Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Semi On-line Bin Packing with Cardinality Constraint

Paolo Dell'Olmo[1], Giorgio Gambosi[2], Gaia Nicosia[3]

RT-DIA-58-2000 **Ottobre 2000**

(1) Dipartimento di Statistica, Probabilità e Statistica Applicata,
University of Rome "La Sapienza",
Piazzale Aldo Moro 5, 00185 Rome, Italy.
(dellolmo@iasi.rm.cnr.it)

(2) Dipartimento di Matematica,
University of Rome "Tor Vergata",
Via della Ricerca Scientifica, 00133 Rome, Italy.
(gambosi@mat.uniroma2.it)

(3) Dipartimento di Informatica e Automazione,
University of Roma Tre,
Via della Vasca Navale, 79
00146 Rome, Italy.
(nicosia@dia.uniroma3.it)

ABSTRACT

In this paper we study the on-line version of the k-item bin packing problem. A semi on-line algorithm with competitive ratio bounded by 2 and running time $O(n \log n)$ is proposed. The previous known result for an on-line algorithm is $2 + f(k)$, where $0 < f(k) < 0.43$ for any $k \geq 3$ and $f(k)$ tends to 0 as k goes to infinity. Moreover, for the particular case of $k = 2$, it is proved that the algorithm yields a competitive ratio of $5/4$, which is shown to be tight.
Keywords: Analysis of algorithms, On-line algorithms, Competitive analysis.

1 Introduction

In the classical and well known bin-packing problem we are given a set of items $L = (a_1, a_2, \ldots, a_n)$, with $a_i \in (0, 1]$ and we want to pack them into a minimum number of unit capacity bins. This problem is one of the most extensively studied in operations research and computer science. However, in several practical applications it is required that not only the size but also the *cardinality* of the set of items packed in each bin must be limited. The *bin packing with cardinality constraint* (or *k-item bin packing*) problem, introduced in [11] and studied also in [1, 10], is defined as a bin packing with the additional constraint that, for a given integer $k > 0$, at most k items can be packed into each bin.

As noticed in those papers, several scheduling and resource allocation problems where a cardinality constraint is imposed (by normative rules or technical reasons) can be modeled as the k-item bin packing problem. Other applications examples are storage/loading problems where two kinds of constraints, such as total weight and maximum number of loading units (e.g. containers) must simultaneously be satisfied and, the cutting and packing problems for which the k-cardinality bin packing is a particular case (see [6, 15] for an example).

In this paper we are interested to scenarios in which the bin packing with cardinality constraint problem must be solved on-line, that is, we require that algorithms must take decisions with only a limited knowledge of the problem instance. In this case, this corresponds to ask an algorithm to deal with a sequence of requests and serve each request by knowing neither the following requests nor the overall length of the sequence.

Let us remark that the term on-line applied to bin packing algorithms (such as First Fit, Bets Fit, and Harmonic$_M$) often corresponds to the tighter assumption that an item, once assigned, cannot be moved to a different bin. In this paper, we relax such an assumption by referring to the original characterization of on-line algorithms given for example in [2, 14]. In particular, we allow that, when satisfying a new request, an algorithm performs a limited (constant) number of reassignments of items to different bins. This model has been introduced in the case of bin packing in [4] and further studied in [5] and [9].

The performance of an on-line algorithm is usually compared with the performance of an optimal off-line algorithm that knows the sequence of events in advance. The maximum ratio between their respective performances, taken over all sequences, is denoted as the *competitive ratio*.

In this paper we introduce a semi on-line algorithm for the bin packing with cardinality constraint problem with competitive ratio bounded by 2 and running time $O(n \log n)$. The previous known result for an on-line algorithm is $2 + f(k)$, where $0 < f(k) < 0.43$ for any $k \geq 3$ and $f(k)$ tends to 0 as k goes to infinity. Moreover, in the case $k = 2$, it is proved that the algorithm yields a worst case performance ratio of 5/4, which is shown to be tight.

