Bin Packing with Fragmentable Items: Presentation and Approximations

Bertrand Lecun, Thierry Mautor, Franck Quessette, Marc-Antoine Weisser

Bin Packing with Fragmentable Items: Presentation and Approximations

Bertrand LeCun[1], Thierry Mautor[1], Franck Quessette[1], Marc-Antoine Weisser[2]

[1] Laboratoire PRiSM, Université de Versailles - Saint Quentin, Versailles, France
{bertrand.lecun,thierry.mautor,franck.quessette}@prism.uvsq.fr

[2] Ecole Supélec, Gif sur Yvette, France
Marc-Antoine.Weisser@supelec.fr

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 then we present a $\frac{6}{5}$-approximation algorithm for a special case in which all bins have the same capacities.

1 Introduction

The bin packing problem is a widely studied problem. In its standard form, many results have been proposed (see for instance [3], [8], [10]) as well as on approximation algorithms for this problem [1], [11]. A variant of this problem has been proposed to model a problem of VLSI Circuit design [2]. In this variant, it is allowed to split an item into two or more fragments. Each fragment needs an additional space to be packed and the objective is to minimize the number of bins used. In [12], 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 [2], 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 their scheduling application). Each fragmentation induces an extra cost in this case.

In this paper, we study a new variant in which the number of bins is fixed and items are fragmentable but without additional space. The goal is there to put the items into the bins by splitting them in a minimum number of fragments. This variant is usefull to modelize 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 payed 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 payed more and need to be refunded by those who have payed less. Obviously, they would like to make a minimum of money transferts. In such an application, a bin corresponds 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 transfert (a cheque for instance) corresponds to one fragment.

The second application is the wavelength assignment. In optical networks using POADM (Packet Optical Add/Drop Multiplexer) technology [5], [4], a node is able to send trafic on any wavelength but it could read trafic on a limited number of wavelengths. The least wavelengths are read, the least energy is consumed. The problem is to assign the trafic sent by node to wavelengths in a way which allows a receiver node to read a minimum number of wavelengths. As it can be seen on Fig. 2, this application also corresponds to a bin packing with fragmentable items. The items are the transmitting nodes with the quantity of trafic they have to send 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 that was not the case in the first application.

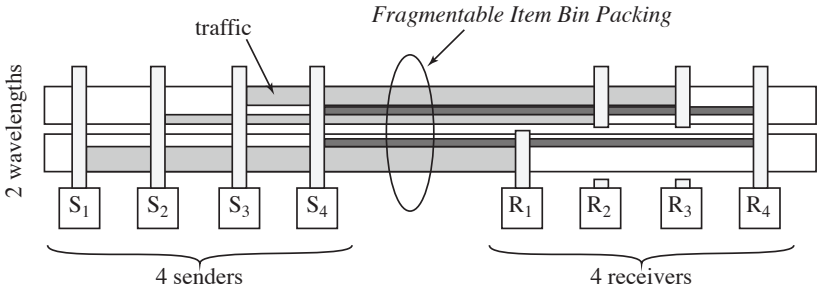

Figure 1: Four nodes send different amount of traffic to four receiver nodes. Traffic sent from node S_4 to R_4 is split on two wavelengths. R_1, R_2 and R_3 read one wavelength but R_4 have to read two wavelengths. By switching the trafic from S_2 on the first wavelength, we can assign all trafic from R_4 to the second wavelength. Then R_4 reads only one wavelength and saves power.

A related problem can be found in wireless networks but the problem is studied in two dimensions [7]. 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.

The remainder of this paper is organized as follows: Section 2 describes the problem, proves its complexity and presents some general properties; Section 3 describes algorithms and particulary a $\frac{6}{5}$-approximation. Finally, Section 4 con-

cludes this paper by summarizing its main achievements and discussing further work.

2 Bin Packing with Fragmentable Items

2.1 Presentation and modelisation of the problem

The Fragmentable Item Bin Packing Problem, FIBP, is a bin packing problem in which each item e is allowed to be splitted into multiple smaller items, called *fragments*, whose the sum of sizes is equal to the size of e. An uncut item is also be considered as a fragment. Let us mention that, in the paper, we will say either that an item is fragmented or that it is cut.

Let us consider the following notations:

- We have N items, the i^{th} ($i \in [1..N]$) of these items has a size s_i.
- We have M bins, the j^{th} ($j \in [1..M]$) of these bins has a capacity c_j. If the capacities of all the bins are equal, the unique capacity is denoted as C.

