New Lower Bound for the Bin Packing Problem

Bassem Jarboui
1 FSEGS, route de l'aéroport km 4 Sfax 3018, Tunisie, bassem_jarboui@yahoo.fr
Abdelwaheb Rebai
ISAS, route de l'aéroport km 4 B.P.N° 1013 Sfax 3018, Tunisie, abdelwaheb.rebai@fsegs.rnu.tn

In this paper, we present new classes of lower bound for the one-dimensional bin packing problem, we prove that the worst-case asymptotic performance ratio of this bound is 3/4. Extensive numerical experiments show that this new bound outperforms the best lower bounds from the literature.

Keywords: bin packing, lower bound.

1. Introduction

The bin packing problem can be defined as follows: Given a set L of n items i, with sizes w_i, for any ($1 \leq i \leq n$) and a set of identical bins with capacity C, with an objective to determine the minimal number of bins necessary to pack all the items, so that the sum of the volumes of the items in the same bin does not exceed the capacity of the bin. The problem has many real-world applications (for example in cutting stock or when loading trucks subject to weight limitations). This problem is known to be NP-hard in the strong sense see Garey and Johnson [11]. Consequently, various researches on this problem have mainly focused on the development of polynomial time approximation algorithms. For an exhaustive survey of this area, the reader might refer to Coffman et al [4]. Only a few papers have been devoted to the development of exact algorithms. This work includes the branch and bound algorithm of Martello and Toth [12] and Scholl et al. [14]. In order to get a high performance of such algorithm, it is desirable to have fast lower bounds. There exist several lower bounds for the BPP.

Two types of lower bounds can be developed for the BPP: constructive and destructive bounds. A simple constructive bound is the sum of size of items divided by the size of bin; this bound is called LB_1. This approach also includes the lower bounds (LB_2 and LB_3) proposed by Martello and Toth [12] and the bound based on the Dual Feasible Functions developed by Fekete and Schepers [9]. Destructive bounds are derived in the following way. Starting with a hypothetical lower bound LB, an attempt is made to obtain contradictions to feasibility. If the method provides that there isn't a feasible solution with a number of bins smaller than or equal to LB, then LB+1 is a valid lower bound value. This approach was used by Alvim et al. [1] and Haouari and Gharbi [10].

The goal and the main contribution of this paper is to develop new constructive bound for the BPP. The performances of the new bound are analyzed, both analytically and computationally.

The following part of paper will focus on a review of the best existing lower bounds for the BPP in Section 2. In Section 3, we introduce a new class of bounds; we prove that the worst-case asymptotic performance ratio of the proposed bound is 3/4. In section 4, we give the computational results which illustrate the effectiveness of the proposed bound. Finally, Section 5 contains some concluding remarks.

2. Lower bounds

In this section we recall a review of some well-known linear-time lower bounds for the BPP.

Definition 2.1. Let L be an instance of the problem. LB a lower bound for a minimisation problem, and $OPT(L)$ denotes the optimum value of L. The absolute worst-case ratio and asymptotic worst-case ratio of $LB(L)$ are respectively defined by:

$$r(LB) = \sup\left\{ \frac{LB(L)}{OPT(L)} \middle| \; \forall L \right\},\; r_\infty(LB) = \lim_{s\to\infty} \sup\left\{ \frac{LB(L)}{OPT(L)} \middle| \; \forall L \text{ with } OPT(L) \geq s \right\}.$$

Let $L=(x_1,\ldots.., x_n)$ be a BPP instance. A simple bound for the BPP is $LB_1(L)$, this bound is obtained by relaxing the constraint witch shows that the items are indivisible:

$$LB_1(L)=\left\lceil \sum_{j=1}^{n} w_j / c \right\rceil$$

The bound $LB_1(L)$ can be computed in $O(n)$ time and it has a worst case performance ratio of 1/2.
A better bound was introduced by Martello and Toth [12] by partitioning the set of items into three subsets $L_1(\alpha)$, $L_2(\alpha)$ and $L_3(\alpha)$ based on the parameter $\alpha \in [0, C/2]$, as follows .

