– PANTHÉON - SORBONNE –
UNIVERSITÉ PARIS 1

Centre d'Economie de la Sorbonne
UMR 8174

Cahiers de la MSE

On-line bin-packing problem : maximizing the number of unused bins

Bernard KOUAKOU

Marc DEMANGE

Eric SOUTIF

2006.38

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Maison des Sciences Économiques, 106-112 boulevard de L'Hôpital, 75647 Paris Cedex 13
http://mse.univ-paris1.fr/Publicat.htm
ISSN : 1624-0340

On-line bin-packing problem: maximizing the number of unused bins

Bernard Kouakou[1], Marc Demange[2], and Eric Soutif[3]

[1] CERMSEM, Université Paris 1, 106–112 bd de l'Hôpital, F-75013 Paris, France, kouakou@univ-paris1.fr
[2] ESSEC, SID department, Avenue Bernard HIRSH, BP 105, F-95021 Cergy Pontoise Cedex, France, demange@essec.fr,
[3] CEDRIC, CNAM, 292 rue Saint-Martin, F-75003 Paris, France, soutif@cnam.fr

Abstract. In this paper, we study the on-line version of the bin-packing problem. We analyze the approximation behavior of an on-line bin-packing algorithm under an approximation criterion called *differential ratio*. We are interested in two types of results: the differential competitivity ratio guaranteed by the on-line algorithm and hardness results that account for the difficulty of the problem and for the quality of the algorithm developed to solve it. In its off-line version, the bin-packing problem, BP, is better approximated in differential framework than in standard one. our objective is to determine if or not such result exists for the on-line version of BP.

keywords: on-line algorithm, bin-packing problem, competitivity ratio.

1 Introduction

In the classical bin-packing problem, we are given a list $L = \{x_1, x_2, \cdots, x_n\}$, each item $x_i \in]0,1]$, and we want to find a packing of these items into a minimum number of unit-capacity bins. In this paper, we study the on-line bin-packing problem denoted by LBP. It is defined [11] by the quadruplet (BP, R, R', Sol_{LBP}) where R denotes the set of informations known at the beginning of the game. R also describes how the final instance is revealed. R' is a set of rules describing how the on-line algorithm constructs the solution and Sol_{LBP} denotes the set of feasible solutions. Modifying rules R and R' means to consider different versions of the on-line bin-packing problem:
We first deal with the version of LBP where items of the final instance are revealed one by one and the algorithm has to irrevocably pack items as soon as they are revealed. For this version, Johnson and al. [10] proved that the classical Next Fit algorithm (NF) and the Worst Fit one, WF, achieve asymptotic ratio $R^{\infty}_{NF} = R^{\infty}_{WF} = 2$. As for the First Fit ($FF$) and Best Fit ($BF$) algorithms, Johnson and al. proved that $R^{\infty}_{FF} = R^{\infty}_{BF} = \frac{17}{10}$. On the other hand, Liang [8] showed that no on-line algorithm can guarantee an asymptotic ratio $R < 1,53634577....$ and Van Vliet [7] improved this bound to 1,540. If for

each item, the choice of where to pack it is restricted to a set of k opens bins, bounded-space algorithms are used to solve LBP. The harmonic+1 algorithm of Richey [9] has the current best known asymptotic ratio. Richey proved that $1,5874 \leq R^{\infty}_{harmonic+1} \leq 1,587936$.
We also deal with another version of the on-line bin-packing problem for which the rules R consist in revealing the final instance in two steps (2 clusters) L_1 and L_2. It can be seen as a boundary between off-line and on-line combinatorial optimization. It is proved in [3] that if A is an off-line algorithm achieving the approximation ratio $\rho(L)$ for the the bin-packing then, there exists an algorithm for the 2-steps problem achieving the competitivity ratio $C_{LA} \geq \frac{2}{3}[\rho(L) - \frac{1}{\beta(L)}]$, where $\beta(L)$ denotes the value of the optimal solution of L. Finally, we study the on-line bin-packing problem with uniform bin sizes and LIB constraint (lower item at the bottom). In that version, each bin is of unit-capacity and the algorithm should not pack a longer item upper a smaller one. Manyem [14] provided an algorithm based on the First Fit principle guaranteeing an asymptotic approximation ratio of 3. Below, we recall the definition of the asymptotic ratio used by Manyem. Let $AL(G)$ denote the number of bins returned by an online algorithm AL for an instance L and let $OPT(L)$ be the optimal value of bin sizes necessary to packing items of L. The asymptotic approximation ratio, AAR, is defined by

$$R_{AL} = \lim_{s\to\infty} \sup_{L}\{\frac{AL(G)}{OPT(L)}, OPT > s\}$$

For the same problem, using adversary arguments, Finlay and Manyem [15] proved that no algorithm can guarantee an AAR less than 1.76. Again for the same problem, in [13], Manyem et al. construct counter-examples to show that the guaranteed AAR's of the well-known First Fit (FF), Best Fit (BF) and Harmonic Fit (HF) algorithms are at least two.
All these results are displayed in the following table.

Problems	**competitivity ratio** (upper bound)	**Hardness results** (lower bound)	General hardness results
LBP	Johnson et al. $R^{\infty}_{NF} = R^{\infty}_{WF} = 2$	2	Van Vliet 1,540
	Johnson et al. $R^{\infty}_{FF} = R^{\infty}_{BF} = \frac{17}{10}$	$\frac{17}{10}$	
	Richey $R^{\infty}_{H} \leq 1,587936$	$R^{\infty}_{H} \geq 1,5874$	
LIB-LBP	Manyem $R^{\infty}_{FF} = 3$	$R^{\infty}_{A} \geq 2$ ($A = FF, BF, HF$)	Finlay and Manyem 1.76

In this paper, we consider both classical and LIB on-line bin-packing problems with uniform bin sizes and analyze the competitivity behavior of algorithms under an approximation criterion, called *differential competitivity ratio*. It is the on-line version of the differential approximation ratio defined in [1].

Definition 1 (Differential competitivity ratio). *Let L be an instance of an optimization problem Π, and A an on-line algorithm supposed to feasibly solve π. We, respectively, denote by $w(L)$, $\lambda_A(L)$, and $\beta(L)$, the values of the worst solution, the approximated one (provided by A) and the optimal one. Let $\delta_A(L)$ be equal to $[w(L) - \lambda(L)]/[w(L) - \beta(L)]$; then the quantity $\delta_A = sup\{r : \delta_A(L) \geq r, L \text{ instance of } \Pi\}$ is the differential competitivity ratio of A for π.*

For the case of bin-packing, the worst-case solution consists in taking a bin per item, i.e., $w(L) = |L| = n$, where n denotes the order of the instance L.
Key-requirement of the differential approximation framework is the stability of any adopted approximation ratio with respect to affine transformation of the objective function.

