutions to (2) (also for $D \geq 2$), the linear relaxation is generally tighter than without the binary requirement. It means that a proper subset of patterns is used:

$$P_{01} = \left\{ (a_1, \ldots, a_{m_{01}})^\top \in \mathbb{Z}_+^{m_{01}} : \sum_{i=1}^{m_{01}} a_i w_i^d \leq W^d \right.$$
$$\left. \text{for all } 1 \leq d \leq D \text{ and } a_i \leq 1 \text{ for all } i \in I \right\} \quad (3)$$

We denote formulation (2) using only binary patterns P_{01} as *binary* VPP (01-VPP). In this case, optimal solutions to the 01-VPP have binary x_p-variables.

Also for VPP with non-unit demand, the patterns' coefficients can be further constrained. Whenever the demand q_i of some item $i \in I$ is smaller than $\lfloor W^d / w_i^d \rfloor$ for all $d \in D$, the pattern set

$$P_B = \left\{ (a_1, \ldots, a_{m_B})^\top \in \mathbb{Z}_+^{m_B} : \sum_{i=1}^{m_B} a_i w_i^d \leq W^d \right.$$
$$\left. \text{for all } 1 \leq d \leq D \text{ and } a_i \leq q_i \text{ for all } i \in I \right\} \quad (4)$$

is a proper subset of P. We denote formulation (2) using only bounded patterns P_B as *bounded* VPP (B-VPP). For completeness, formulation (2) with no additional bounds on pattern coefficients is referred to as *unbounded* VPP (U-VPP) and the pattern set is explicitly denoted by $P_U = P$ in the following.

Table 1 summarizes differences between the pure binary formulation 01-VPP, the bounded formulation B-VPP, and the unbounded formulation U-VPP. Note that for a given set of individual items $\mathcal{I}$, the formulations 01-VPP and U-VPP are unique. They represent the two extremes of aggregation (completely disaggregated vs. fully aggregated), while B-VPP is a family of formulations resulting from different types of aggregation and exploitation/disregarding of the coefficient bounds $a_i^p \leq q_i$ in (4).

Due to the huge number of variables, formulation (2) is typically solved with the help of a column-generation algorithm (Desaulniers, Desrosiers, & Solomon, 2005). One starts with a (small) subset of patterns $P' \subset P$ and the linear relaxation of (2) using only variables x_p with $p \in P'$. This so-called restricted master program (RMP) is then optimized. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)^\top$ be the dual solution w.r.t. the covering constraints (2b). The task of the pricing subproblem is then to provide (at least) one new pattern p with negative reduced cost $\tilde{c}_p = 1 - \boldsymbol{\pi}^\top \mathbf{a}^p$ or to prove that no such pattern exists. In the former case, the resulting RMP (with the generated pattern/s) is re-optimized and the column-generation process is repeated. In the latter case, the linear relaxation of (2) is solved providing a lower bound z_{LP}^{VPP} on z^{VPP}. An optimal integer solution to (2) requires the integration of column generation into a branch-and-bound scheme a.k.a. branch-and-price (Lübbecke & Desrosiers, 2005).

The pricing problems of the different formulations 01-VPP, B-VPP, and U-VPP seek to find a pattern with negative reduced cost $\tilde{c}_p$ that is feasible for the pattern set considered in the respective formulation. They are equivalent to solving one of the following