$$L_1(\alpha)=\left\{ j \in L; w_j > C-\alpha \right\}$$

$$L_2(\alpha)=\left\{ j \in L; C-\alpha \ge w_j > c/2 \right\}$$

$$L_3(\alpha)=\left\{ j \in L; C/2 \ge w_j \ge \alpha \right\}$$

$LB^{MT}(\alpha ,L)$ presents the number of bins necessary to pack the three subsets of items:

$$LB^{MT}(\alpha,L)= \left|L_1(\alpha)\right| + \left|L_2(\alpha)\right| + \max\left\{0, \left\lceil \frac{\sum_{j\in L_3(\alpha)} w_j - \left(\left|L_2(\alpha)\right|.C - \sum_{j\in L_2(\alpha)} w_j\right)}{C} \right\rceil \right\}$$

The bound $LB_2(L)$ is obtained from the maximal value of $LB^{MT}(\alpha ,L)$ for $\alpha \in [0, C/2]$.

$$LB_2(L)= \max_{\alpha\in[0,C/2]} LB^{MT}(\alpha,L)$$

Martello and Toth [12] have proved that this bound has an asymptotic worst-case ratio 2 /3 and it can be computed in $O(n)$ time if the items are sorted by non-increasing volumes.
Fekete and Schepers [9] propose a new approach based on dual feasible function, to obtaining a new lower bound for bin packing problem. A function u: $[0,1]\rightarrow[0,1]$ is called dual feasible, if for any finite set S of non-negative real numbers, we have the relation $\sum_{x\in S} x \le 1 \Rightarrow \sum_{x\in S} u(x) \le 1$.
Let u be a dual feasible function and let $L=(x_1,\ldots.., x_n)$ be a BPP instance. The lower bound obtained from the transformed BPP instance $u(L)=(u(x_1),\ldots.., u(x_n))$ is also a lower bound for L.
The new lower bounds $LB_*^{(p)}(L)$ can be obtained by applying the same lower bound on the transformed BPP instance $u^{(k)}(L)$ where :

$$LB_*^{(p)}(L) = \max\left\{ LB_2(L), \max_{k=2,\ldots,p} LB_2^{(k)}(L) \right\}$$

$$u^{(k)}(x_1) = \begin{cases} x & for\ x(k+1) \in \square \\ \lfloor (k+1)x \rfloor \dfrac{1}{k} & otherwise \end{cases}$$

$$LB_2^{(k)}(L) = LB_2(u^{(k)}(L))$$

Fekete and Schepers [9] have proved that $LB_*^{(p)}(L)$ has an asymptotic worst-case performance of 3/4 and can be computed in $O(n)$ time if the items are sorted by non-increasing volumes.
Alvim et al. [1] propose the bound $LB_\vartheta(L)$ inspired from the work of Dell'Amico and Martello [7]. This bound can be presented as follows:

Let $\theta = \max_{q=1,\ldots..,n} \left\{ q; \quad \sum_{i=n-q+1}^{n} w_i \le C \right\}$

be an upper bound to the number of items per bin for any feasible solution with LB bins.

Theorem 2.1.

$$\vartheta = \max\left\{ \max_{\sigma=1,\ldots,\lfloor n/LB \rfloor} \left\{ \sigma : \left\lceil \sum_{i=\sigma}^{n} \frac{w_i}{LB-1} \right\rceil > C \right\}, \quad \max_{\sigma=1,\ldots,\lfloor n/LB \rfloor} \left\{ \sigma : \sum_{i=1}^{\sigma} w_i \le C \right\} \right\}$$

is a lower bound to the number of items per bin for any feasible solution with LB bins.