2 The $n - steps$ on-line bin-packing problem

In this section, we consider the on-line bin-packing problem where, for every instance L, its n items are revealed one by one (such instance is called an *n-steps instance*). In [1], it is proved that both First Fit (FF) and Best Fit (BF) algorithms guarantee a differential competitivity ratio $\delta \geq 1/2$; moreover, bound 1/2 is tight. The proof is based on the notion of *n-worst instance*: given an **NP**-hard problem Π and an on-line algorithm LA for Π, an n-worst instance is an instance for which LA performs the worst possible with respect to the differential approximation ratio. The following lemma is proved in [1].

Lemma 1. *Let A be FF or BF, let $n > 0$ and L be an n-worst instance for (BP, A). Let $b_1, \cdots, b_{\lambda_A(L)}$ $(b_i \neq \emptyset)$ be the solution provided by A. If there exists a set of bins $I \subset \{1, \cdots, \lambda_A(L)\}$ such that the list $L^{'} = L \setminus \bigcup_{i\in I} b_i$ satisfies $\beta(L^{'}) \leq \beta(L) - x$, with $0 \leq x \leq y \leq z, x \neq z$, where $y = |I|$ and $z = \left|\bigcup_{i\in I} b_i\right|$, then $\delta_A(L) \geq \frac{z-y}{z-x}$.*

Here, we prove that not only FF and BF but also every algorithm solving the on-line version of the bin-packing problem, LBP, (items are revealed one by one) cannot guarantee a differential competitivity ratio $\delta > \frac{1}{2}$. Let us note that if FF ranges items in decreasing order then, it is called FFD, First Fit Decreasing.

Theorem 1. *Let A be an on-line algorithm solving LBP. A cannot guarantee a competitivity ratio $\delta > 1/2$ even in the particular case where the on-line list is revealed in increasing order. However, if items of the on-line instance are revealed in decreasing order then, algorithm First Fit Decreasing guarantees a differential competitivity ratio $\delta \geq 3/4$ and this bound is tight.*

Proof. Let us point out the following remark.

Remark 1. Let A be an algorithm for the on-line bin-packing problem guaranteeing a differential competitivity ratio $\delta > 1/2$ and let L be an n-steps instance for LBP with an optimal value $\beta = n/2$. Moreover, let us assume that there does not exist a bin with more than two items in every feasible solution (even in the optimal one) of L. If A guarantees a ratio $\delta > 1/2$ then, in the solution returned by A, the number λ_2 of 2-bins (bins containing exactly 2 items) satisfies $\lambda_2 > n/4$. Indeed, if we denote by λ_1, λ_2, λ respectively, the number of 1-bins , the number of 2-bins and the total number of bins returned by A, we have $\lambda = \lambda_1 + \lambda_2$ and $w = n = \lambda_1 + 2\lambda_2$. So, the ratio $\delta(L) = \frac{w-\lambda}{w-\beta}$ becomes $\delta(L) = \frac{2\lambda_2}{n}$, since $\beta = n/2$. Therefore, $\delta > 1/2$ is equivalent to $\frac{2\lambda_2}{n} > 1/2$ i.e. $\lambda_2 > \frac{n}{4}$, which justifies the remark.

Consequently, if A is an algorithm for the (n-steps) on-line bin-packing problem guaranteeing a differential competitivity ratio $\delta > 1/2$ then, A returns for the 4-steps instance $L_1 = \{\frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon\}$, two 2-bins: $b_1 = \{\frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon\}$ and $b_2 = \{\frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon\}$. After these observations, let us consider an on-line algorithm A guaranteeing for LBP a differential competitivity $\delta > 1/2$ and let us apply A to the following $8-$*steps* instance $L = \{\frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon, \frac{1}{2}-\epsilon, \frac{1}{2}+\epsilon, \frac{1}{2}+\epsilon, \frac{1}{2}+\epsilon, \frac{1}{2}+\epsilon\}$. We note that A does not know the size of the final instance. So, it does not know if the adversary will reveal other items or not after the four first ones. In order to guarantee a ratio $\delta > 1/2$, if no item arrives later (after the four first ones), A must form two 2-bins with the four first items revealed. In this case, if the adversary reveals four items equal to $1/2 + \epsilon$ then, four 1-bins are necessarily formed. Finally, A needs six bins to pack items of L: two 2-bins (b_1, b_2) and four 1-bins. For this instance L, we have $\lambda_2 = 2 = n/4$, which contradicts remark 1 since for this list, we might have $\lambda_2 > \frac{n}{4} = 2$. Consequently, A cannot guarantee a competitivity ratio $\delta > 1/2$.

3 The 2-steps on-line bin-packing problem

Here, we assume that the final on-line instance is revealed in two steps (such instance is called a *2-steps instance*). One can remark that the First Fit algorithm guaranteeing the differential competitivity ratio $\delta \geq 1/2$ for the $n-steps$ version, guarantees at least the same ratio for the $2-clusters$ version of LBP. Here, we improve this ratio by providing the algorithm called DLA, defined below:

Phases of the on-line differential algorithm, DLA.

1. DLA first studies the case where the three smallest items x_1, x_2, x_3 of the first cluster, L_1, can be packed in a same bin, i.e., $x_1 + x_2 + x_3 \leq 1$. In this case, it uses algorithm FF to solve list L_1 after ordering its items in increasing order. Then, Algorithm FF is also used to solve the second cluster, L_2 (it does not matter if L_2 is ordered in an increasing order or not).
2. The second phase of the algorithm deals with the case where no bin from L_1 can contain at least three items (even in an optimal solution of L_1). We give in this case, the following strategy, called LA.

