An experimental comparison of some heuristics for cardinality constrained bin packing problem

Maja Remic
University of Ljubljana, Ljubljana, Slovenia

Gašper Žerovnik
Jožef Stefan Institute, Reactor Physics Division, Ljubljana, Slovenia

Janez Žerovnik
Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia,
and
Faculty for Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia

Abstract

Background: Bin packing is an NPhard optimization problem of packing items of given sizes into minimum number of capacitylimited bins. Besides the basic problem, numerous other variants of bin packing exist. The cardinality constrained bin packing adds an additional constraint that the number of items in a bin must not exceed a given limit Nmax. **Objectives:** Goal of the paper is to present a preliminary experimental study which demostrates adaptations of the new algorithms to the general cardinality constrained bin packing problem. **Methods/Approach:** Straightforward modifications of First Fit Decreasing (FFD), Refined First Fit (RFF) and the algorithm by Zhang et al. for the bin packing problem are compared to four cardinality constrained bin packing problem specific algorithms on random lists of items with 0%, 10%, 30% and 50% of large items. The behaviour of all algorithms when cardinality constraint Nmax increases is also studied. **Results:** Results show that all specific algorithms outperform the general algorithms on lists with low percentage of big items. **Conclusions:** One of the specific algorithms performs better or equally well even on lists with high percentage of big items and is therefore of significant interest. The behaviour when Nmax increases shows that specific algorithms can be used for solving the general bin packing problem as well.

Keywords: cardinality constrained bin packing problem heuristics, approximation algorithm.
JEL classification: C61
Paper type: Research article

Recieved: 28, February, 2012
Revised: 21, June, 2012
Accepted: 18, July, 2012

Citation: Remic, M., Žerovnik, G., Žerovnik, J. (2012). " An experimental comparison of some heuristics for cardinality constrained bin packing problem", Business Systems Research, Vol. 3, No. 2, pp. 57-63.
DOI: 10.2478/v10305-012-0013-1

Acknowledgements: The research was supported in part by the Slovenian Research Agency ARRS.

Introduction

The cardinality constrained bin packing problem can be described as follows: A list *I* of *n* items with specified size (or weight/volume/etc.) x_i has to be arranged into bins of limited capacity C_{max} and maximum number of items per bin (cardinality constraint) N_{max} to minimize *m*, the number of bins used. The minimization problem is known to be NP-hard, see for example (Žerovnik and Žerovnik, 2011). The problem was first studied by Krause, Shen and Schwetman (1975) when optimizing execution of tasks on a multiprocessor computer system. There are numerous applications of the cardinality constrained bin packing problem. The cardinality constrained bin packing problem can, for example, be applied to optimization of the spent nuclear fuel deposition in deep repository (Žerovnik et al., 2009). Swedish concept of deep repository in hard rock (Milnes, 2002) is currently seriously regarded in Slovenia as an option for nuclear power plant Krško decommissioning program (Železnik, et al., 2004). Motivated by this application, several heuristics for the cardinality constrained bin packing problem with N_{max} =4 were designed in Žerovnik and Žerovnik (2011). The new heuristics were compared against an obvious adaptation of the first fit decreasing (FFD) algorithm and were proven to clearly outperform the FFD algorithm on the datasets of interest. It is well known that FFD algorithm is a good approximation algorithm for the general bin packing problem (more precisely, it is well known (Korte and Vygen, 2000) that FFD always gives a solution with at most 11/9 OPT(I) + C bins, where OPT(I) stands for the value of the optimal solution). Therefore, it is natural to ask how the generalized versions of the new algorithms behave on the bin packing problem with arbitrary cardinality constraints. A preliminary experimental study is presented in this report which shows that the obvious adaptations of the new algorithms are competitive also on the general cardinality constrained bin packing problem.