Theorem 2.2. Let LB be a lower bound to the number of bins.

$$LB_\vartheta(L)\begin{cases} LB+1, & if\ (a),(b), or\ (c)\ is\ verified \\ (a) & \theta < \lceil n/LB \rceil \\ (b) & \left\lceil \sum_{i=LB_\vartheta.\vartheta+1}^{n} \dfrac{w_i}{LB-LB_\vartheta} \right\rceil > C \\ (c) & \theta = \vartheta + 1\ et\ C_\vartheta(LB) > C; \\ LB, & otherwise \end{cases}$$

LB_ϑ is an improved lower bound to BPP, where

$$C_\vartheta(LB) = \max\left\{ \left\lceil \sum_{i=n-\vartheta.LB_\vartheta+1}^{n} w_i / LB_\vartheta \right\rceil, \left\lceil \sum_{i=n-\theta.LB_\theta+1}^{n} w_i / LB_\theta \right\rceil \right\} \text{ and } LB_\vartheta = \max\{LB-(n-\vartheta.LB), 0\}$$

3. The new lower bound

In this section we present a new lower bound called $LB_2^\theta(L)$

Proposition 3.3. Let L be an instance of the BP problem and $1 \le k \le \theta$, $k \in$, then $\psi(k)$ is an upper bound to the number of bins containing a number of items larger than or equal to k.

$$\psi(k) = \min\left\{ \left\lfloor \frac{n}{k} \right\rfloor ; \max_{q=0,\dots,n-k} \left\{ q; \sum_{i=n-q-k+1}^{n-q} w_i \le C \right\} \right\}$$

Proof. If $\sum_{i=n-q-k+1}^{n-q} w_i > C$, then any item with a size larger than or equal to w_{n-q} cannot be packed into a bin containing k items only if this bin contains at least one item with a size smaller than w_{n-q}.

Thus, for any feasible solution of L

$$\max_{q=0,\dots,n} \left\{ q; \sum_{i=n-q-k+1}^{n-q} w_i \le C \right\}$$

is an upper bound to the number of bins containing a number of items larger than or equal to k.

Let $k_i \ge k$, the maximal number of bins having a number of items larger than or equal to k is bounded by $\max_{q=0,\dots,n} \left\{ q; \sum_{i=1}^{k} k_i \le n \right\} \le \max_{q=0,\dots,n} \left\{ q; \sum_{i=1}^{k} k \le n \right\} = \left\lfloor \frac{n}{k} \right\rfloor$

Then,

$$\psi(k) = \min\left\{ \left\lfloor \frac{n}{k} \right\rfloor ; \max_{q=0,\dots,n-k} \left\{ q; \sum_{i=n-q-k+1}^{n-q} w_i \le C \right\} \right\}$$

Theorem 3.4. Let L be an instance of the BP problem and θ the maximal number of items per bin, then

$LB^\theta(L) = \psi(k) + \left\lceil \dfrac{n-\theta.\psi(k)}{\theta-1} \right\rceil$ is a lower bound to the optimum value of L.

Proof. Let m be the number of bins containing a number of items equal to θ in the optimal solution of L and m' the remaining bins of this solution.

We have $m' \ge \left\lceil \dfrac{n-\theta.m}{\theta-1} \right\rceil$.

If $\psi(k) = m$ then $m + m' \le \psi(k) + \left\lceil \frac{n - \theta.\psi(k)}{\theta - 1} \right\rceil$

If $\psi(k) > m$

Let

$$\psi(k) + \frac{n - \theta.\psi(k)}{\theta - 1} = \frac{\theta.\psi(k) - \psi(k) + n - \theta.\psi(k)}{\theta - 1}$$

$$= \frac{n - \psi(k)}{\theta - 1} < \frac{n - m}{\theta - 1} = m + \frac{n - \theta.m}{\theta - 1} \le m + \left\lceil \frac{n - \theta.m}{\theta - 1} \right\rceil$$

We have $\psi(\theta) + \frac{n - \theta.\psi(k)}{\theta - 1} < m + m'$ then $\psi(k) + \left\lceil \frac{n - \theta.\psi(k)}{\theta - 1} \right\rceil \le m + m'$

Proposition 3.5. The computation of $LB^{\theta}(L)$ requires $O(n)$ time for pre-sorted items.