Strategy LA

1. $i \leftarrow 1$: order the first cluster, L_1, in increasing order: $x_1 \leq x_2 \cdots \leq x_n$;
2. solve optimally $L_1 \setminus \{x_1, x_2, x_3\}$ (see OPT, the last phase of algorithm FFI in [1]);
4. $b_2^+ \leftarrow card(B_2^+)$ % number of 2-bins containing an item greater than 1/2.%
5. $\beta_2^- \leftarrow card(B_2^-)$ % number of 2-bins not containing any item greater than 1/2.%
6. $b_1^+ \leftarrow card(B_1^+)$ % number of 1-bin containing an item greater than 1/2.%
7. $\beta_1^- \leftarrow \left|B_1^-\right|$ % number of 1-bin not containing any item greater than 1/2.%
8. Order items of $B_2^- \cup B_1^-$ in decreasing order and arrange them (in this order) by putting two items per bin;
9. $x \leftarrow$ the smallest number greater or equal to $\frac{1}{3}(\beta_2^- - \beta_1^-)$.
10. $b_2^- \leftarrow (\beta_2^- - x)$
11. If $b_2^+ + b_2^- > M$ then transform x 2-bins containing the smallest items of B_2^- in $2x$ 1-bins: $b_1^- \leftarrow \beta_1^- + 2x$ % we have broken up x *2-bins* to form $2x$ *1-bins* %
12. Form two new bins: $\{x_1, x_2\}$ and $\{x_3\}$;
13. $i \leftarrow 2$: order L_2 in an increasing order; $y_1 \leq\ y_2 \cdots \leq\ y_p \leq z_1 \leq \cdots \leq z_q$
14. If $x_1 + x_2 + y_1 \leq 1$, provide the 3-bin, $\{x_1, x_2, y_1\}$, and process $L_2 \setminus \{y_1\}$ with FF
15. else
16. If $x_3 + y_1 + y_2 \leq 1$, provide the 3-bin, $\{x_3, y_1, y_2\}$, and process $L_2 \setminus \{y_1, y_2\}$ with FF
17. Else
18. IF $y_1 + y_2 + y_3 \leq 1$, form the 3-bin, $\{y_1, y_2, y_3\}$, and process $L_2 \setminus \{y_1, y_2, y_3\}$ with FF
19. Else
20. Select items of type z_i in a decreasing order to complete bins of $B_1^- \cup \{x_3\}$.
21. Order bins of B_1^+ in decreasing order, and complete them with items of type y_i. These items of type y_i must be selected in decreasing order;
22. Form the maximum number of bins of type $\{y_i, z_j\}$ with the left items of type y_i and z_j.
23. Complete (if possible) bins of type $B_1^- \cup \{x_3\}$ with items of type y_i.
24. Finally, form bins of type $\{z_i\}$, $\{y_i, y_j\}$ and possibly a bin of type $\{y_i\}$.
25. Return $LA(L)$ %the on-line solution%

The following lemma holds.

Lemma 2. *Let L be a 2-steps instance for the on-line bin-packing problem. Denote by β_3 and β_3^+ the number of 3-bins (resp., the number of bins containing 3 or more than 3 items) in an optimal solution. If at the end of the on-line process the algorithm does not return a bin containing at least 3 items then, $\beta_3 = \beta_3^+ \leq 1$.*

Proof. If at the end of the on-line process the algorithm does not return a bin containing at least 3 items then, every feasible (or optimal) solution of $L_1 \cup L_2$ cannot contain a bin with 3 items in L_1 or 3 items in L_2. We cannot also have (in an optimal solution) 3-bins of type $\{x_i, x_j, z_k\}$ since the sum of the smallest items in L_1 satisfies $x_1 + x_2 + x_3 > 1$ and $z_k > 1/2 \geq x_3$, for all $k = 1 \cdots q$. Moreover, we cannot return 3-bins of type $\{x_i, y_j, z_k\}$ in an optimal solution. Indeed, for all x_i, y_j, z_k, $x_i + y_j + z_k > x_1 + y_j + x_2$, since $x_i \geq x_1$ and $z_k > \frac{1}{2} \geq x_2$. Let us recall that $\forall j = 1 \cdots p, x_1 + y_j + x_2 > 1$ (if not, our algorithm would return a

3-bin). So, we have for all x_i, y_j, z_k, $x_i + y_j + z_k > 1$.
Let us also note that an optimal solution cannot range x_3, y_1 and y_2 in a same bin. In the opposite case the algorithm could form the 3-bin $b = \{x_3, y_1, y_2\}$. Only 3-bins of type $\{x_1, y_j, y_k\}$ and $\{x_2, y_n, y_m\}$ can be returned by an optimal algorithm. Let us then assume that the optimal algorithm returns two 3-bins, $\{x_1, y_j, y_k\}$ and $\{x_2, y_n, y_m\}$. In this case, we have $x_1 + y_j + y_k \le 1$ and $x_2 + y_n + y_m \le 1$. Then, summing both inequalities, we have:

$$x_1 + y_j + y_k + x_2 + y_n + y_m \le 2. \quad (1)$$

Since $x_1 + x_2 + y_1 > 1$, and $\forall j = 1 \dots p$, $y_j \ge y_1$, we have $x_1 + x_2 + y_j > 1$, for all $j = 1, \cdots, p$. Then, we deduce from inequality (1) that $y_k + y_n + y_m < 1$. This is a contradiction since our algorithm cannot return a 3-bin from L_2, which implies $\beta_3 \le 1$. So the optimal algorithm returns at most one 3-bin. It either of type $\{x_1, y_j, y_k\}$ or of type $\{x_2, y_n, y_m\}$. We can prove with a similar argument that an optimal algorithm cannot return a bin with more than 3 items, so $\beta_3 = \beta_3^+ \le 1$.

The following theorem holds.

Theorem 2. $\delta_{DLA} \ge \frac{2}{3}(1 - \frac{14}{3(b_2^+ + b_2^-)+11})$.

Proof. Let us recall [1] that if for an instance L, Algorithm FF packs three (3) items in a same bin then, it guarantees the differential competitivity ratio of $\frac{2}{3}$. So according to the structure of our algorithm, it suffices to prove that the strategy LA (second phase of Algorithm DLA) guarantees the differential competitivity ratio $\delta \ge \frac{2}{3}[1 - \frac{14}{3(b_2^+ + b_2^-)+11}]$.
We recall that:
B_2^+ denotes the set of 2-bins containing an item greater than 1/2.
(B_2^-) is the set of 2-bins not containing any item greater than 1/2.
(B_1^+) denotes the set of 1-bin containing an item greater than 1/2.
B_1^- is the set of 1-bin not containing any item greater than 1/2.
Now, let us denote by Z_2^+ (respectively Z_2^-) the number of 2-bins of type $\{z_i, x_i\}$ in the optimal solution, with $x_i \in B_2^+, |B_2^+| = b_2^+$, (resp. with $x_i \in B_2^- \cup \{x_1, x_2\}$, $|B_2^-| = b_2^-$.) We assume that $Z_2^- \ge 2$.
Z_1^- is the number of 2-bins of type $\{z_i, x_i\}$ in the optimal solution, with $x_i \in B_1^- \cup \{x_3\}$, $|B_1^-| = b_1^-$.
ZY is the number of 2-bins of type $\{z_i, y_i\}$ in the optimal solution.
K_1 and K are respectively the number of 2-bins of type $\{x_i, x_j\}$ in the optimal solution, with $x_i \in B_1^+$, $x_j \in \{x_1, x_2, x_3\}$ (resp. with x_i *or* $x_j \in B_2^+$ and $x_i > 1/2$).
Y_1 and Y_2 denote respectively the number of 2-bins of type $\{x_i, y_j\}$ in the optimal solution, with $x_i \in B_1^+, |B_1^+| = b_1^+$. (resp. with $x_i \in B_2^+$ and $x_i > 1/2$).
If we denote by β the optimal number of used bins in the final instance L, we deduce from lemma 2 that $\beta = \beta_1 + \beta_2 + \beta_3$, where β_i is the number of i-bins in the optimal solution. Let us consider an optimal solution which maximizes ZY.

