Theoretical Computer Science 602 (2015) 50–59

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Bin packing with fragmentable items: Presentation and approximations

Bertrand LeCun [a], Thierry Mautor [a,*], Franck Quessette [a], Marc-Antoine Weisser [b]

[a] Laboratoire PRiSM, Université de Versailles-Saint Quentin, Versailles, France
[b] Ecole Supélec, Gif sur Yvette, France

ARTICLE INFO

Article history:
Received 26 March 2014
Accepted 4 August 2015
Available online 3 September 2015
Communicated by A. Fiat

Keywords:
Approximation algorithm
Bin-packing
Combinatorial optimization

ABSTRACT

We consider a variant of the Bin Packing Problem dealing with fragmentable items. Given a fixed number of bins, the objective is to put all the items into the bins by splitting them in a minimum number of fragments. This problem is useful for modeling splittable resource allocation. In this paper we introduce the problem and its complexity. We give and prove several properties then we present various approximation algorithms and specially a $\frac{6}{5}$-approximation algorithm.

1. Introduction

The bin packing problem is a widely studied problem. In its standard form, many theoretical results have been proposed (see for instance [1–3]) as well as many approximation algorithms [4,5].

In this paper, we study a new variant in which the number of bins is fixed and items are fragmentable without additional space. The goal is there to put all the items into the bins by splitting them in a minimum number of fragments. This variant is useful to model splittable resource allocation such as the two following applications.

The first application deals with settling accounts. A group of friends goes in holidays. Common charges, such as food, locations... are paid by some people of the group and recorded in a list. At the end of the holidays the friends settle accounts. Some of them have paid more and need to be refunded by those who have paid less. Obviously, they would like to make a minimum of money transfers. In such an application, a bin corresponds to a member of the group who has to be refunded while an item is a member who has to pay. The total size of items is equal to the total space of bins. Finally, each money transfer (a cheque for instance) corresponds to one fragment. Remark that in this application the sizes of the bins are not equal. This application will be modeled by a problem called Min-FIBP (Minimum Fragmentable Items Bin Packing).

The second application concerns the wavelength assignment in optical networks using POADM (Packet Optical Add/Drop Multiplexer) technology [6,7]. In such networks, a node is able to send traffic on any wavelength but it could read traffic on a limited number of wavelengths. The least wavelengths are read, the least energy is consumed. The problem is to assign the traffic sent by nodes to wavelengths in a way which allows a receiver node to read a minimum number of wavelengths. As it can be seen on Fig. 1, this application also corresponds to a bin packing with fragmentable items. The items are the

* Corresponding author.
E-mail addresses: bertrand.lecun@prism.uvsq.fr (B. LeCun), thierry.mautor@prism.uvsq.fr (T. Mautor), franck.quessette@prism.uvsq.fr (F. Quessette), marc-antoine.weisser@supelec.fr (M.-A. Weisser).

http://dx.doi.org/10.1016/j.tcs.2015.08.005
0304-3975/© 2015 Published by Elsevier B.V.

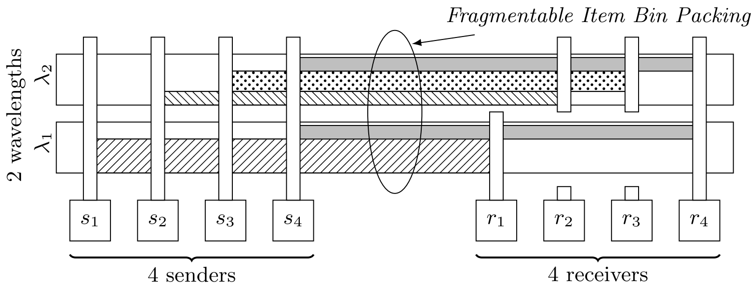

Fig. 1. Four nodes send traffic to four receiver nodes. Traffic sent from node s_4 to r_4 (represented in light gray) is split on two wavelengths (λ_1 and λ_2). r_1, r_2 and r_3 read one wavelength but r_4 have to read two wavelengths. By switching the traffic from s_2 on the first wavelength, we can assign all traffic from r_4 to the second wavelength. Then r_4 reads only one wavelength.

couples of nodes (sender node, receiver node) with the corresponding quantity of traffic while the bins are the wavelengths. Here, the total size of the items could differ from the total space of the bins but, on another hand, all the bins have the same capacity (capacity of a wavelength) that was not the case in the first application. This specific version of the problem with identical bins will be called MIN-FIBP-EQ.