The remaining of the paper is organized as follows. In Section 2 we briefly describe the state of the art and recall some aspects of on-line algorithms for the bin packing problem necessary to make the paper self contained. We also show that the competitive ratio obtained by applying directly one of the best known algorithms for bin packing, namely Harmonic$_M$, to bin packing with cardinality constraint is strictly greater than 2. In Section 3 we describe our algorithm, denoted as k-Harmonic, and analyze its competive ratio both in the general case and for the case with $k = 2$. Finally, some conclusions are drawn.

2 On-line algorithms for bin packing

The standard bin packing problem is well known to be NP-hard ([7]) and several approximation algorithms have been proposed in the literature (see [3] for a survey).

The bin packing with cardinality constraint problem can be seen as a special case of the 2-dimensional vector packing where the second weight of each item equals to $1/k$ ([8]). It was first studied in [11, 12] where two heuristics with performance ratio 2 were proposed. The classical bin packing algorithms were also analyzed [6]. The performance ratio of First Fit algorithm is $R_{FF} = 2.7 - \frac{12}{5k}$, while that of First Fit Decreasing is $R_{FFD} = 2 - \frac{2}{k}$, and Best Fit $R_{BFD} = 2 - \frac{2}{k}$. Recently, an off-line algorithm, denoted as Pack, has been proposed by Kellerer and Pferschy [10] with an asymptotic performance ratio bounded by 3/2.

A common feature of these works consists in the assumption that complete information is available (off-line setting). Here, we consider the on-line version of the problem. This problem has been studied also by Babel and Kellerer ([1]), who propose an algorithm which has a worst-case performance ratio depending on k. Their algorithm produces a solution that, asymptotically, is at most $2 + f(k)$ times the optimal one, where $f(k)$ assumes positive values smaller than 0.43 and tends to 0 as k goes to infinity.

In this paper, we slightly relax the on-line restriction by allowing bounded repacking, i.e. each assignment of an item to a bin may be accompanied by a limited amount of rearrangements of the items in the bin (see [3] for a summary of results on on-line bin packing and [4, 5, 9] for semi on-line algorithms). One of the best known on-line algorithms, $HARMONIC_M$, is described next.

2.1 The algorithm $HARMONIC_M$

The algorithm $HARMONIC_M$ was proposed in [13]. Let us briefly summarize it. Given a set of items $L = (a_1, a_2, \ldots, a_n)$, where $a_i \in (0, 1]$ for all i, each item a_i is classified in the following way. The unit interval (0,1] is partitioned in M intervals $\{I_1, I_2, \ldots, I_M\}$, where $I_j = (1/(j+1), 1/j]$ for $j = 1, \ldots, M-1$ and $I_M = (0, 1/M]$. An item a_i is called a I_j-item (or of type I_j) if $a_i \in I_j$. Similarly, the bins are classified in M categories and each bin is designated to pack items of one type only. The asymptotic worst case performance ratio of $HARMONIC_M$ is bounded by $1.69103\ldots$ ([13]), however when applied to bin packing with cardinality constraint, the algorithm has a competitive ratio strictly grater than 2, as proved by the following theorem.

$HARMONIC_M$ when applied to bin packing with cardinality constraint, has a competitive ratio at least 5/2 for $k > 4$, 7/3 for $k \geq 7$, and 8/3 for $k \geq 20$. Let $M = k > 4$. Consider an instance where the input consists of:

2 I_1-items of size $1/2 + \epsilon$ and

2 I_2-items of size $1/3 + \epsilon$,

$2k - 4$ I_k-items of size ϵ,

where $1/2 + 1/3 + k\epsilon \leq 1$. $HARMONIC_M$ uses 5 bins (1 bin of type I_3, 2 bins of type I_2 and 2 I_k bins), while the optimal solution uses 2 bins, each one filled with 1 I_3 item of size $1/3 + \epsilon$, 1 I_2 item of size $1/2 + \epsilon$ and $(k-2)$ I_k items of size ϵ.