One can remark that

$$\beta_2 \leq Z_2^+ + Z_2^- + Z_1^- + ZY + Y_2 + Y_1 + K + K_1 + \frac{1}{2}(p - ZY - Y_1 - Y_2) + \frac{1}{2}(b_2^+ + 2b_2^- + 2 + b_1^- + 1 - Z_2^+ - Z_2^- - Z_1^- - K - K_1 - \beta_3)$$

As $w = \beta_1 + 2\beta_2 + 3\beta_3$, we have $w - \beta = \beta_2 + 2\beta_3$ then,

$$w-\beta \leq \frac{1}{2}(Z_2^+ + Z_2^- + Z_1^- + ZY + Y_2 + Y_1 + K + p + b_2^+ + b_1^- + K_1) + b_2^- + \frac{3}{2}\beta_3 + \frac{3}{2}.$$

One can again remark that $K + Z_2^+ + Y_2 \leq 2b_2^+$ and $K_1 \leq 3$. We have also proved that $\beta_3 \leq 1$. So,

$$w - \beta \leq \frac{3}{2}b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{9}{2} \qquad (2)$$

We now analyse the behavior of the algorithm. The following remarks will be useful in the sequel.

Remark 2. In strategy LA, bins of B_1^+ are ordered in decreasing order and, before assigning items of type y_i to others bins, they (y_i-items) are first used to complete bins of B_1^+ by also selecting these y_i-items in decreasing order. So, if we denote by Y_{1A} the number of $\{y_i, x_j\}$-bins, with $x_j \in B_1^+$, returned by the algorithm at the end of the on-line process, it satisfies $Y_{1A} \geq Y_1$. We put $Y_{1A} = Y_1 + N$, $N \geq 0$.

Remark 3. We know that the optimal solution contains:
Z_2^- bins of type $\{z_j, x_i\}$ with $x_i \in B_2^- \cup \{x_1, x_2\}$;
Z_1^- bins of type $\{z_j, x_i\}$ with $x_i \in B_1^- \cup \{x_3\}$;
Y_1 bins of type $\{y_i, x_j\}$, where $x_j \in B_1^+$;
ZY bins of type $\{z_j, y_i\}$;
and other bins of different types.
Moreover, (i) every items in $B_1^- \cup \{x_3\}$ is smaller than each item in B_2^-, (ii) algorithm A uses, at the second step, the biggest items of type z_i to complete bins in $B_1^- \cup \{x_3\}$. It also uses the biggest items of type y_i to complete bins in B_1^+.
So, if algorithm A uses less than (or exactly) $Z_1^- + Z_2^- - 2$ items of type z_i and exactly Y_1 items of type y_i to respectively complete bins in $B_1^- \cup \{x_3\}$ and bins in B_1^+, it can form at least ZY bins of type $\{z_i, y_j\}$, as in the optimal solution.
Now the question is to know what happens if the algorithm uses:

- more than $Z_1^- + Z_2^- - 2$ items of type z_i to complete bins in $B_1^- \cup \{x_3\}$
- more than Y_1 items of type y_i to complete bins of B_1^+
- more than $Z_1^- + Z_2^- - 2$ items of type z_i and more than Y_1 items of type y_i, at the same time, to complete bins in $B_1^- \cup \{x_3\}$ and bins in B_1^+ respectively.

This remark leads us to the following cases (we denote by Z_A the number of items of type z_i used by the algorithm to complete bins in $B_1^- \cup \{x_3\}$).

case 1: $b_1^- + 1 \leq Z_2^- + Z_1^- - 2$.
As the Strategy LA first complete bins in $B_1^- \cup \{x_3\}$ with z_i items before assigning others bins to the left items of type z_i, the algorithm uses, here, less than (or exactly) $Z_2^- + Z_1^- - 2$ items of type z_i in order to transform each 1-bin of $B_1^- \cup \{x_3\}$ in 2-bin, i.e $Z_A \leq Z_2^- + Z_1^- - 2$. So we focus our analysis on the number of $y_i - items$ used by the algorithm to transform some 1-bins of B_1^+ in 2-bins.
Let us denote by ZY_A the number of $\{z_i, y_j\}$-bins returned by the algorithm at the end of the on-line process. Then,

$$\lambda_2 = b_2^+ + b_2^- + 1 + b_1^- + 1 + ZY_A + Y_{1A} + \frac{1}{2}(P - ZY_A - Y_{1A} - \epsilon_1) \qquad (3)$$

with $\epsilon_1 = 0$, if $P - ZY_A - Y_{1A}$ is even, or 1 if not.
As ZY bins of type $\{z_i, y_j\}$ have been returned in the optimal solution, We distinguish the following two sub-cases.
case 1-a: $ZY_A \geq ZY$, i.e $ZY_A = ZY + M_A$, with $M_A \geq 0$.
It means that after constructing Y_{1A} bins of type $\{y_i, x_j\}$, there are enough y_i-items to form at least ZY bins of type $\{z_i, y_j\}$. Equality 3 (see above) becomes

$$\lambda_2 = b_2^+ + b_2^- + 1 + b_1^- + 1 + ZY + M_A + Y_1 + N + \frac{1}{2}(P - ZY - M_A - Y_1 - N - \epsilon_1)$$

which leads us to

$$w - \lambda = \lambda_2 \geq b_2^+ + b_2^- + b_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{3}{2}$$

case 1-b: $ZY_A < ZY$, i.e $ZY_A = ZY - M_A$, with $M_A > 0$.
Here, after constructing Y_{1A} bins of type $\{y_i, x_j\}$, there are not enough y_i-items to form, at least, ZY bins of type $\{z_i, y_j\}$. We know that $Y_{1A} = Y_1 + N$ and for the current sub-case, $ZY_A = ZY - M_A$. Then, one can remark that (for this sub-case) $N \geq M_A$. In the opposite case, revisiting Remark 3, one can see that the optimal solution cannot contain ZY bins of type $\{z_i, y_i\}$. Equality 3 becomes

$$\lambda_2 = b_2^+ + b_2^- + 1 + b_1^- + 1 + ZY - M_A + Y_1 + N + \frac{1}{2}(P - ZY + M_A - Y_1 - N - \epsilon_1)$$

which leads us to (since $N \geq M_A$)

$$w - \lambda = \lambda_2 \geq b_2^+ + b_2^- + b_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{3}{2}$$