As it can be seen in the following section on related works, many problems similar to MIN-FIBP have already been studied but none of them perfectly fits to the two applications described above. Section 3 describes the two versions of the problem and proves their complexity. Section 4 presents some general properties. Section 5 describes algorithms and particularly a $\frac{6}{5}$-approximation. Finally, Section 6 concludes this paper by summarizing its main achievements and discussing further work.

2. Related works

The problem defined in this paper is similar to the classical bin packing problem and some of its variants. However the existing results and algorithms are not adapted to the version with fragmentable items. To the best of our knowledge, MIN-FIBP and MIN-FIBP-EQ have not been studied and the results and algorithms presented in this paper are original.

The classical bin packing problem (BP), [1], seems to be really similar to MIN-FIBP-EQ since the entries of both problems are the same (a set of items and the size of the bins). However solutions are very different and algorithms used to solve BP are not well adapted to MIN-FIBP-EQ. For example, the *Best Fit Decreasing* and *First Fit Decreasing* strategies are classical and simple heuristic approaches used for the BP. For BP, they constitute some good heuristics with good approximation rates. However, they are not good strategies for MIN-FIBP-EQ. The reason is simply that to fill some bins at – say 99% – is a very good result for the BP while it is generally a bad thing for MIN-FIBP-EQ leading to fragmentations of items on these bins.

Of course, the difference between BP and MIN-FIBP-EQ mainly comes from the cost function. In the first problem we are interested in the number of bins and in the second one in the number of cuts. However, some variants of BP, introducing cuts of items, have been proposed. For example, they allow to model a problem of VLSI Circuit design [8]. In this variant, it is possible to split an item into two or more fragments. But each fragment needs an additional space to be packed and the objective is to minimize the number of bins used. Once again solutions for this problem could be bad for MIN-FIBP-EQ.

In [9], the application is a scheduling problem in CATV (Community Antenna TeleVision) networks. Again, it is possible to fragment items and two variants are considered. In the first one, as in [8], overhead units are added to the size of each fragment. The second concerns the bin packing problems where a cost function has to be minimized (the processing time or the reassembly delay in this scheduling application). Each fragmentation induces an extra cost in this case.

A problem related to MIN-FIBP can also be found in wireless networks but the problem is studied in two dimensions [10]. Due to its complexity, it has been solved using simple heuristic algorithms without performance guarantee. To the best of our knowledge no other work deals with this variant of the bin packing problem with fragmentable items.

To conclude, proximity may be found between MIN-FIBP and preemptive scheduling [11] where the tasks could be cut and placed on different processors. We can consider items as tasks and bins as processors. But in fact, the problems are quite different. The objectives are not the same (minimization of time for scheduling problems) and specific precedence constraints exist for scheduling where for instance two different parts of a same task cannot be treated during the same time.

3. Presentation and complexity of MIN-FIBP and MIN-FIBP-EQ

The Minimum Fragmentable Items Bin Packing Problem is defined by the set I of items and the set B of bins. Each item $i \in I$ has an integer size s_i and each bin $j \in B$ has an integer capacity c_j. We consider that there is place enough into the bins for all the items: $\sum_{i \in I} s_i \leq \sum_{j \in B} c_j$. Each item i is allowed to be split into multiple smaller items of integer size, called *fragments*, whose the sum of sizes is equal to s_i (for consistency, an uncut item is also considered as a fragment) and each fragment has to be placed in one single bin. The objective is to minimize the total number of fragments. As already