Moreover, let us consider the following instance with ϵ such that $1/2+1/3+1/7+k\epsilon \leq 1$, i.e. $\epsilon \leq 1/(42k)$. Let $k \geq 7$. Let the input be:

6 I_1-items and of size $1/2 + \epsilon$

6 I_2-items of size $1/3 + \epsilon$

6 I_3-items of size $1/7 + \epsilon$

$6(k-3)$ I_k-items of size ϵ

$HARMONIC_M$ uses $6 + 3 + 1 + \left\lceil \frac{6(k-3)}{k} \right\rceil$ bins, while the optimal solution consists of 6 bins, each filled with one I_1-item, one I_2-item, one I_3-item and $(k-3)$ I_k-items. Therefore, the competitive ratio of $HARMONIC_M$, R_{HM}, applied to the bin packing problem with cardinality constraints is

$$R_{HM} \geq \frac{10 + \left\lceil \frac{6(k-3)}{k} \right\rceil}{6} \geq \begin{cases} 7/3 & \text{if } k < 20 \\ 8/3 & \text{if } k \geq 20 \end{cases}$$

The above result calls for a different algorithm which is described and analyzed in the next section.

3 Algorithm k-Harmonic

In this section we describe a new algorithm, denoted as k-Harmonic, and analyze its performance.

Let us denote as l_i the *load* on bin i, i.e. the sum of the weights of the items packed in bin i; moreover, let the *gap* of bin i be equal to $1 - l_i$. All bins and items are classified like in $HARMONIC_M$ with $M = k$. Items of type $I_1, I_2, \ldots I_{k-1}$ are assigned permanently to bins, while I_k pieces are put in I_k bins, but may be rearranged, i.e. moved to another bin, therefore these bins are called *temporary*. Bins of type I_j with $j = 1, 2, \ldots, k-1$ are called *closed* if they contain k elements or if their load is 1, *open* if they have $h < j$ items (notice that there is only one open bin of type I_j for each $j = 1, 2, \ldots, k-1$), *filled* if they have j items of type I_j and h, with $0 \leq h < k - j$, of type I_k.

Roughly speaking, when the algorithm has assigned the j-th item to a bin of type I_j with $j \neq k$, then we pack in it (if possible) the smallest $k - j$ items of type I_k.

First of all, the algorithm opens one bin of each type. Then, suppose we are considering the r-th item of the input sequence with size a_r. Let the item be of type I_j. Suppose $j \neq k$, then the algorithm will perform the following actions:

1. The item r is assigned to the open bin B of type I_j.
2. If B becomes filled, then a new I_j bin is opened. In this case, if B is not full and if there are temporary bins then:
   (a) Let i_{min} be the item with minimum size in the temporary bins and let a_{min} be its size.

(b) If i_{min} fits in B then it is removed from the temporary bin it belongs to and assigned permanently to B.

(c) If B is not full and contains less than k items then go to step 2a. Else make bin B closed.

If $j = k$ then the algorithm will perform the following actions:

1. If there are no temporary bins, then:

   (a) If the item r fits in one of the filled bins, then r is assigned permanently to it.

   (b) Else r is assigned to a newly created temporary bin.

2. Else:

   (a) If a_r is smaller than a_{min}, then if the item r fits in one of the filled bins, it is assigned permanently to it.

   (b) Else add r to any one of the temporary bins or to a newly created one.

The sizes of the items in temporary bins and the different gaps of filled bins, together with references to the corresponding bins, are contained in two different dictionaries where the usual operations *ExtractMin, ExtractMax, Delete, Insert* are defined.

It can easily be verified that the complexity of the algorithm is $O(n \log n)$ since each item of the input sequence of length n is moved from the initially assigned bin at most once.