For these two sub-cases, we have

$$w - \lambda \geq b_2^+ + b_2^- + b_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{3}{2} \qquad (4)$$

Inequalities 2 and 4 imply:

$$\delta = \frac{w - \lambda_{DLA}}{w - \beta(L)} \geq \frac{b_2^+ + b_2^- + b_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{3}{2}}{\frac{3}{2}b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{9}{2}}$$

Since $Z_1^- \leq b_1^- + 1$ and $Z_2^- \leq 2b_2^- + 2$, we have

$$\delta \geq \frac{b_2^+ + b_2^- + b_1^- + [\frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P] + \frac{3}{2}}{\frac{3}{2}b_2^+ + 2b_2^- + b_1^- + [\frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P] + 6}$$

As the above expression increases in $[\frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P]$, we have

$$\delta \geq \frac{b_2^+ + b_2^- + b_1^- + \frac{3}{2}}{\frac{3}{2}b_2^+ + 2b_2^- + b_1^- + 6}.$$

Since $x = \lceil \frac{1}{3}(\beta_2^- - \beta_1^-) \rceil$, we have $b_1^- = b_2^- + \epsilon$, with $\epsilon \leq 2$. So,

$$\delta \geq \frac{b_2^+ + 2b_2^- + \epsilon + \frac{3}{2}}{\frac{3}{2}(b_2^+ + 2b_2^- + \frac{2}{3}\epsilon + 4)} = \frac{2}{3}(1 - \frac{15 - 2\epsilon}{6b_2^+ + 12b_2^- + 4\epsilon + 24}) = \delta_1(\epsilon).$$

case 2. $b_1^- + 1 > Z_1^- + Z_2^- - 2$
In this case, the algorithm uses at least $Z_2^- + Z_1^- - 2$ items of type z_i to complete bins in $B_1^- \cup \{x_3\}$, i.e. $Z_A \geq Z_2^- + Z_1^- - 2$. We also know by definition of Z_A that it satisfies $Z_A \leq b_1^- + 1$. So, if $b_1^- + 1 > Z_2^- + Z_1^- - 2$ (this is the case in this part) we have $Z_2^- + Z_1^- - 2 \leq Z_A \leq b_1^- + 1$. we put $Z_A = Z_2^- + Z_1^- - 2 + M$ ($M \geq 0$). We then distinguish two sub-cases.

case 2.1 $Z_A = b_1^- + 1 > Z_1^- + Z_2^- - 2$

In this first sub-case, each 1-bin of $B_1^- \cup \{x_3\}$ is completed (at the second step) by items of type z_i; but the number of z_i-items used by the algorithm to complete bins of $B_1^- \cup \{x_3\}$ is greater than the number of z_i-items used by the optimal algorithm to form bins of type $\{x_i, z_i\}$, with $x_i \in B_1^- \cup B_2^-$.
Let us note that after forming Y_{1A} bins of type $\{y_i, x_i\}$, $x_i \in B_1^+$, and completing the $b_1^- + 1$ bins of $B_1^- \cup \{x_3\}$ with items of type z_i, two situations may occur:

a): $ZY_A = ZY + M_A$, with $M_A \geq 0$. This corresponds to the case where the algorithm has enough items of type z_i and y_i to form at least ZY bins of type $\{z_i, y_i\}$; (recall that before forming $\{z_i, y_j\}$ bins, the algorithm had to form Y_{1A} bins of type $\{y_i, x_i\}$, $x_i \in B_1^+$ and $Z_A = b_1^- + 1$ bins of type $\{z_i, x_j\}$ with $x_j \in B_1^- \cup \{x_3\}$). We have

$$\lambda_2 = b_2^+ + b_2^- + 1 + b_1^- + 1 + ZY_A + Y_{1A} + \frac{1}{2}(P - ZY_A - Y_{1A} - \epsilon_1)$$

As $ZY_A = ZY + M_A$ and $Y_{1A} = Y_1 + N$, with $M_A \geq 0$ and $N \geq 0$, the last equality implies (*since* $\epsilon_1 \leq 1$):

$$\lambda_2 \geq b_2^+ + b_2^- + b_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + \frac{3}{2}.$$

This is inequality (4), the one of case 1.

b) $ZY_A = ZY - M_A$, with $M_A > 0$. This is the case where the algorithm does not have enough items of type z_i and/or it does not have enough items of type y_i to form at least ZY bins of type $\{z_i, y_i\}$. The equality $\lambda_2 = b_2^+ + b_2^- + 1 + b_1^- + 1 + ZY_A + Y_{1A} + \frac{1}{2}(P - ZY_A - Y_{1A} - \epsilon_1)$ implies (since $b_1^- + 1 = Z_A = Z_2^- + Z_1^- - 2 + M$) :

$$\lambda_2 \geq b_2^+ + b_2^- + 1 + \frac{1}{2}(Z_2^- + Z_1^- - 2 + M) + \frac{1}{2}(Z_2^- + Z_1^- - 2 + M) + \frac{1}{2}ZY_A + \frac{1}{2}Y_{1A} + \frac{1}{2}P - \frac{1}{2} \quad (5)$$

Let us recall that $ZY_A = ZY - M_A$ and $Y_{1A} = Y_1 + N$. The last inequality, (5), becomes:

$$\lambda_2 \geq b_2^+ + b_2^- + \frac{1}{2}(b_1^- + 1) + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P - \frac{1}{2} + [\frac{1}{2}M + \frac{1}{2}N - \frac{1}{2}M_A], \quad (6)$$

As $M + N \geq M_A$, (In the opposite case, revisiting Remark 3, one can see that the optimal solution cannot contain ZY bins of type $\{z_i, y_i\}$), inequality 6 leads us to

$$w - \lambda = \lambda_2 \geq b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P.$$

Then, demonstration similar to the one of *case 1* leads us to

$$\delta \geq \frac{2}{3}(1 - \frac{6 - b_2^- - \frac{1}{3}\epsilon}{2(b_2^+ + b_2^- + \frac{1}{3}\epsilon + 3)}) = \delta_{2.1}(\epsilon).$$

case 2.2. $b_1^- + 1 > Z_A$.
Here, some items in $B_1^- \cup \{x_3\}$ cannot be matched by z_i-items. So, y_i-items can be used to complete bins in $B_1^- \cup \{x_3\}$ and transform them in 2-bins. Revisiting algorithm LA, one can see that before completing bins in $B_1^- \cup \{x_3\}$ by y_i-items, the algorithm has to form $Y_{1A} + ZY_A$ bins; each bin containing an item of type y_i. So it remains $p - Y_{1A} - ZY_A$ items of type y_i. We can distinguish the following two sub-cases.

case 2.2.1 $P - Y_{1A} - ZY_A \geq b_1^- + 1 - Z_A$
Here, all items in $B_1^- \cup \{x_3\}$ which have not been completed by a z_i-item are now completed by items of type y_i.