Moreover, we consider that the total size of the items is lower than the total capacity of the bins:
$\sum_{i=1}^{N} s_i \leq \sum_{j=1}^{M} c_j$ (*or* $M.C$)

This means that there is capacity enough for all the items into the bins. The difficulty is just to minimize the number of fragments when placing these items into the bins.

In order to give a clear formulation of the problem, we propose a possible mathematical model under the form of a Linear Program:

$$
\begin{array}{lll}
Minimize & (\sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij}) & (1) \\
so\ that: & \sum_{j=1}^{M} x_{ij} = s_i \quad \forall i \in [1..N] & (2) \\
 & \sum_{i=1}^{N} x_{ij} \leq c_j \quad \forall j \in [1..M] & (3) \\
 & \frac{x_{ij}}{s_i} \leq y_{ij} \quad \forall i \in [1..N],\ \forall j \in [1..M] & (4) \\
 & x_{ij} \in \mathbb{N},\ y_{ij} \in \{0,1\} \quad \forall i \in [1..N],\ \forall j \in [1..M] & (5)
\end{array}
$$

The variable x_{ij} represents the size of the fragment of the item i put in the bin j. We consider this variable as integer (included between 0 and s_i) (see constraints (5)). Constraints (2) mean that the sum of the sizes of the fragments of any item i must be equal to s_i: the item i must be entirely put in the different bins.
The meaning of constraints (3) is that the sum of the sizes of the fragments put in any bin j must not exceed the capacity c_j of this bin.

As soon as a part of the item i is present in the bin j (even a very small part), y_{ij} is equal to 1. Otherwise y_{ij} is equal to 0. So, y_{ij} is a bivalent variable,

equal to 0 or 1 (see constraints (5)) that detects the presence of item i in bin j. Constraints (4) force the variable y_{ij} to be equal to 1 as soon as the variable x_{ij} is strictly greater than 0.
The sum of all the variables y_{ij} corresponds to the total number of fragments of items. It is this quantity that we want to minimize (function (1)).

2.2 Complexity

Let us now present the problem FIBP under the following decision form : given a set of M bins of capacity $c_j \in \mathbb{N}^+$, N items of size $s_i \in \mathbb{N}^+$ (with $\sum_{i=1}^{N} s_i \leq \sum_{j=1}^{M} c_j$) and an integer $K \geq N$, does-it exist a packing of the N items into the M bins such that the number of fragments is less than K?

Theorem 1. FIBP *is strongly* NP*-complete.*

Proof. Clearly, FIBP belongs to *NP*. Moreover the most classical version of the bin packing problem is a sub problem of FIBP in which all bins have the same size and maximum number of fragment allowed is $K = N$ (no fragmentation). The bin packing problem is strongly *NP*-complete [6], so FIBP is strongly *NP*-complete.

□

Min-FIBP, the minimization problem associated to FIBP consists in minimizing the total number of fragments.

2.3 Some properties

We now present different easy but interesting properties on optimal solutions. We begin with properties true for any instances and we will pursue with properties only true for instances with identical bins.

Property 1. *The value of the optimal solution is at most equal to* $N + M - 1$ *(with* M *the number of bins and* N *the number of items).*

Proof. An obvious heuristic consists in sequentially placing the items in any order into the bins also taken in any order. If the current item cannot be entirely put in the current bin, it is cut in two fragments, the first one to complete the current bin and the second starts the next bin. This heuristic provides a solution (and thus an upper bound) where there are at most $M - 1$ cuts. We then have at most $N + M - 1$ fragments. □

Property 2. *For some instances, this upper value is reached. The optimal solution is sometimes equal to* $N + M - 1$.

Proof. Let us consider the following instance:

- We have M identical bins of capacity $C = M + 1$.
- We have $M + 1$ identical items of size $s = M$.

There is no way to put a subset of entire items in k bins with $k < M$. With the exception of the last one, each time a new bin is filled, at least one cut has to be made. The optimal solution is equal to $N + M - 1$. An example of such an instance is given in Figure 2. □

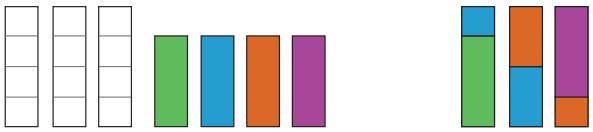

Figure 2: On the right an instance (three bins and four items), on the right an optimal solution with 2 cuts.

