Computers & Operations Research 75 (2016) 202–213

Contents lists available at ScienceDirect

Computers & Operations Research

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

Exactly solving packing problems with fragmentation

Marco Casazza, Alberto Ceselli *

Università degli Studi di Milano, Dipartimento di Informatica, Via Bramante 65, 26013 Crema (CR), Italy

ARTICLE INFO

Article history:
Received 13 April 2015
Received in revised form
10 June 2016
Accepted 10 June 2016
Available online 15 June 2016

Keywords:
Bin packing
Item fragmentation
Mathematical programming
Branch-and-price

ABSTRACT

In packing problems with fragmentation a set of items of known weight is given, together with a set of bins of limited capacity; the task is to find an assignment of items to bins such that the sum of items assigned to the same bin does not exceed its capacity. As a distinctive feature, items can be split at a price, and fractionally assigned to different bins. Arising in diverse application fields, packing with fragmentation has been investigated in the literature from both theoretical, modeling, approximation and exact optimization points of view.

We improve the theoretical understanding of the problem and we introduce new models by exploiting only its combinatorial nature. We design new exact solution algorithms and heuristics based on these models. We consider also variants from the literature with different objective functions and the option of handling weight overhead after splitting. We present experimental results on both datasets from the literature and new, more challenging, ones. These show that our algorithms are both flexible and effective, outperforming by orders of magnitude previous approaches from the literature for all the variants considered. By using our algorithms we could also assess the impact of explicitly handling split overhead, in terms of both solutions quality and computing effort.

1. Introduction

Logistics has always been a benchmark for combinatorial optimization methodologies, as practitioners are traditionally familiar with the competitive advantage granted by optimized systems. As an example, Vehicle Routing Problems (VRP) [6] are popular since decades for operational planning. Indeed, as more problems are understood from a computational point of view, more details are required to be included in state-of-the art models, driving research for new solution methods in a virtuous cycle.

The Bin Packing Problem with Item Fragmentation (BPPIF) has been introduced to provide such an additional level of details. As in the traditional Bin Packing Problem (BPP), a set of items of known weight is given, together with a set of bins of limited capacity; the task is to find an assignment of items to bins such that the sum of items assigned to the same bin does not exceed its capacity. However, unlike BPPs, items can be split at a price, and fractionally assigned to different bins.

The BPPIF has first been introduced to model message transmission in community TV networks, VLSI circuit design [17] and preemptive scheduling on parallel machines with setup times/setup costs. It has also been employed in fully optical network planning problems [24]: connection requests are given, that can be split among different transmission channels to fully exploit bandwidth, but each split is known to introduce delays and loss of energy in the transmission process.

The BPPIF arises also as tactical counterpart of the Split Delivery Vehicle Routing Problem (SDVRP) [2] which extends VRP models by allowing the demand at each customer to be split among multiple vehicle visits to better exploit vehicle capacities and decrease the routing costs: BPPIF models allow us to estimate the minimum fleet size that is required to perform all deliveries.

We also mention that, from a methodological point of view, there is currently a strong concern in tackling problems in which different kinds of decisions need to be taken, some being combinatorial while others being continuous in nature. It is the case, for instance, when simultaneously routing and planning the recharge of electrical vehicles [23] or when rebalancing the location distribution of bicycles in bike sharing systems [15]. The BPPIF is one of the most fundamental problems in this family.

Many variants of BPPIF have been discussed in the literature. In fact, for what concerns capacity consumption, two versions of the BPPIF can be found: the simpler BPPIF with Size Preserving Fragmentation (BPPSPF) where the weight of each fragment remains constant, and the BPP with Size Increasing Fragmentation (BPPSIF) where the weight of each fragment is increased by a certain amount after splitting. With regard to the objective function, the BPPIF arises in the literature in both fragmentation–minimization form, where an upper limit on the total number of bins is imposed,

* Corresponding author.
E-mail addresses: marco.casazza@unimi.it (M. Casazza), alberto.ceselli@unimi.it (A. Ceselli).

http://dx.doi.org/10.1016/j.cor.2016.06.007
0305-0548/

Table 1
Variants of BPPIF addressed in the literature.

	Fragmentation–minimization (fm)	Bin-minimization (bm)
Size-preserving (SP)	fm-BPPSPF [20,4]	bm-BPPSPF [26,25,5,12,11]
Size-increasing (SI)	fm-BPPSIF [17]	bm-BPPSIF [20,21,26,25]

and bin-minimization form where an upper limit on the total number of fragmentations is imposed. A full list of the variants addressed in the literature, and their corresponding references, is reported in Table 1. Indeed, such a list of variants is still far from covering all practical applications of BPPIF. For instance, modeling of cloud storage services may require the handling of heterogeneous bins.