Then, $w - \lambda = \lambda_2 = b_2^+ + (b_2^- + 1) + (b_1^- + 1) + Y_{1A} + ZY_A + \frac{1}{2}(P - Y_{1A} - ZY_A - b_1^- - 1 + Z_A - \epsilon_1)$
Arguments similar to the one of item (b) of case 2.1 yield:

$$w - \lambda = \lambda_2 \geq b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P,$$

which brings us back to the last case 2.1.

case 2.2.2. $p - Y_{1A} - ZY_A < b_1^- + 1 - Z_A$
In this case, some items in $B_1^- \cup \{x_3\}$ which are not completed by a z_i-item will not be matched by y_i-items. Only $Z_A + P - Y_{1A} - ZY_A$ items in $B_1^- \cup \{x_3\}$ have been matched by items of the second cluster.
Then, $w - \lambda = \lambda_2 \geq b_2^+ + (b_2^- + 1) + Y_{1A} + ZY_A + Z_A + P - Y_{1A} - ZY_A$, i.e: $w - \lambda \geq b_2^+ + b_2^- + P + Z_2^- + Z_1^- - 1$, since $Z_A \geq Z_2^- + Z_1^- - 2$.
We know that $w - \beta \leq \frac{3}{2}b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + \frac{1}{2}ZY + \frac{1}{2}Y_1 + \frac{1}{2}P + 9/2$.
Moreover, $ZY + Y_1 \leq P$, so:

$$w - \beta \leq \frac{3}{2}b_2^+ + b_2^- + \frac{1}{2}b_1^- + \frac{1}{2}Z_2^- + \frac{1}{2}Z_1^- + P + \frac{9}{2}$$

By considering once again sub-cases $ZY_A = ZY + M_A$ and $ZY_A = ZY - M_A$ (as in the case 2.1) and using similar arguments to the ones of case 1, we have:

$$\delta \geq \frac{2}{3}(1 - \frac{12 + \epsilon}{3b_2^+ + 3b_2^- + \epsilon + 9})$$

It is easy to verify that $\delta_1(\epsilon)$ and $\delta_{2.1}(\epsilon) = \delta_{2.2.1}(\epsilon)$ increase in ϵ and $\delta_{2.2.2}(\epsilon)$ is decreasing in ϵ. It implies that ($\epsilon \in]0, 2]$):
case 1:

$$\delta \geq \delta_1(0) = \frac{2}{3}(1 - \frac{15}{6b_2^+ + 12b_2^- + 24}) = \frac{2}{3}(1 - \frac{5}{2b_2^+ + 4b_2^- + 8})$$

case 2.1:

$$\delta \geq \delta_{2.1}(0) = \frac{2}{3}(1 - \frac{6 - b_2^-}{2(b_2^+ + b_2^- + 3)})$$

case 2.2.1:

$$\delta_{2.2.1}(0) = \delta_{2.1}(0)$$

case 2.2.2:

$$\delta \geq \delta_{2.2.2}(2) = \frac{2}{3}(1 - \frac{14}{3(b_2^+ + b_2^-) + 11}).$$

It is easy to prove that $\delta_{2.2.2}(2) \leq \delta_{2.1}(0)$ and $\delta_{2.2.2}(2) \leq \delta_1(0)$, which means

$$\delta \geq \delta_{2.2.2}(2) = \frac{2}{3}(1 - \frac{14}{3(b_2^+ + b_2^-) + 11}).$$

In what follows, we give an hardness result that account, on the one hand, for the difficulty of the LBP problem and, on the other hand, for the quality of the algorithm developed to solve the $2 - steps$ on-line bin-packing problem.

Theorem 3. *If the final instance is revealed in two clusters, then no algorithm can guarantee a differential competitivity ratio greater than* $2/3$*, for* LBP*.*

Proof. Let us consider the following lists, L_1 and L_2, where L_1 contains n items equal to $\frac{1}{2} - \epsilon$ and L_2 contains also n items all identical to $\frac{1}{2} + \epsilon$. Let L be $L_1 \cup L_2$. Applying any algorithm A to instance L_1, it returns a solution B_1 composed of two types of bins: 1-bins and 2-bins. Moreover, let us suppose that the second cluster to be revealed is empty and let x_A be the number of 1-bins returned by A; then we have $\frac{n-x_A}{2}$ 2-bins. i.e., $\lambda_A(L_1) = x_A + \frac{n-x_A}{2} = \frac{n+x_A}{2}$, while $\beta(L_1) = \frac{n}{2}$. Consequently $w(L_1) - \lambda_A(L_1) = \frac{n-x_A}{2}$ and $w(L_1) - \beta(L_1) = \frac{n}{2}$. Therefore, $[w(L_1) - \lambda_A(L_1)]/[w(L_1) - \beta(L_1)] = \frac{n-x_A}{n}$. Let us now assume that the second cluster is not empty. It is exactly equal to L_2 which contains n items all identical to $\frac{1}{2} + \epsilon$. Let us remark that $|L| = |L_1 \cup L_2| = 2n$ and $\beta(L) = n$. Applying algorithm A to instance $L = L_1 \cup L_2$, in best case, A completes the x_A 1-bins returned at the first step to get x_A 2-bins. And at the end of the on-line process, in best case, algorithm A returns exactly $n - x_A$ 1-bins (composed of items of L_2). So, $\lambda_A(L) \geq \frac{n+x_A}{2} + n - x_A = \frac{3n-x_A}{2}$.
It leads us to: $w(L) - \lambda_A(L) \leq \frac{n+x_A}{2}$ and $w(L) - \beta(L) = n$ (recall that $\beta(L) = n$ and $w(L) = 2n$). Then,

$$[w(L) - \lambda_A(L)]/[w(L) - \beta(L)] \leq \frac{n + x_A}{2n},$$

which implies, for every algorithm A:

$$\delta_A \leq min\left\{\frac{n - x_A}{n}; \frac{n + x_A}{2n}\right\} \text{ and } \max_A\{\delta_A\} \leq \max_A \left\{min\{\frac{n - x_A}{n}; \frac{n + x_A}{2n}\}\right\}.$$

Since $\max_A \left\{min\{\frac{n-x_A}{n}; \frac{n+x_A}{2n}\}\right\}$ is tight when $\frac{n-x_A}{n} = \frac{n+x_A}{2n}$, i.e., $x_A = \frac{n}{3}$, we can conclude that $\max_A\{\delta_A\} \leq 2/3$, which also concludes the proof of the theorem.