Property 3. *It is always possible to consider that the sum of the sizes of the items is equal to the sum of the capacities of the bins.*

Proof. It has been said in the description of the problem that the total size of the items is considered lower than the total capacity of the bins. If the total size of the items is strictly lower than the total capacity of the bins, it is possible to add $(\sum_{j=1}^{M} c_j - \sum_{i=1}^{N} s_i)$ items of size 1 corresponding to holes. Thus, we obtain an equivalent problem with more items but with the equality between total sizes of items and bins. □

From now on, we model a bin of capacity C as a linear bin with elementary parts numered from 1 to C. With such a graduation, we can define a part of a bin as an interval $[a, b]$ of size $b - a + 1$.

Property 4. *If the size of an item i is equal to the capacity of a bin j ($s_i = c_j$), there is at least one optimal solution in which this item i is entirely placed in this bin j. In such a case, it is possible to reduce the problem by deleting this item and this bin. The reduced problem contains $N - 1$ items and $M - 1$ bins.*

Proof. Assume an optimal configuration with item i splitted in k bins with one fragment per bin of respective sizes f_1, ..., f_k. First, cut items in bin j in k contiguous parts $[1, f_1]$, $[f_1 + 1, f_1 + f_2]$, ..., $[f_{k-1} + 1, f_k]$. Each part has the size of one fragment of item i. In the worst case, this step adds $k - 1$ cuts to the solution. Second, each part of bin j is exchanged with the corresponding fragment of item i of same size. This step groups all the fragments of item i in bin j and withdraws $k - 1$ fragments. Finally, the number of fragments is unchanged and item i completly fills bin j.

□

We have now properties that are true when all the bins have the same capacity but that can be false when it is not the case.

We first define a basic *fragment-exchange* operation as follows:

- Let us assume that an item i has a fragment of size f_1 in bin j and a fragment of size f_2 in bin k and that the capacity of bin j is greater than the total size of these two fragments: $c_j \geq f_1 + f_2$.
- Reorganise bin j to have the fragment of size f_1 in position $[1, f_1]$ and exchange part $[f_1 + 1, f_1 + f_2]$ of bin j with the fragment of size f_2 in bin k.
- Let us remark that this *fragment-exchange* operation may add a new cut at position $f_1 + f_2$ but, on another hand, the concatenation of the two fragments of item i in bin j reduces by 1 the number of cuts.

Property 5. *In the case where all the bins have the same capacity C, if the sum of the sizes of two items k and l is equal to C, there is at least one optimal solution in which these items k and l are placed in the same bin. In such a case, it is possible to reduce the problem by deleting these two items and a bin. The reduced problem contains $N - 2$ items and $M - 1$ bins.*

Proof. Assume an optimal configuration. Choose a bin i with one fragment of item k. Make a *fragment-exchange* operation for each other fragment of item k. At the end of these operations, the configuration is still optimal and item k is completly in bin i in position $[1, s_k]$. Assume that item l does not have any fragment in bin i. Choose a fragment of item l, let say of size f and exchange this fragment with part $[s_k + 1, s_k + f]$. This operation could add one cut. Make a *fragment-exchange* operation for each other fragment of item l except one. These operations do not change the number of cuts. Finally for the last fragment, the *fragment-exchange* operation reduces the number of cuts by one. At the end the number of cuts is unchanged and the two items k and l completly fill the bin. We can note that, if item l has a fragment in bin i at the beginning, the construction reduces by one the number of cuts. It is not possible since the original configuration is optimal. □

Remark that this property is true only if the capacities of all the bins are the same. Let us consider for example an instance where we have six items of respective size 4, 2, 3, 3, 3, 3 and three bins of respective capacity 6, 5, 7. The sum of the sizes of the two first items is equal to the capacity of the first bin (4+2 = 6) but if we place these two items in this bin we have a solution where at least one item of size 3 is cut, while the optimal solution has no cut $(3 + 3,\ 2 + 3,\ 4 + 3)$.