From a computational complexity point of view, the BPPIF is NP-Hard [17]. It has first been tackled in [20,21] where the approximation properties of traditional BPP heuristics are discussed. In [25,26], the authors present fast and dual asymptotic fully polynomial time approximation schemes. Similar models have been introduced in the context of memory allocation problems by [5] and considered in [12]: BPPs are presented in which items can be split, but each bin can contain at most k item fragments; the theoretical complexity is discussed for different values of k, and simple approximation algorithms are given. Such results have been refined in [11], where the authors provide efficient polynomial-time approximation schemes, and consider also dual approximation schemes. Approximation algorithms for the fm-BPPSPF have been recently proposed in [16].

BPPs are in general appealing benchmarks for decomposition and column generation algorithms [13]: surveys like [8] contributed to make packing models and algorithms popular in the operations research community. State-of-the-art decomposition algorithms can now successfully tackle involved BPPs [9,19]. Indeed, we previously tackled a particular BPPIF in which a fixed number of bins are given, and a solution needs to be found that minimizes the number of item fragmentations [4]. We proposed a branch-and-price algorithm that allows us to solve instances with up to 20 items in one hour of computing time.

In this paper we address BPPIFs with homogeneous bins. We first investigate about further BPPIF properties that yield compact models without fractional variables. We design both heuristics and new exact algorithms relying on these models that (a) are more flexible, as they can be applied to all BPPIF variants described above, including both bin-minimization and fragmentation–minimization objectives, and both size preserving and size increasing variants and (b) are more effective, since they are orders of magnitude faster when applied to the variant discussed in [4]. Exploiting our new tools, we also present a computational comparison on BPPIF models, assessing the impact of overhead handling on solution values and computing hardness. Preliminary results were presented in [3].

For the ease of exposition, in Section 2 we consider the bin-minimization BPPSPF, its mathematical programming model and a few theoretical properties on the structure of optimal solutions, and in Section 3 we detail our new exact algorithm to solve it. Then, in Section 4 we discuss how to extend it in order to tackle all the BPPIF variants listed above. In Section 5 we present our experimental analysis. Finally, in Section 6 we summarize our results and present some brief conclusions.

2. Models

In the following we formalize the *bin-minimization BPPSPF* (*bm-BPPSPF*). Then, we recall and exploit some properties on the structure of optimal solutions to get an improved formulation that avoids the use of fractional variables, thus yielding better computational behavior. We also discuss dominance between bounds. To clearly distinguish theorems reported in the literature from new ones, the former are always marked with their reference at the beginning of the claim.

2.1. Mathematical formulation

We are given a set of items I and a set of bins B. Let w_i be the weight of each item $i \in I$ and let C be the capacity of each bin. Each item has to be fully packed, but may be split into fragments and fractionally assigned to different bins. The sum of the weights of the (fragments of) items packed into a single bin must not exceed the capacity C.

The bm-BPPSPF can be stated as the problem of packing all the items into the minimum number of bins by performing at most F fragmentations. It can be formalized as follows:

$$\min \sum_{j \in B} u_j \tag{2.1}$$

$$\text{s. t.} \sum_{j \in B} x_{ij} = 1 \quad \forall\, i \in I \tag{2.2}$$

$$\sum_{i \in I} w_i \cdot x_{ij} \le C \cdot u_j \quad \forall\, j \in B \tag{2.3}$$

$$(BM) \sum_{i \in I, j \in B} z_{ij} - |I| \le F \tag{2.4}$$

$$x_{ij} \le z_{ij} \quad \forall\, i \in I, \forall\, j \in B \tag{2.5}$$

$$0 \le x_{ij} \le 1 \quad \forall\, i \in I, \forall\, j \in B \tag{2.6}$$

$$z_{ij} \in \{0, 1\} \quad \forall\, i \in I, \forall\, j \in B \tag{2.7}$$

$$u_j \in \{0, 1\} \quad \forall\, j \in B \tag{2.8}$$

where each variable x_{ij} represents the fraction of item i packed into bin j, each binary variable z_{ij} is 1 if any fragment of item i is packed into bin j, and each binary variable u_j is 1 if bin j is used, that means j is allowed to contain items and fragments.

The objective function (2.1) minimizes the number of used bins. Constraints (2.2) ensure that each item is fully packed. Constraints (2.3) have a double effect: they forbid the assignment of items to bins that are not used, and ensure that the capacity of each used bin is not exceeded. Constraints (2.5) enforce consistency between variables, so that no fragment x_{ij} of each item i is packed into bin j unless z_{ij} is set to 1. Constraint (2.4) ensures that the packing is performed with at most F fragmentations. In fact, as observed in [4]:

Observation 2.1 (*[4]*). *Given any BPPIF solution, the number of fragmentations is equal to the overall number of fragments minus the number of items.*

We first observe that:

Observation 2.2. *If an item $i \in I$ has weight $w_i > C$, then an optimal bm-BPPSPF solution can always be obtained by splitting i and*