Proof. Let $q^* = \max_{q=1,\dots,n-\theta+1} \left\{ q; \sum_{i=n-q-\theta+2}^{n-q+1} w_i \le C \right\}$

For any value of θ, it is easy to say that if θ is valid then $q^* \ge 1$, therefore to compute q^*, we can start from q=2. We have $\sum_{i=n-q-\theta+1}^{n-q} w_i = \sum_{i=n-q-\theta+2}^{n-q+1} w_i + w_{n-q-\theta+1} - w_{n-q+1}$, these imply that the computation of q^* requires a constant number multiplied by q^*. We have $\theta \le n$ and $q^* \le n$, which makes the computation of q^* require $O(n)$ time for computing θ and $O(n)$ time for computing q^*. Then $\psi(\theta)$ is computed in $O(n)$ time.

So, $LB^{\theta}(L) = \psi(\theta) + \left\lceil \frac{n - \theta.\psi(\theta)}{\theta - 1} \right\rceil$ can be computed in $O(n)$ time.

In what follows, we present a refinement of the above lower bound $LB^{\theta}(L)$. Similar to the bound $LB_2(L)$, our bound is based on a partition of the set of items into two subsets $L_1(\alpha)$ and $L_4(\alpha)$, where

$$L_1(\alpha) = \left\{ j \in L; w_j > C - \alpha \right\}$$

$$L_4(\alpha) = \left\{ j \in L; C - \alpha \ge w_j > \alpha \right\}$$

We define $LB_{MT}^{\theta}(\alpha, L)$ by

$$LB_{MT}^{\theta}(\alpha, L) = \left| L_1(\alpha) \right| + \max\left\{ LB_1\left(L_4(\alpha)\right), LB^{\theta}\left(L_4(\alpha)\right) \right\}$$

Then a valid lower bound is

$$LB_2^{\theta}(L) = \max_{\alpha \in [0, C/2]} LB_{MT}^{\theta}(\alpha, L)$$

Proposition 3.6. The computation of $LB_2^{\theta}(L)$ requires $O(n)$ time for pre-sorted items.

Proof. Martello and Toth[12] have proved that all the updating of $LB_1(L_4)$ can be computed in $O(n)$ time, then it suffices to prove that all the updating of $LB^{\theta}(L)$ can be computed in $O(n)$ time. To compute $LB_2^{\theta}(L, \alpha)$, we consider the α values which are limited by $\alpha_i = \left\{ w_i \in L : w_i \le C/2 \right\}$.

Let $\theta(\alpha_i)$ be the value of the maximal number of items per bin associated to α_i, θ_l the set of values of the maximal number of items per bin obtained during the computation of $LB_2^{\theta}(L, \alpha)$ with $l = 1, \dots, r$ and $\theta_l \ge \dots \ge \theta_l \ge \theta_{l+1} \ge \dots \ge \theta_r$.

Let n_l be the number of values taken by α_i decremented by 1, obtained directly after the passage from θ_{l+1} to θ_l

and $q_l^* = \max_{q=0,\dots,n-n_l-\theta_l} \left\{ q; \sum_{j=n-n_l-q-\theta_l+1}^{n-n_l-q} w_j \le C \right\}$

Computation of all the updating of maximal number of items per bin associated to α_i
Let $\theta(\alpha_i) = \theta_l$,
If $i - n_l \le q_l^*$
then $\theta(\alpha_i) = \theta(\alpha_{i-1})$
otherwise we have to compute the new value of $\theta(\alpha_i)$
if $q_l^* \ge \theta_l$,
the computation of θ_{l+1} would begin from the item with index $\sum_{i=1}^{l} q_i^*$ then the total number of operations is equal to $\sum_{l=1}^{r} \theta_l \le \sum_{l=1}^{r} q_l^*$, it is clear that $\sum_{l=1}^{r} q_l^* \le n$, we then have $\sum_{l=1}^{r} \theta_l \le n$
if $q_l^* < \theta_l$ and $\theta_{l+1} > \theta_l - q_l^*$,
then the computation of the value θ_{l+1} requires a subtraction of the items going from $w_{n-n_l-q_l^*+1}$ through w_{n-n_l} and the addition of items which goes from $w_{n-n_l-\theta_{l+1}-q_l^*+1}$ through $w_{n-n_l-\theta_l}$ thus, the total number of operations will be equal to :

$$\theta_1 + \sum_{l=1}^{r-1} (n-n_l) - (n-n_l-q_l^*+1) + (n-n_l-\theta_l) - (n-n_l-\theta_{l+1}-q_l^*+1)$$