Property 6. *In the case where all the bins have the same capacity C, if an item i has a size greater than C, there is at least one optimal solution in which this item i completely fills $\lfloor \frac{s_i}{C} \rfloor$ bins. The reduced problem contains N or $N-1$ (if the previous fraction is integer) items and $(M - \lfloor \frac{s_i}{C} \rfloor)$ bins. By repeating this operation on all the items whose size is greater than C, we obtain an equivalent problem in which the sizes of the items are all strictly lower than C.*

Proof. Choose a bin j with a fragment of item i. Apply the *fragment-exchange* operation on other fragments of item i until no more fragment can be completly

exchanged in bin j (the remaining part in bin j that is not filled with item i has a size lower than the size of any fragment of item i in other bins) . Now, in bin j let $\overline{f}_i$ be the size of this remaining part not filled with fragment of item i. Assume we have a fragment of object i in bin k of size f with $f > \overline{f}_i$. Cut the fragment in k in two sub-fragments, with one of size $\overline{f}_i$ and make an exchange of this sub-fragment with the fragments not belonging to i in j. This exchange do not change the number of cuts and we have bin j entirely filled with a fragment of i. This process can be repeated on $\lfloor \frac{s_i}{C} \rfloor - 1$ other bins. □

Property 7. *Even in the case where all the bins have the same capacity C, if the sum of the sizes of three items j, k and l is equal to this capacity C, it could happen that there is no optimal solution where these three items are both placed in the same bin.*

Proof. Let us consider the following instance:

- we have four bins of capacity 7,
- we have ten items, one of size 5, two of size 1 and seven of size 3.

The solution where the item of size 5 and the two items of size 1 are placed in the same bin gives 12 fragments because two cuts are needed for the seven items of size 3 in the three remaining bins. The optimal solution is composed of 11 fragments with only one cut: two bins with items 3+3+1 and one cut for the remaining items in the two last bins.

□

Property 8. *If all the items are of size less than C, any solution is equivalent to a solution where:*

- *each item is completly in one bin or is cut in two fragments in two consecutive bins (j and $j+1$),*
- *for any couple of consecutive bins $(j, j+1)$, there is at most one item with a fragment in j and the other one in $j+1$.*

Proof. First, from property 6, we can consider that, after reduction, all the items are of size less than C. We start from bin 1 and choose any item k with a fragment in bin 1. Then, we apply *fragment-exchange* operations until item k is completly in bin 1. We keep on choosing items with a fragment in bin 1 and apply *fragment-exchange* operations until items are completly in bin 1 or until bin 1 is full. The last item l treated by this process may be completly in bin 1. In this case, the process is repeated with bin 2. In the other case where the last object l is just partially in bin 1, we start by renumbering bins in such a way that a fragment of l is in bin 2 (remember all the bins are identical of capacity C) and apply *fragment-exchange* operations to put all the remaining fragments of item l in bin 2. At this step, bin 1 is filled with complete objects except the item l that is shared between bin 1 and bin 2. The property is thus proved for all the items in bin 1 and for the couple of bins $(1, 2)$. We have just

to repeatedly apply the same procedure on the other bins starting from bin 2 to complete the property.

□

Based on the above property, we have a friendly representation of a solution as depicted in figure 3.

3 Approximation algorithms

3.1 Even the simplest algorithm provides an approximation guarantee

We present here a first algorithm, probably the simplest. It does not optimize the placement of the items in the bin but we will show that it leads to a first approximation ratio. This algorithm works with any instance with the only restriction that the number of items is greater than the number of bins. This is the case when the sizes of all the items are smaller than the capacity of bins.

The algorithm, denoted by $\mathcal{A}_0$ is the following. Fill the bins with the items chosen in an arbitrary order. If an item is larger than the remaining space in the current bin, cut the item in two fragments. Complete the current bin with the first fragment and start a new bin with the second one.

In the worst case, this algorithm creates $M-1$ cuts (with M the number of bins) even if a solution with no cut exists. We illustrate the algorithm in Figure 3.

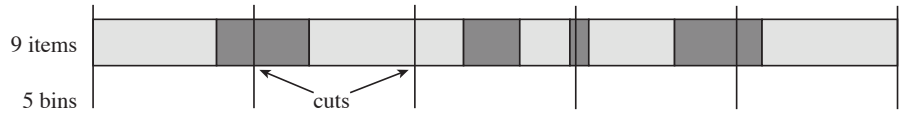

Figure 3: Illustration of the simplest algorithm with 9 items and 5 bins. It places the items in an arbitrary order and cuts them if needed. In that case, it creates 4 cuts.