All those results are displayed in the following table.

Problems	**on-line version**		
	ratio	*specific hardness results*	*general hardness results*
Standard LBP	$\delta_{FF} \geq 1/2$	$\delta_{FF} \leq 1/2$	$\delta \leq 1/2$
(LBP, I)	$\delta_{FF} \geq 1/2$	$\delta_{FF} \leq 1/2$	$\delta \leq 1/2$
(LBP, D)	$\delta_{FFD} \geq 3/4$	$\delta_{FFD} \leq 3/4$	?
LBP 2-clusters	$\delta^{\infty}_{DLA} \geq 2/3$	$\delta_{DLA} \leq 2/3$	$\delta \leq 2/3$

4 On-line bin-packing problem: *LIB* version

The on-line bin-packing problem with longest items at the bottom (LIB) is defined below: in any bin, for any pair of items i and j, if the size of j is greater than the one of i, then j should be placed into the bin before i. For this version, we first deal with the differential competitivity ratio guaranteed by algorithms First Fit (FF) and Best Fit (BF). Then we study the limits of these algorithms (hardness results) and finally we give another hardness result available for every on-line algorithm solving ($LBP - LIB$), the on-line bin-packing problem with LIB constraint. It is proved in [13] (under the standard approximation ratio) that no algorithm can do better than 1.76 while First Fit can do better than 2. Let $w(I)$, $\lambda(I)$ and $\beta(I)$ be respectively the worst solution, the solution returned by an on-line algorithm A and the **on-line optimal** solution of the instance L. Moreover, let put $\delta_A(I) = [w(I) - \lambda(I)]/[w(I) - \beta(I)]$; then the quantity $\delta_A = sup\{r : \delta_A(I) \geq r, I\ instance\ of\ (BP - LIB)\}$ denotes the differential competitivity ratio guaranteed by the algorithm A. We emphasis that here, we use the **on-line optimum**. It is easy to verify that lemma 1 (section 2) can be adapted to the LIB version. We now give the differential ratio provided by *First Fit* when it is used to solve the LIB-version of the bin-packing problem.

Theorem 4. *$\delta_{FF} \geq 1/2$; Moreover, this bound is tight.*

Proof. We prove that the differential ratio guaranteed by *First Fit*, FF, when applied on a worst-instance is at least 1/2. Let L be a worst-instance for the pair $(LBP - LIB, FF)$.
If $\lambda_{FF}(L) = w(L) = n$, then $\lambda_{FF}(L) = \beta(L)$. Indeed, if $\lambda_{FF}(L) = w(L) = n$ then, for all pair (x, y) of items in L, we have the two cases $x + y > 1$ or $x + y \leq 1$ and in this last case, $\max\{x, y\}$ has been revealed after $\min\{x, y\}$. Therefore the optimal **on-line** algorithm will pack each item in its own bin, i.e., $\beta(L) = \lambda_{FF}(L) = w(L) = n$. This situation is not possible if L is a worst instance for $(LBP - LIB)$ since for such an instance, algorithm A has the worst

behavior. As $\lambda_{FF}(L) \neq n$, there exists a bin $b \in FF(L)$ containing at least two items. Let us set $L^{'} = L \setminus b = (\bigcup_{b_i \in FF(L)} b_i) \setminus b$. The list $L^{'}$ satisfies the hypothesis of the lemma 1 page 3, with $x = 0,\ y = 1$ and $z = |b| \geq 2$, which implies $\delta_A(L) \geq \frac{(z-1)}{z}$. As the function $\frac{(z-1)}{z}$ is increasing in z and $z \geq 2$, we have $\delta_A(L) \geq \frac{1}{2}$. The analysis being made with a worst-instance, we can conclude that, for all L, $\delta_A(L) \geq \frac{1}{2}$, which means $\delta_{FF}(L) \geq \frac{1}{2}$.
In order to prove that this ratio is tight, we construct an instance L of size $4k$ (k is an integer) for which the algorithm achieves the ratio 1/2. The instance L is the concatenation of lists $L_1 = \{\frac{1}{2}, \cdots, \frac{1}{2}\}$ and $L_2 = \{\frac{1}{2} - 2k\epsilon, \frac{1}{2} - (2k-1)\epsilon, \cdots, \frac{1}{2} - \epsilon\}$ (each list is of size $2k$). $L = L_1 \cup L_2$. Algorithm FF packs L_1 in exactly k bins when it packs the $2k$ elements of the list L_2 in $2k$ bins (one item per bin), since list L_2 is increasing and the LIB constraint does not allow us to pack a longer item above a smaller one. Therefore, $\lambda_{FF}(L) = 3k$. As an optimal solution requires $2k$ bins of type $\{\frac{1}{2}, \frac{1}{2} - j\epsilon\},\ j = 1, \cdots 2k$, we have

$$\frac{n - \lambda_{FF}}{n - \beta(L)} = \frac{4k - 3k}{4k - 2k} = \frac{1}{2}.$$

We have previously given the limits of the First Fit Algorithm for $(LBP - LIB)$. We now prove that this hardness result holds for every algorithm solving $(LBP - LIB)$.

Theorem 5. *Let A be an on-line algorithm solving $(LBP - LIB)$. If A guarantees a competitivity ratio δ, then $\delta \leq \frac{1}{2}$ i.e no on-line algorithm for $(LBP - LIB)$ can guarantee a competitivity ratio $\delta > 1/2$.*

Proof. Let us consider an on-line algorithm for $(LBP - LIB)$ guaranteeing a differential competitivity ratio $\delta > 1/2$. The following lemma can be seen as the LIB version of remark 1.