$$= \theta_1 + \sum_{l=1}^{r-1} (2q_l^* + \theta_{l+1} - \theta_l - 2) = \theta_1 + \sum_{l=1}^{r-1} 2q_l^* - 2(r-1) \le 3n - 2(r-1)$$

if $q_l^* < \theta_l$ and $\theta_{l+1} = \theta_l - q_l^*$,
then the computation of the value θ_{l+1} requires a subtraction of the items going from $w_{n-n_l-q_l^*+1}$ to w_{n-n_l} and the addition of items which goes from $w_{n-n_l-\theta_{l+1}-q_l^*+1}$ through $w_{n-n_l-\theta_l}$.This implies that the total number of operations will be equal to :

$$\theta_1 + \sum_{l=1}^{r-1} \left[(n-n_l) - (n-n_l-q_l^*+1) \right] = \theta_1 + \sum_{l=1}^{r-1} q_l^* - (r-1) \le 2n - (r-1)$$

Computation of maximal number of bins for any value of α_i
We have

$$\psi(\theta(\alpha_i)) = \min\left\{ \left\lfloor \frac{|L_4|}{\theta(\alpha_i)} \right\rfloor ; \max_{q=0,\dots,n-i-\theta(\alpha_i)} \left\{ q; \sum_{i=n-i-q-\theta(\alpha_i)+1}^{n-i-q} w_i \le C \right\} \right\} \text{ And } \theta(\alpha_i) = \theta_l$$

If $\theta(\alpha_i) = \theta(\alpha_{i-1})$
then $\max_{q=0,\dots,n-i-\theta(\alpha_i)} \left\{ q; \sum_{i=n-j-q-\theta(\alpha_i)+1}^{n-i-q} w_j \le C \right\} = q_l^* - (i - n_l)$
this implies that, $\psi(\theta(\alpha_i))$ can be computed in linear time.
if $\theta(\alpha_i) < \theta(\alpha_{i-1})$,
then, to compute q_l^*, we consider the set of items used for computing θ_l and we start with $q_l^* = 2$. We have proved in Proposition 3.3 that the computation of q_l^* requires a constant number multiplied by q_l^*, we have $\sum_{l=1}^{r} q_l^* \le n$, these imply that $\psi(\theta)$ can be computed in a linear time for all the updating. Then, the computation of $LB_2^{\theta}(L)$ requires $O(n)$ time for pre-sorted items.

Theorem 3.7. Let L be an instance of the BP problem with the size of items larger than $C/3$. Then, $LB_2^{\theta}(L) = OPT(L)$

Proof.

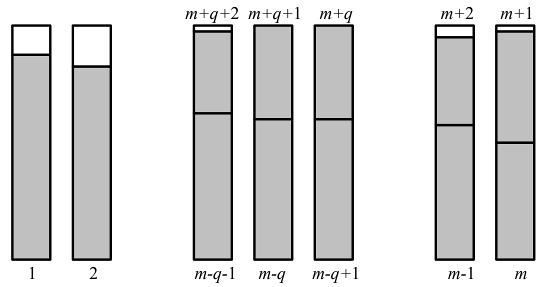

Figure 1: Normal form of an optimal solution for a BPP instance with sizes $w_i > C/3$

Let $m = OPT(L)$ be the number of bins of optimal solution and $\theta = 2$ the maximal number of items per bin.

If $m = \frac{n}{2}$,

then for any value of $\alpha = 0$, we have

$$LB_2^{\theta}(L) \geq |L_1| + \left\lceil \frac{|L_4|}{\theta} \right\rceil = 0 + \frac{n}{2} = m = OPT(L) .$$

If $m > \frac{n}{2}$,

then, there exists at least one item w_{m+q+1} which cannot be packed with any other item having a size larger than $w_{m-q} > C/2$ for any feasible solution with m bins. Then, for any value of $\alpha = C - w_{m-q}$, we have

$$LB_2^{\theta} \geq |L_1| + \left\lceil \frac{|L_4|}{\theta} \right\rceil \geq m - q - 1 + \frac{((m+q+1)-(m-q)+1)}{2} = m = OPT(L)$$