Property 9. *There exists a* $2-$*approximation algorithm for* Min-FIBP *restricted to instances with more items than bins.*

Proof. Consider an instance of the problem with N items and M bins. In the worst case, the simple algorithm described above produces a solution of $N+M-1$ fragments whereas the optimal solution has no cut, this means N fragments. The approximation ratio is

$$\rho_0 = \frac{N+M-1}{N}$$

Since the number of items is larger than the number of bins, $N \geq M$,

$$\rho_0 \leq \frac{2N-1}{N} < 2$$

□

This means that the problem is approximable and that any algorithm with an approximation ratio greater than 2 is useless. As shown on Figure 4, this is a tight bound.

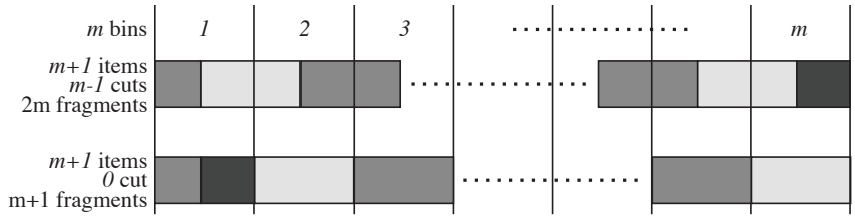

Figure 4: Two solutions for the same instance. By placing the items arbitrary, the simple algorithm could produce the first solution with M-1 cuts whereas the best solution has 0 cut. In such a case, the approximation ratio could be as close to 2 as desired.

3.2 Perfect fitting objects

In the example of Figure 4 the previous algorithm is particularly not efficient because we know from property 4 that if an object has exactly the size of a bin, it should be put alone in this bin and this simple property is not used. We build a second algorithm taken into account this property. First place all the perfect fitting objects in bins then use algorithm $\mathcal{A}_0$ to place the remaining objects. This algorithm is denoted by $\mathcal{A}_1$. $\mathcal{A}_1$ is at least as good as $\mathcal{A}_0$. To prove that it is better we need first to introduce a new notation.

Definition 1. *In a given solution, a* **block** *is a subset of items (say k items) that perfectly fills a given number of bins (say l bins). The sum of the sizes of these k items is so exactly equal to $l \times C$. At least, the first and the last item are not fragmented since they start and finish the block. Moreover, a block can not be divided into sub-blocks: it does not exist a subset of the k items that exactly fills l' bins, with $l' < l$. The number of items k contained in a block is called the* **size of a block**. *The figure 5 illustrates the notion of block.*

Property 10. *The number of fragments of such a block is: $k + l - 1$*

Proof. As there is no possibility of sub-blocks in the block, there are k items and $l-1$ cuts for a total number of $k + l - 1$ fragments. □

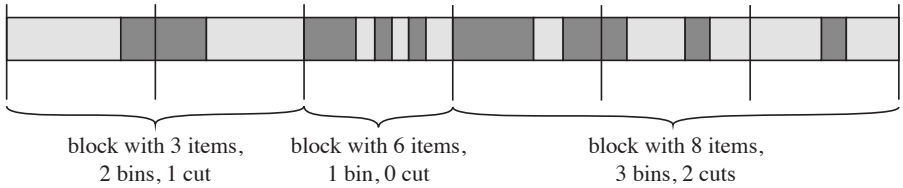

Figure 5: An exemple of solution composed of three blocks.

Property 11. *It is equivalent to minimize the number of fragments and to maximize the number of blocks that compose the solution.*

Proof. A solution composed of a total of B blocks is a solution with $N+M-B$ fragments: one cut less for each block, see property 10. Maximizing B is thus equivalent to minimize the number of fragments. □

Property 12. $\mathcal{A}_1$ *is a* $\frac{3}{2}-$*approximation algorithm for* MIN-FIBP *restricted to instances with identical bins and more items than bins.*

Proof. Consider an instance of the problem with N items and M bins. We consider an optimal solution transformed in such a way that all the blocks of size 1 are in the optimal solution (thanks to property 4). We note b_1^* the number of these blocks of size 1 and $b_{>2}^*$ the number of blocks of size 2 or more in the optimal solution. The value S^* of this solution is

$$S^* = N + M - b_1^* - b_{>2}^* \Leftrightarrow N + M = S^* + b_1^* + b_{>2}^*$$

The solution of $\mathcal{A}_1$ is composed of b_1 blocks of size 1 and b_2 blocks of size 2 or more. In the solution produced by $\mathcal{A}_1$, the number of blocks of size 1 is automatically maximized in the first step of the algorithm. So, $b_1 = b_1^*$. We can express the value S of the solution of $\mathcal{A}_1$ as follows:

$$S = N + M - b_1 - b_{>2} \Leftrightarrow S = S^* + b_{>2}^* - b_{>2}$$

$$S \leq S^* + b_{>2}^* \tag{1}$$

In the optimal solution, there are exactly b_1^* items contained in the blocks of size 1 and at least $2b_{>2}^*$ items in all the blocks of size 2 or more. Then we can bound the number N of items by a function depending on the number of blocks.