3.1 Competitive Analysis

After n tasks have been assigned we have the following situation. There will be:

- m_1 closed bins, that is bins of type $I_1, I_2, \ldots, I_{k-1}$, each filled with exactly k elements (notice that for a bin of type I_j, $k - j$ such items are of type I_k);
- m_2 filled bins, that is bins of type $I_1, I_2, \ldots, I_{k-1}$, each filled with $h_1, h_2, \ldots, h_{m_2} < k$ items (a bin of type I_j containing h items will have j items of type I_j and $h - j$ items of type I_k.);
- m_3 temporary bins of type I_k, each containing k elements.

During the competitive analysis we consider two different lower bounds on the optimum, that is $OPT \geq \sum_{i=1}^{n} a_i$ and $OPT \geq \frac{n}{k}$. The first lower bound dominates when m_1 and m_2 are greater than m_3, while the other bound dominates when there are many closed and temporary bins, i.e. m_1 and m_3 are quite larger than m_2.

Algorithm k-Harmonic for $k \geq 3$ is at most $(2 - \epsilon)$-competitive. If $m_2 = 0$, then algorithm k-Harmonic finds the optimal solution, since it fills all the temporary bins with k items. If $m_3 = 0$ then the algorithm packs items at least as efficiently as $HARMONIC_M$: therefore the competitive ratio, in this case, is at most 1.75 for $k \geq 3$.

Let us consider the case when $m_1 = 0$. Let A_3 be the average size of the items in temporary bins. We may use this value to refine both lower bounds previously introduced.

(*i*) $\sum_{i=1}^{n} a_i \geq kA_3m_3 + (1 - A_3)m_2$
(since each temporary bin is in average filled for kA_3 and each filled bin has a gap smaller than any element in temporary bins);

(*ii*) $\frac{n}{k} \geq \frac{m_2}{k} + m_3$.

Let H be the number of bins used by Algorithm k-Harmonic. Clearly, the competitive ratio of k-Harmonic is $R_{kH} \leq \min\{R_1, R_2\}$, where $R_1 = \frac{H}{\sum_{i=1}^{n} a_i}$ and $R_2 = \frac{H}{n/k}$. Therefore, we have to prove that at least one of the following two inequalities holds for some $\rho \leq 2$.

$$R_1 \leq \frac{m_2 + m_3}{kA_3m_3 + (1 - A_3)m_2} \leq \rho \tag{1}$$

$$R_2 \leq \frac{m_2 + m_3}{\frac{m_2}{k} + m_3} \leq \rho \tag{2}$$

Let $c = \frac{m_3}{m_2}$. Clearly, $R_1 \leq \rho$ if

$$\frac{1 + c}{kA_3c + 1 - A_3} \leq \rho, \tag{3}$$

that is, since $A_3 < 1$, if

$$c(1 - \rho kA_3) \leq \rho - \rho A_3 - 1.$$

As a consequence, we have $R_1 \leq \rho$

1. if $1 - \rho kA_3 > 0$ for
$$c \leq \frac{\rho - \rho A_3 - 1}{1 - \rho kA_3}, \tag{4}$$
2. if $1 - \rho kA_3 < 0$, for
$$c \geq \frac{\rho - \rho A_3 - 1}{1 - \rho kA_3}. \tag{5}$$
3. if $1 = \rho kA_3$, for any value of c if $A_3 \leq \frac{1}{1+k}$

Clearly, $R_2 \leq \rho$ if

$$\frac{1 + c}{1/k + c} \leq \rho$$

Since $\rho > 1$, then $R_2 \leq \rho$ if

$$c \geq \frac{\rho/k - 1}{1 - \rho}. \tag{6}$$

Assume $(1 - \rho kA_3) > 0$ (case 1). From inequalities 4 and 6, we may say that, for all c either $R_1 \leq \rho$ or $R_2 \leq \rho$ if

$$\frac{\rho - \rho A_3 - 1}{1 - \rho kA_3} \geq \frac{\rho/k - 1}{1 - \rho}$$

that is,

$$\rho \geq \frac{A_3k + A_3 + 1/k - 2}{2A_3 - 1} \tag{7}$$

The right-hand-side of Inequality 7 is a monotonically non-increasing function of A_3, therefore, since $0 < A_3 \leq 1/k$, its maximum value tends to $2 - \frac{1}{k}$ from above, when $A_3 \to 0$.