Algorithm adjustments

In this paper, three algorithms (referred to as Alg1, Alg2, and Alg3) from Žerovnik and Žerovnik (2011) (see Appendix) are experimentally compared to First Fit Decreasing (FFD) (Coffman et al., 1997), Refined First Fit (RFF) (Yao, 1980), algorithm of Zhang, Cai and Wang (2000) (ZCW) and algorithm of Kellerer and Pferschy (1999) (KP). Since the FFD, RFF and ZCW are designed for the original bin packing problem, the following obvious modifications have been performed in order to adapt to the additional cardinality constraint:

- For any item, the FFD algorithm chooses the first bin with both enough space (capacity constraint) and at least one empty slot (cardinality constraint).
- Similarly, the RFF algorithm places each item in the first possible bin of the suitable group.
- ZCW algorithm closes the (active or additional) bin also when the cardinality constraint is reached.

All input data for all algorithms have been sorted by size, even though for RFF it is not necessary.

Experimental comparison of algorithms

First, the quality of solutions, obtained by different algorithms, was experimentally compared for different input data distributions. The main purpose of this investigation was to asset the performance of algorithms from Žerovnik and Žerovnik (2011), specifically designed for the cardinality constrained bin packing, relative to algorithm KP and the (suitably adapted) algorithms for the general bin packing problem. The instances for the experiments were generated regarding this goal, and were generated according to the probability distributions given below which may not be the usual distributions in bin packing literature. A similar experiment was already presented in Žerovnik and Žerovnik (2011), with two significant differences. In this paper, RFF, ZCW and KP algorithms were added for comparison. Furthermore, in the present experiments additional instances with different fractions of large items were regarded.

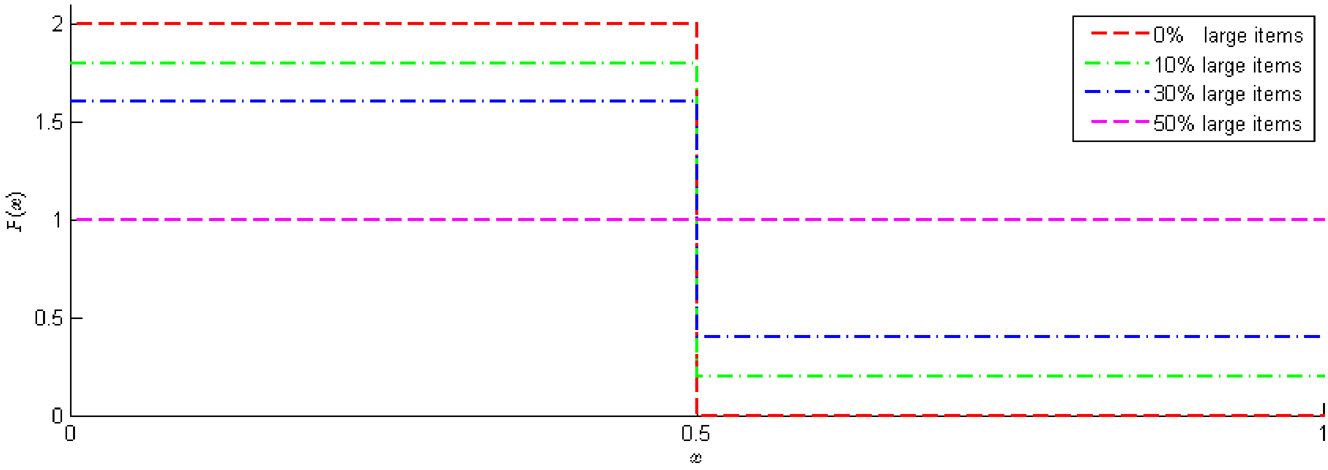
Experiments with fixed $N_{max} = 4$

Default values of $N_{max} = 4$ items per bin and capacity $C_{max} = 1$ were adopted. Four different types of input data of length $n = 100$ were used:

- lists without large items (items, larger than $C_{max}/2$),
- lists with 10% large items,
- lists with 30% large items, and
- lists with 50% large items.

All input lists were generated by default Octave random number generator, using uniform distribution with compositions given above. The input data distributions are shown on Fig. 1. The experiment has been repeated 10000 times with input data generated from different random seeds.