$$b_1^* + 2b_{>2}^* \leq N$$

$$b_{>2}^* \leq \frac{N - b_1^*}{2} \tag{2}$$

By combining equations 1 and 2 we obtain

$$S \leq S^* + \frac{N - b_1^*}{2} \leq S^* + \frac{N}{2} \tag{3}$$

The approximation ratio is

$$\rho_1 = \frac{S}{S^*} \leq 1 + \frac{N}{2S^*}$$

As $S^* \geq N$ we deduce that $\mathcal{A}_1$ is an approximation algorithm of ratio $\rho_1 \leq \frac{3}{2}$. □

Equation 3 gives an indication of instances for which the ratio is reached: when there is no perfect fitting object (Figure 6). In such instances, $\mathcal{A}_1$ is not able to optimize the placement of items but these instances contain enough items, at least two per bins, to limit the weight of cuts.

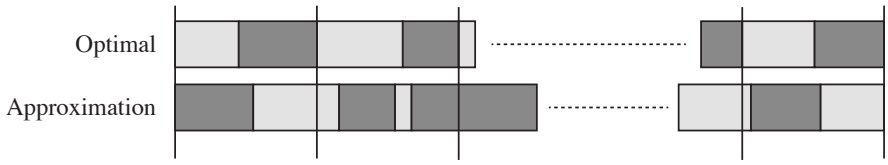

Figure 6: An exemple of instance without exact fitting object. In that case, the weight of the optimal solution is $2M$ and the weight of the solution produced by $\mathcal{A}_1$ could be, in the worst case, $3M - 1$.

3.3 Perfect fitting couples of objects

Previously, we adapt $\mathcal{A}_0$ to take into account the perfect fitting objects. We can do the same thing for the perfect fitting couples of objects. We build a new algorithm $\mathcal{A}_2$ which works as follows. First, place all the perfect fitting objects in bins, next place as many perfect fitting couples of objects as possible and then use the simple algorithm $\mathcal{A}_0$ to place the remaining objects.

Finding all perfect fitting objects and couples is equivalent to the maximum set packing (with sets of size 1 and 2). This can be done in polynomial time using matching techniques [6].

Property 13. *$\mathcal{A}_2$ is a $\frac{4}{3}-$approximation algorithm for* Min-FIBP *restricted to instances with identical bins and more items than bins.*

Proof. Consider an instance of the problem with N items, M bins and an optimal solution composed of b_1^* blocks of size 1, b_2^* blocks of size exactly 2 and $b_{>3}^*$ blocks of size 3 or more. We consider also that properties 4 and 5 have been applied on the optimal solution and that b_1^* and b_2^* are maximal. The value S^* of this solution is

$$S^* = N + M - b_1^* - b_2^* - b_{>3}^* \Leftrightarrow N + M = S^* + b_1^* + b_2^* + b_{>3}^*$$

Similarly, the solution of $\mathcal{A}_2$ is

$$S = N + M - b_1 - b_2 - b_{>3} \Leftrightarrow S = S^* + b_1^* + b_2^* + b_{>3}^* - b_1 - b_2 - b_{>3}$$

In the solution produced by $\mathcal{A}_2$, the number of blocks of size 1 is maximized as well as the number of blocks of size 2. This means that $b_1 = b_1^*$ and $b_2 = b_2^*$.

$$S = S^* + b_{>3}^* - b_{>3} \leq S^* + b_{>3}^* \tag{4}$$

We can bound the number N of objects by a function depending on the number of blocks.

$$b_1^* + 2b_2^* + 3b_{>3}^* \leq N$$

$$b_{>3}^* \leq \frac{N - b_1^* - 2b_2^*}{3} \tag{5}$$

By combining equations 4 and 5 we obtain

$$S \leq S^* + \frac{N - b_1^* - 2b_2^*}{3} \leq S^* + \frac{N}{3} \tag{6}$$

The approximation ratio is

$$\rho_2 = \frac{S}{S^*} \leq 1 + \frac{N}{3S^*}$$

As $S^* \geq N$ we deduce that $\mathcal{A}_2$ is an approximation algorithm of ratio $\rho_2 \leq \frac{4}{3}$. □

The approximation ratio is a tight bound. Similary to the example proposed for the $\mathcal{A}_1$ (figure 6), we can provide instances without perfect fitting object and couple in which the approximation ratio is reached.