Let $(1 - \rho k A_3) < 0$ (case 2), then, Inequality 5 holds if $1 + 1/k \leq \rho$. Moreover, if $\rho k A_3 = 1$ (case 3), $R_1 \leq \rho$ if $\rho \geq 3/2$.

Clearly, $R_{kH} \leq \max\{2 - \frac{1}{k} + \epsilon, 1 + \frac{1}{k}, \frac{3}{2}\} < 2 - \frac{1}{k}$ for $k \geq 3$ (and $R_{kH} \leq \frac{3}{2}$ if $k = 2$).

Therefore, the competitive ratio of k-Harmonic algorithm in the case $m_1 = 0$ is

$$R_{kH} < 2 - \frac{1}{k}.$$

(if $k \geq 3$).

Let us consider the case $m_1 \neq 0$. Let n_1, n_2 and n_3 be the number of items in the closed, filled and temporary bins, respectively.

Case A. If $n/k \geq \sum a_i$ then, since we had

$$\frac{n_3}{k} + m_2 \leq \left(2 - \frac{1}{k}\right) \frac{n_3 + n_2}{k}$$

for $m_1 = 0$, we obtain

$$\frac{n_1}{k} + \frac{n_3}{k} + m_2 \leq \left(2 - \frac{1}{k}\right) \frac{n_3 + n_2}{k} + \frac{n_1}{k} \leq \left(2 - \frac{1}{k}\right) \frac{n_3 + n_2 + n_1}{k}$$

and, since $\frac{n_3+n_2+n_1}{k}$ is clearly a lower bound of the optimum, $R_{kH} \leq 2 - \frac{1}{k}$.

Case B. If $\sum_i a_i > n/k$, then observe that the algorithm has ratio $2 - \frac{1}{k}$ if only filled and temporary bins are considered and, since the number of closed bins returned by k-Harmonic is at most twice their number in the optimum solution, we have $R_{kH} < 2$.

In the case $k = 2$ the competitive ratio of algorithm k-Harmonic is $R = 5/4$. Assume $m_1 = 0$. In filled bins there is only one item and it is of size greater than $1/2$, while in temporary bins there are exactly two items of size smaller than $1/2$. If $m_1 = 0$, then it means that the items in temporary bins do not fit in any filled bin. Therefore, the optimum, in this case, cannot do better than algorithm k-Harmonic. If $m_2 = 0$ or $m_3 = 0$, then k-Harmonic finds the optimal solution.

Assume, therefore, that m_1, m_2, and $m_3 \neq 0$. The only way to improve the result obtained by k-Harmonic is to eliminate all temporary bins, i.e. put all the items in temporary bins in closed bins and the I_k items in closed bins in filled bins. Therefore, the worst case is when $m_1 = m_2$ and $m_3 = 1/2 m_2$. In this case, the optimum cannot do better than $m_1 + m_2$, therefore $R_{kH} = 5/4$.

In the case $k = 2$, there exists a sequence of tasks such that k-Harmonic, when applied to such a sequence, has competitive ratio $R = 5/4$. Consider the instance, where the following items arrive in the given order and N is even.

1. N items of size $\frac{1}{2} + \epsilon$, with $0 < \epsilon < \frac{1}{2}$;
2. N items of size ϵ;
3. N items of size $1 - \epsilon$;
4. N items of size a, with $\epsilon < a < \frac{1}{2} - \epsilon$.