Figure 1
Distributions P(x) over the item size x for different types of input data.



The theoretical lower bound *MIN(I)* for the solutions of cardinality constrained bin packing problem can be expressed as:

$$MIN(I) = \max\left(\left\lceil \frac{\sum_{i=1}^{n} x_i}{C_{\max}} \right\rceil, \left\lceil \frac{n}{N_{\max}} \right\rceil, \ell\right) \tag{1}$$

where ℓ is the number of large items. This obvious bound is a useful and conservative approximation for the optimal solution $OPT(I) \geq MIN(I)$. The quality of the obtained solution *A(I)* is measured by $A(I)/MIN(I)$ and can be estimated by . The values for all algorithms, averaged over 10000 instances, are presented in Table 1.

Table 1
Average values of the *A(I)* / *MIN(I)* ratio for selected algorithms at different lists.

A(I) / MIN(I)	0% large items	10% large items	30% large items	50% large items
FFD	1.1275	1.0928	1.0350	1.0416
RFF	1.3317	1.3194	1.3253	1.3320
ZCW	1.3607	1.3530	1.2682	1.2579
KP	1.0415	1.0483	1.0635	1.0563
Alg1	1.0263	1.0696	1.1428	1.2056
Alg2	1.0494	1.0554	1.0503	1.0439
Alg3	1.0259	1.0573	1.0991	1.1430

The results clearly show the influence of the fraction of large items on the quality of the solution for individual algorithms. The ZCW and FFD algorithms work much better with significant fractions of large items, while the opposite can be stated for Alg1 and Alg3. For RFF, KP and Alg2, the sensitivity of the quality of solutions on large item fractions is insignificant.

Direct comparison between algorithms shows that ZCW and RFF in general give much worse solutions than FFD. For RFF that kind of behaviour is expected since it is (in contrast to other five algorithms) basically an online algorithm. Worse performance of ZCW algorithm may be due to linear time complexity. All three algorithms from Žerovnik and Žerovnik (2011) perform significantly better than FFD, RFF and ZCW for small (up to 10%) fractions of large items, whereas for 50% of large items, only Alg2 is comparable to FFD and KP. This is expected since Alg2 was designed for input data with significant fraction of large items while Alg1 and Alg3 were designed for input data with (almost) no large items. Let us mention at this point that Alg3 is random number based, therefore its solution may be improved significantly when taking advantage of multi-start mode (Žerovnik and Žerovnik, 2011).

Increasing the cardinality constraint

The purpose of the second experiment was to examine the behaviour of the algorithms when increasing the cardinality constraint Nmax. When Nmax is converged to the total number of items n, the cardinality constraint becomes irrelevant, consequently the cardinality constrained bin packing problem converges to the original bin packing problem in this limit.

In this experiment, list length n = 100 and capacity Cmax = 1 was used throughout. Two different types of input data lists were used: lists without large items and with 50% large items (Fig. 1), which were generated in exactly the same way as in the first experiment. The cardinality constraint was changed between Nmax = 2 and Nmax = n/4 since the latter was experimentally observed to be enough to achieve the convergence. The experiment was repeated 1000 times with different input data, each generated with different random number sequence. The results are shown in Figs. 2 and 3.

Figure 2
Average number of bins as a function of the cardinality constraint for lists without large items.