Then, LB_2^{θ} is optimal for any solution with the items' sizes larger than $C/3$

Theorem 3.8.

$$r_{\infty}(LB_2^{\theta}) = \frac{3}{4}$$

Proof. Let $L' \in L$ be the sub-set of items with a size $w_i > C/k$. Crainic and al.[5] have proved that for a given value of $k > 1, k \in$ □ and for a lower bound LB of the BPP, if $LB(L') = OPT(L')$, $LB(L) \geq LB(L')$ and $LB(L) \geq LB_1(L)$, the asymptotic worst-case ratio of LB is

$$r_{\infty}(LB) = \frac{k}{k+1} .$$

Then, it suffices to prove that $LB_2^{\theta}(L) \geq LB_2^{\theta}(L')$.

We have proved in Theorem 3.7 that the optimal solution of any instance with the size of items larger than $C/3$ can be reached from the value of $\alpha = C - w_{m-q} > C/3$.

For any value of $\alpha > C/3$, we have $LB_{MT}^{\theta}(\alpha, L) = LB_{MT}^{\theta}(\alpha, L')$ then,

$LB_2^{\theta}(L) \geq OPT(L') = LB_2^{\theta}(L')$

Thus, $r_{\infty}(LB_2^{\theta}) = \frac{3}{4}$.

4. Computational results

A computational experiment was conducted to evaluate the performance of the proposed new lower bounds. The algorithms were coded in C++, and the computational experiments were run on a PC Pentium IV, 3.2 GHz Personal Computer with 512 MB RAM.

The class of problems is generated in a similar way to the instances described by Fekete and Schepers [9]. For a given number n of items, the sizes were generated randomly with uniform distribution on the sets S1 ={1,…,100}, S2 ={1,…,90}, S3 ={1,…,80} , S4 ={20,…,80} , S5 ={20,…,70}, for the container size C=100.

We consider instances with 32, 100, 316, and 1000 items. One thousand random instances are generated for each problem class and number of items.

We measure the quality of bounds according to the number of improved instances and the total gaps in the number of bins. Let "imp" and "gap" be respectively the number of instances improved by *LB* from *LB'* (*LB*/*LB'*) and the total gaps obtained by *LB*-*LB'*.

We define a new bound by exploiting the dual function proposed by Fekete and Schepers [9] which is as follows:

$$LB_{FS}^{\theta}(L,p) = \max\left\{LB_2^{\theta}(L), \max_{k\in[0,p]}\left\{LB_2^{\theta}(u^{(k)})\right\}\right\}.$$

Table 1: number of improvements and total gaps in the number of bins for 1000 instances

		$LB_2^{\theta}(L)/LB_2(L)$		$LB_{FS}^{\theta}(L,20)/LB_*^{(20)}(L)$		$LB_{FS}^{\theta}(L,50)/LB_*^{(50)}(L)$		$LB_{FS}^{\theta}(L,100)/LB_*^{(100)}(L)$	
		imp	gap	imp	Gap	imp	gap	imp	gap
1 100	32	90	90	169	169	169	169	167	167
	100	53	54	111	112	101	102	99	100
	316	70	86	151	176	112	141	110	139
	1000	183	243	356	755	109	156	126	173
1 90	32	96	96	192	192	189	189	185	185
	100	59	64	115	115	100	101	99	100
	316	54	68	155	187	114	149	117	152
	1000	121	184	294	612	113	192	115	194
1 80	32	65	65	153	153	151	151	151	151
	100	33	39	53	53	68	69	66	67
	316	9	18	19	23	35	47	36	48
	1000	0	0	0	0	19	22	18	21
20 80	32	67	67	153	153	158	158	154	154
	100	54	59	116	117	108	115	105	112
	316	113	147	252	370	146	199	151	204
	1000	221	373	492	1437	104	159	114	169
20 70	32	54	54	67	67	73	74	73	74
	100	12	16	14	14	20	22	19	21
	316	1	1	1	2	1	3	2	4
	1000	0	0	0	0	0	0	0	0