3.4 $\frac{6}{5}$-Approximation Algorithm

We have presented three algorithms, the first one, $\mathcal{A}_0$, does not optimize the block number. The second, $\mathcal{A}_1$, maximizes the number of blocks of size 1 and the third, $\mathcal{A}_2$, maximizes the number of blocks whose sizes are 1 and 2. We present now a generalization. For a fixed k, $\mathcal{A}_k$ maximizes the number of blocks whose sizes are less or equal than k. The algorithm is composed of four steps:

1. place the maximum number of blocks of size 1;
2. place the maximum number of blocks of size 2;
3. place the maximum number of blocks of size 3 to k;
4. place the remaining items using $\mathcal{A}_0$.

With the properties 4 and 5, we know that there exists at least one optimal solution S^* in which the numbers of blocks of size 1 and 2, are the same as in the solution S provided by $\mathcal{A}_k$.

Solving the third step of our algorithm is equivalent to solving the maximum set packing problem with set of size less or equal than $k \geq 3$. This cannot be done in polynomial time but it can be approximated within $2/k$ [9]. This means

that in S the total number of blocks of size 3 to k is at least $2/k$ times the maximum number of blocks of size 3 to k that can be obtained by any solution, and in particular by S^*. As we will see, the approximation ratio of our algorithm is then linked with the approximation ratio of maximum set packing.

Property 14. *$\mathcal{A}_4$ is a $\frac{6}{5}-$approximation algorithm for* MIN-FIBP *restricted to instances with identical bins and more items than bins.*

Proof. Consider an instance of the problem with N items, M bins and the optimal solution S^* maximizing the number of blocks of size 1 and 2. It is composed of $b_1^*, b_2^*, \ldots, b_k^*$ blocks whose size are $1, 2, \ldots, k$ and $b_{>k}^*$ block of size $k+1$ and more. The weight S^* of this solution is

$$S^* = N + M - \sum_{i=1}^{k} b_i^* - b_{>k}^* \Leftrightarrow N + M = S^* + \sum_{i=1}^{k} b_i^* + b_{>k}^*$$

Similarly, the solution of $\mathcal{A}_k$ is

$$S = N + M - \sum_{i=1}^{k} b_i - b_{>k} \Leftrightarrow S = S^* + \sum_{i=1}^{k} b_i^* + b_{>k}^* - \sum_{i=1}^{k} b_i - b_{>k}$$

In the solution produced by $\mathcal{A}_k$, the number of blocks of sizes 1 and 2 are maximized : $b_1 = b_1^*$ and $b_2 = b_2^*$. We use an approximation algorithm to find as many blocks of size 3 to k as possible. The $\frac{2}{k}$-approximation provided by [9] allows us to find $\sum_{i=3}^{k} b_i \geq \frac{2}{k} \sum_{i=3}^{k} b_i^*$ blocks.

$$S \leq S^* + \sum_{i=3}^{k} b_i^* + b_{>k}^* - \frac{2}{k} \sum_{i=3}^{k} b_i^* - b_{>k} \leq S^* + \frac{k-2}{k} \sum_{i=3}^{k} b_i^* + b_{>k}^* \qquad (7)$$

We can bound the number N of objects by a function depending on the number of blocks.

$$\sum_{i=1}^{k} i b_i^* + (k+1) b_{>k}^* \leq N$$

$$b_{>k}^* \leq \frac{N - \sum_{i=1}^{k} i b_i^*}{k+1} \qquad (8)$$

By combining equations 7 and 8 we obtain

$$S \leq S^* + \frac{k-2}{k} \sum_{i=3}^{k} b_i^* + \frac{N - \sum_{i=1}^{k} i b_i^*}{k+1}$$

$$S \leq S^* + \sum_{i=3}^{k} \left(\frac{k-2}{k} - \frac{i}{k+1}\right) b_i^* + \frac{N}{k+1} - \frac{b_1^*}{k+1} - \frac{2b_2^*}{k+1}$$

$$S \leq S^* + \sum_{i=3}^{k} (\frac{k-2}{k} - \frac{i}{k+1}) b_i^* + \frac{N}{k+1}$$

$$\frac{S}{S^*} \leq 1 + \frac{1}{S^*} \sum_{i=3}^{k} \left(\frac{k-2}{k} - \frac{i}{k+1} \right) b_i^* + \frac{N}{S^*(k+1)}$$