Lemma 3. *Let A be an on-line $(LBP - LIB)$-algorithm guaranteeing a differential competitivity ratio $\delta > 1/2$. Then A returns two 2-bins for the 4-steps instance $L_1 = \{\frac{5}{12} + 4\epsilon, \frac{5}{12} + 3\epsilon, \frac{5}{12} + 2\epsilon, \frac{5}{12} + \epsilon\}$.*

The proof of this lemma is similar to the one of the remark 1: as remark 1 also holds for the LIB version of LBP, it is easy to see that if A does not return two 2-bins, it never guarantees a ratio $\delta > 1/2$ for the 4-steps instance here considered.
Let us come back to the proof of the theorem. It suffices to consider the list L which is the concatenation of the following sub-lists:
$L_1 = \{\frac{5}{12} + 4\epsilon, \frac{5}{12} + 3\epsilon, \frac{5}{12} + 2\epsilon, \frac{5}{12} + \epsilon\}$, $\quad L_1^{'} = \{\frac{1}{3} + \epsilon, \frac{1}{3} + 2\epsilon, \frac{1}{3} + 3\epsilon, \frac{1}{3} + 4\epsilon\}$.

$L_2 = \{\frac{5}{12} + 6\epsilon, \frac{5}{12} + 5\epsilon\}$, $\quad L_2^{'} = \{\frac{1}{3} + 5\epsilon, \frac{1}{3} + 6\epsilon\}$.

$\vdots \qquad \vdots$

$L_k = \{\frac{5}{12} + 2(k+1)\epsilon, \frac{1}{2} + (2k+1)\epsilon\}$, $\quad L_k^{'} = \{\frac{1}{3} + (2k+1)\epsilon, \frac{1}{3} + 2(k+1)\epsilon\}$.
$L = \bigcup_{k \geq 1}(L_k \cup L_k^{'})$ where items of L are revealed in order $L_1, L_1^{'}, L_2, L_2^{'}, \cdots, L_k, L_k^{'}$ i.e. items of L_1 are first revealed, then the ones of $L_1^{'}$ follow and so on. Using

the LIB constraint and arguments similar to the ones of the theorem 1, one can easily conclude the proof of the theorem.

We now study some particular cases.
1.**First Fit: increasing sizes**
Here, if an item x_i is revealed before item x_j, then $x_i \geq x_j$. For every bin b, we denote by *totalsize* the sum of the items held by b. The FF algorithm works as follows: if an arriving item i is greater than its predecessor $i-1$, it is placed in its own (new) bin b_{k+1}. Otherwise, if $x_i = x_{i-1}$, place i on top of $i-1$ in b_k (unless $x_i > 1 - totalsize(k)$, in this case, i is placed into a new bin). Clearly, an exact algorithm (one that always returns an optimal solution) cannot do better than FF. It follows that, for these instances, the competivity ratio of algorithm FF is one.
2. **First Fit: decreasing sizes**
This version is identical to the case with no LIB-constraints. So all results we got for (LBP, D) also hold for $(LBP - LIB, D)$.
As for the general hardness result of algorithms solving the two-clusters on-line bin-packing with LIB constraint, we provide a similar theorem to theorem 3.

Theorem 6. *If the final instance of the bin-packing problem with LIB constraint is revealed in two clusters, then no algorithm can guarantee a competitivity ratio greater than 2/3.*

Proof. it suffices to consider the two clusters $L_1 = \{\frac{1}{2}, \cdots, \frac{1}{2}\}$ (2k items) $L_2 = \{\frac{1}{2} - 2k\epsilon,\ \frac{1}{1} - (2k-1)\epsilon, \cdots, \frac{1}{2} - \epsilon, \}$. Then, the rest of the proof is exactly similar to the one of theorem 3.

5 Conclusion

In this paper,we have considered three versions of the on-line bin-packing problem. For each one, we used an approximation criterion called differential ratio to analyse the competitivity behavior of algorithms developed to solve it. In both $n-steps$ and $2-steps$ versions, we proved that no algorithm can do better than the ones here developed. So, $n-steps$ and $2-steps$ on-line bin-packing problems are well solved under the differential approximation criterion. Our hardness results also imply that the n-steps and 2-steps bin-packing problem do not admit a differential approximation schema, contrary to the off-line framework. We also studied the on-line bin-packing problem with the additional constraint that the items have to be placed in bins in the order of their length. The longest item is required to be placed at the bottom (the LIB constraint). For this problem, we prove that, the $n-steps$ version (items are revealed one by one) is well studied with the differential approximation, since $\delta_{FF} \geq 1/2$ and no algorithm can do better than 1/2. For the $2-steps$ LIB version, no algorithm can do better than 2/3. An open problem is to develop an algorithm guaranteeing differential ratio better than 1/2.

References

1. *M.Demange, Jérôme Monnot, V.Th.Paschos.* Maximizing the number of unused bins. Foundation of Computing and Decision Sciences 26(2), 145-168, 2001.
2. *M.Demange.* Reduction off-line to on-line: an example and its applications. Yugoslav Journal of Operations Research 13 (1), 3-24, 2003.
3. *M.Demange et V.Th.Paschos.* Two-steps combinatorial optimization. Proc. workshop OLCP'01, pp. 37-44, Ed. S. de Givry and J. Mattioli, Thales R D, december 2001
4. *M. Demange, J. Monnot et V. Th. Paschos.* Bridging gap between standard and differential polynomial approximation: the case of bin-packing. Applied Mathematics Letters, vol. 12, pp. 127-133, 1999.
5. *M.M.Halldòrsson.* Approximations via partitioning. JAIST,Japan Advanced Institute of Science and technology, Japan, 1995.
6. *P.Crescenzi et V.Kann.* A compendium of NP optimization problems. URL: http://www.nada.kth.se/∼ viggo/problemlist/compendium.html , 1995.
7. *A. Van Vliet.* An improved lower bound for the on-line bin-packing algorithms. *Inform. process. Lett., 43:277-284, 1992.*
8. *F. M. Liang.* A lower bound for the on-line bin-packing. *inform. process. lett.,* 10(2):76-79, 1980.
9. *M. B. Richey.* Improved bounds for harmonic-based bin-packing algorithms. *Discr. Appl. Math.,* 3':203-22è, 1991.
10. *M.R.Garey and D.S.Johnson.* Computers and intractability. A guide to the theory of NP-completeness. CA.Freeman, San Francisco, 1979.
11. *X.Paradon.* Algorithmique on-line. Thèse de doctorat, Université Paris Dauphine, 2000.
12. *P. Manyem.* Bin-packing and covering with longest item at the bottom: the *The ANZIAN journal* 43 no.E, E186–E231, 2002.
13. *P. Manyem, R. L. Salt and Visser.* Lower bound and heuristic for the on-line LIB bin-packing and covering. *Proceedings of the thirteenth australian workshop of combinatorial algorithms*, 2002
14. *P. Manyem.* Uniform sized Bins
15. *L. Finlay and P. Manyem* Online LIB problems: Heuristics for the