Table 1 shows the results obtained by applying the bounds $LB_2^{\theta}(L)$, $LB_{FS}^{\theta}(L,20)$, $LB_{FS}^{\theta}(L,50)$ and $LB_{FS}^{\theta}(L,100)$ on the class of problems. We note that these bounds outperform respectively the bounds $LB_2(L)$, $LB_*^{(20)}(L)$, $LB_*^{(50)}(L)$ and $LB_*^{(100)}(L)$. Table 2 reports the average CPU time (in 10^{-3} s) necessary to compute the different bounds. We note that $LB_{\varphi}(L)$ is a very fast bound which provides high quality bounds under an acceptable computational effort.

Table 2. Average CPU time (in 10^{-3} s)

	LB_2	$LB_*^{(20)}$	$LB_*^{(50)}$	$LB_*^{(100)}$	$LB_2^\theta(L)$	$LB_{FS}^\theta(20)$	$LB_{FS}^\theta(50)$	$LB_{FS}^\theta(100)$
32	0.0231	0.761	1,3282	4.3246	0,0468	1,2134	3,3158	6,7622
100	0,0372	1,8054	3,6428	9,4381	0,0776	2,9182	7,7444	17,3302
316	0,1412	4,9798	12,1334	25,9674	0,2564	9,1594	26,1022	50,553
1000	0,3941	20,5464	37,9156	123,495	0,7016	30,2448	77,0992	138,879

5. Conclusion

A new lower bound for the bin packing problem is presented. We have proved that this bound has an asymptotic worst case ratio of 3/4. The computational results have proved the effectiveness of this proposed bound to obtain the best results among existing lower bounds. The new bounds are easy to be implemented and often provide high quality bounds under an acceptable computational effort.

References

[1] A. Alvim, C. Ribeiro, F. Glover, D. Aloise (2004), A hybrid improvement heuristic for the one-dimensional bin packing problem, *Journal of Heuristics* **10**, 205 – 229.

[2] J.M. Bourjolly, V. Rebetez (2005), An analysis of lower bound procedures, *Computers and Operations Research* **32**, 395 – 405.

[3] B. Chen, B. Srivastava (1996), An improved lower bound for the bin packing problem, *Discrete Applied Mathematics* **66**, 81 – 94.

[4] E.G. Coffman Jr., M.R. Gary, D.S. Johnson (1997), *Approximation algorithms for bin packing*: *a survey*, *in*: *D. Hochbaum* (*Ed*.), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, Boston.

[5] G. Crainic, G. Perboli, M. Pezzuto, R. Tadei R, Computing the asymptotic worst-case of bin packing lower bounds. *European Journal of Operational Research*, (article in press).

[6] G. Crainic, G. Perboli, M. Pezzuto, R. Tadei R (2007), New bin packing fast lower bounds. *Computers and Operations Research* **34**, 3439 – 3457.

[7] M. Dell'Amico, S. Martello (1995), Optimal scheduling of tasks on identical parallel processors, *ORSA J. Comput* **7**, 191 – 200.

[8] S. Elhedhli (2005), Ranking lower bounds for the bin packing problem, *European Journal of Operational Research* **160,** 34 – 46.

[9] S. Fekete, J. Schepers (2001), New classes of fast lower bounds for bin packing problems, *Math. Programming* **91**, 11 – 31.

[10] M. Haouaria, A. Gharbia (2005), Fast lifting procedures for the bin packing problem, *Discrete Optimization* **2**, 201 – 218.

[11] M.R. Garey, D.S. Johnson (1979), *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco.

[12] S. Martello, P. Toth (1990), *Knapsack Problems: Algorithms and Computer Implementations*, Wiley, Chichester.

[13] S. Martello, P. Toth (1990), Lower bounds and reduction procedures for the bin packing problem, *Discrete Applied Mathematics* **28**, 59 – 70.

[14] A. Scholl, R. Klein, C. Jürgens (1997), BISON: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem, *Computers and Operations Research* **24**, 627–645.

[15] F. Vanderbeck (1999), Computational study of a column generation algorithm for bin-packing and cutting stock problems, *Mathematical Programming* **86**, 565 – 594.