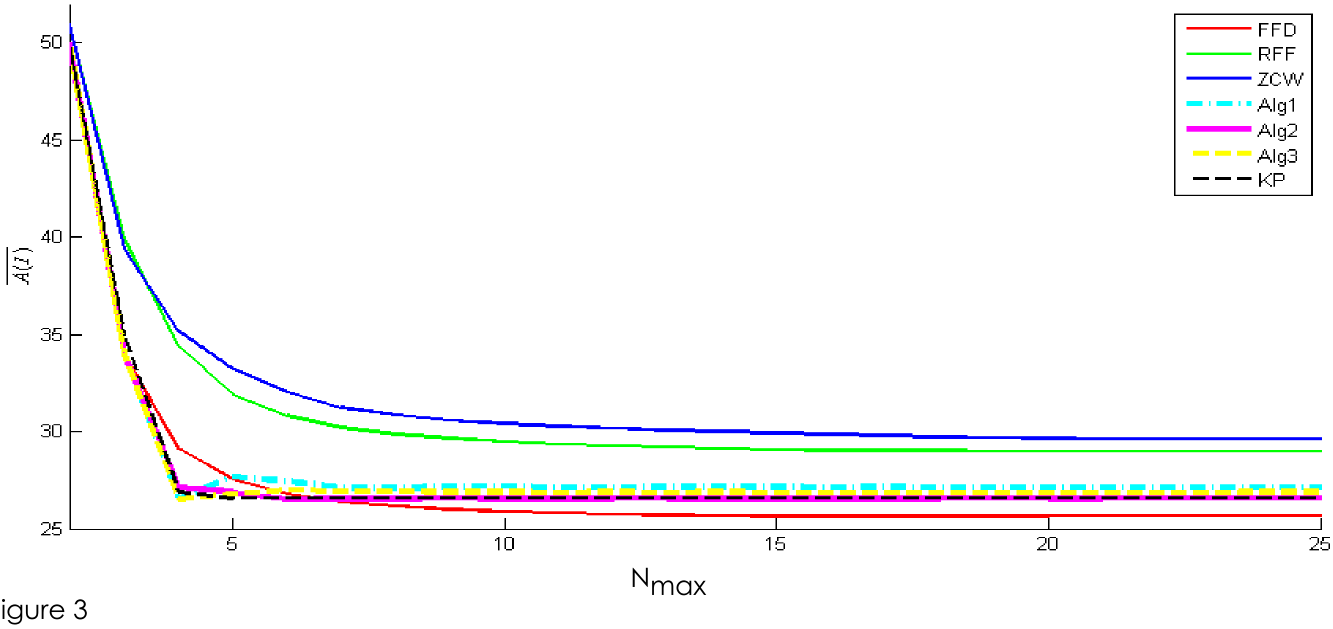
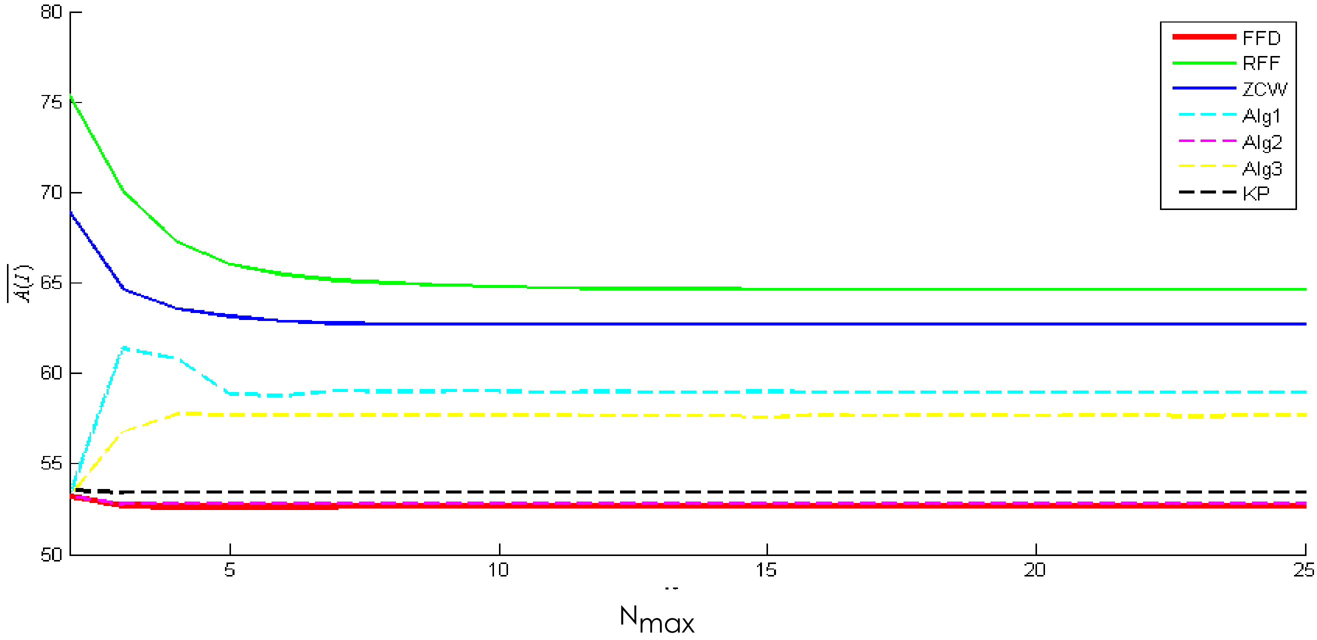