As $S^* \geq N$

$$\frac{S}{S^*} \leq \frac{k+2}{k+1} + \frac{1}{S^*} \sum_{i=3}^{k} \left(\frac{k-2}{k} - \frac{i}{k+1} \right) b_i^* \tag{9}$$

The approximation ratio obtained in equation 9 is bounded by $\frac{k+2}{k+1}$ as long as all the terms with b_i^* are negative. This is the case when $k = 3$: with $i = 3$, $\frac{k-2}{k} - \frac{i}{k+1} = \frac{-5}{12}$. This is still the case when $k = 4$: $\frac{k-2}{k} - \frac{i}{k+1} = \frac{-1}{10}$ and $\frac{-3}{10}$ for $i = 3$ and 4. But it does not work anymore when $k = 5$ and $i = 3$: $\frac{k-2}{k} - \frac{i}{k+1} = \frac{1}{10} > 0$. The best approximation ratio is so reached and, in this case, $\mathcal{A}_4$ is a $\frac{6}{5}$-approximation algorithm. □

As for the previous algorithms, this is a thight bound which can be asymptotically reached for large instances without block of size strictly less than 5.

4 Conclusion

In this paper a new variant of bin packing problem has been introduced. The items are allowed to be splitted into several fragments so that all the items can fill a given number of bins but the total number of fragments has to be minimized. This problem is specially usefull to model splittable resource allocation. We have first defined this problem formally and presented properties on optimal solutions. We have then presented a family of approximation algorithms for instances of the problem in which the bins are all of the same size and each item is smaller than a bin. The best one provides a $\frac{6}{5}$ approximation ratio.

Many questions remain open but we think that the following ones are the most important. First is it possible to provide a better approximation ratio than $\frac{6}{5}$. Then, how to generalize the approximation algorithm to the general case where the bins can have different capacities. Finally, which optimization methods - heuristics and exact methods - can be proposed to efficiently solve this problem.

References

[1] E. Coffman Jr, M. Garey, D. Johnson, Approximation algorithms for bin packing: A survey, in: Approximation algorithms for NP-hard problems, PWS Publishing Co., p 46-93, 1996.

[2] P. Mandal, P.P Chakrabarti and S. Ghose, Complexity of fragmentable object bin packing and an application, Computers & Mathematics with Applications 35 (11), p 91-97, 1998.

[3] D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey and R.L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, *S*IAM Journal of Computing, 3, p 299-325, 1974

[4] D. Poulain, J. Tomasik, M.A. Weisser and D. Barth, Minimization of the receiver cost in an all-optical ring with a limited number of wavelengths, Proceedings of 27th International Symposium on Computer and Information Sciences (ISCIS 2012), p 239-247, 2012.

[5] D. Chiaroni, D. Neilson, C. Simonneau and J.C. Antona, Novel optical packet switching nodes for metro and backbone networks, ONDM 2010, 14th conference on Optical Network Design and Modeling, 2010.

[6] M. Garey, D. Johnson, Computers and intractability, Vol. 174, Freeman San Francisco, CA, 1979.

[7] A. Lodi, S. Martello, M. Monaci, C. Cicconetti, L. Lenzini, E. Mingozzi, C. Eklund, J. Moilanen. Efficient two-dimensional packing algorithms for mobile WiMAX, *M*anagement Science 57, p 2130-2144, 2011.

[8] H. Dyckhoff, A typology of cutting and packing problems, *E*uropean Journal of Operational Research 44, p 145-160, 1990.

[9] V. Kann, Maximum bounded 3-dimensional matching is max snp-complete, *I*nformation Processing Letters 37 (1), p 27-35, 1991.

[10] R. Yesodah and T. Amudha, A comparative study on heuristic procedures to solve Bin Packing problems, *I*nternational Journal in Foundations of Computer Science and Technology, Vol.2, No 6, p 37-49, 2012.

[11] N. Karmarkar and R.M. Karp, An efficient approximation scheme for the one-dimensional bin packing problem, in proceedings of the 23rd Annual Symp. on Foundations of Computer Science, p 312-320, 1982.

[12] N. Menakerman and R. Rom, Bin packing with item fragmentation, in proceedings of the 7th international Workshop on Algorithms and Data Structures (WADS'01), p 313-324, 2001