In this case, algorithm k-Harmonic uses $m_1 + m_2 + m_3 = \frac{5}{2}N$ bins ($m_1 = m_2 = 2m_3 = N$), where the m_1 closed bins have one item of size $\frac{1}{2} + \epsilon$ and one of size ϵ, the m_2 filled bins have only one item of size $1 - \epsilon$ and the m_3 temporary bins are each with two items of size a. In this case, the optimal algorithm uses $2N$ bins by packing in N bins one item of size $\frac{1}{2} + \epsilon$ and one of size a and in the remaining N bins one of size $1 - \epsilon$ and one of size ϵ.

4 Conclusions

In this paper we propose a (semi) on-line algorithm for bin packing with cardinality constraints. We remark that this is the first algorithm for such a problem in an on-line setting. We prove that $HARMONIC_M$, when applied to bin packing with cardinality constraint, has a competitive ratio at least $5/2$ for $k > 4$, $7/3$ for $k \geq 7$, and $8/3$ for $k \geq 20$. The algorithm we proposed has running time $O(n \log n)$, competitive ratio $2 - \epsilon$ in the general case, and $5/4$ in the case $k = 2$.

Interesting open problems on the subject include the improvement of the algorithm analysis in terms of both upper and lower bounds on the competitive ratio. An upper bound of $2 - 1/k$, in our opinion, might be reachable. A general lower bound on the competitive ratio (for any k) could also be derived.

References

[1] L. Babel, H. Kellerer (1998) *An On-Line Algorithm for Cardinality Constrained Bon Packing Problems*, Technical Report 06/1998, Faculty of Economics, University of Graz.

[2] A. Borodin, N. Linia, M. Saks (1987) *An Optimal online algorithm for metrical tasks Systems*, in proceedings of the 19th ACM Symposium on Theory of Computing, pp.373-382.

[3] E. G. Coffman, M. R. Garey, D. S. Johnson (1997) *Approximation Algorithms for Bin Packing: A Survey*, in Approximation Algorithms for NP-hard Problems, ed. D. S. Hochbaum, pp.46-86.

[4] G. Gambosi, A. Postiglione, M. Talamo (1990) *New algorithms for on-line bin packing*, Proceedings 1st Italian Conference on Algorithms and Complexity.

[5] G. Gambosi, A. Postiglione, M. Talamo (1997) *On line maintenance of an Approximate Bin-Packing Solution*, Nordic Journal of Computing, 4, no. 2, pp. 151-166.

[6] M. R. Garey, R. L. Graham, D. S. Johnson, A. C. Yao (1976) *Resource constrained scheduling as generalized bin packing*, Journal of Combinatorial Theory Ser. A, 21, pp. 257-298.

[7] M. R. Garey, D. S. Johnson (1979) Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman.

[8] B. T. Han, G. Diehr, J. S. Cook (1994) *Multiple-type, two-dimensional bin packing problems: Applications and Algorithms*, Annals of Operations Research, 50, pp. 238-261.

[9] Z. Ivkovic and E. Lloyd (1998) *Fully dynamic algorithms for bin packing: Being (mostly) myopic helps*, SIAM Journal on Computing, 28(2), pp.574-611.

[10] H. Kellerer, U. Pferschy (1999) *Cardinality Constrained Bin-Packing Problems*, Annals of Operations Research, to appear.

[11] K. L. Krause, V. Y. Shen, H. D. Schwetman (1975) *Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems*, Journal of the ACM, 22, pp.522-550.

[12] K. L. Krause, V. Y. Shen, H. D. Schwetman (1977) *Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems, Errata*, Journal of the ACM, 24, pp. 527.

[13] C. C. Lee, D. T. Lee (1985) *A simple on-line bin-packing algorithm*, Journal of the ACM, 3, pp. 562-572.

[14] M. S. Manasse, L. A. McGeoch, D. D. Sleator (1990) *Competitive Algorithms for Server Problems*, Journal of Algorithms, 11, pp. 208-230.

[15] S. Martello, D. Vigo (1998) *Exact solutions of the two-dimensional finite bin packing problem*, Management Science, 44, no. 3, pp. 388-399.