Figure 3
Average number of bins as a function of the cardinality constraint for lists with 50% of large items.



Expectedly, the number of bins used in most cases decreases with the increase of Nmax, up to the point where N_{max} no longer has impact and the capacity C_{max} of the bin becomes the limiting factor in Eq. (1). The speed of convergence depends on the percentage of large items in the list. When the percentage is high, the impact of N_{max} is smaller, meaning faster convergence. This is evident in lists with 50% large items (Fig. 3).

For lists without large items, all three algorithms for solving the cardinality constrained bin packing problem perform best at N_{max} = 4, which is the setting the algorithms were developed for, and deteriorate slightly as N_{max} increases. Similarly, for lists with 50% large items, both Alg1 and Alg3 experience unexpected swings at N_{max} < 5. This behaviour can be explained by the functioning of those two algorithms, since at small Nmax, larger items are considered first.

It is notable that the increasing of N_{max} has little influence over the final number of bins used by each algorithm. Even at the non-limiting N_{max} , the algorithms for solving the general bin packing problem find no better solutions than the specific algorithms, the FFD algorithm being the only exception and performing slightly better on lists without large items.

Discussion and conclusion

As the experiments have shown, all three (cardinality constrained bin packing problem) specific algorithms perform better compared to the general (bin packing) algorithms on lists without large items or with a low percentage of large items, while having a similar time complexity. In such cases, the use of specific algorithms is recommended. In the case of input data with significant fractions of large items, the general FFD algorithm perform best, closely followed by the specific Alg2. Furthermore, in any of the considered cases, at least one of the three proposed algorithms is comparable to the cardinality constraint KP algorithm. However, note that the KP algorithm has larger time complexity O(*n log2 n*) compared to O(*n log n*) complexity of Alg1-3. Alg3 has multi-start option enabling further improvements of the solution quality at the price of additional running time.

Finally, we wish to remark that all specific algorithms can be generalized to the basic bin packing problem by sufficiently increasing the cardinality constraint. Surprisingly or not, the quality of solutions is hardly compromised. The best performer is again Alg2, therefore it can be regarded as a good alternative to the FFD algorithm even for the basic bin packing problem.

References

1. Coffman, E. G, Garey, M. R, Johnson, D. S. (1997), „Approximation algorithms for binpacking: A survey“, in Hochbaum, D. (Ed.), Approximation Algorithms for NPHard Problems, PWS, Boston, pp. 4693.
2. Kellerer, H, Pferschy, U. (1999), „Cardinality constrained bin-packing problems“, Annals of Operations Research, Vol. 92, No. 0, pp. 335-348.
3. Korte, B, Vygen, J. (2000), Combinatorial Optimization, Theory and Algorithms, Berlin, Springer.
4. Krause, K. L, Shen, V. Y, Schwetman, H. D. (1975), „Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems“, Journal of ACM, Vol. 22, No. 4, pp. 522-550.
5. Milnes, A. G. (2002), "Guide to the documentation of 25 year of geoscientific research (1976-2000)", Technical Report, No. 02-18, Swedish Nuclear Fuel and Waste Management Co., Sweden, Stockholm, March 2002.
6. Yao, A. C. (1980), „New algorithms for bin packing“, Journal of the ACM, Vol. 27, No. 2, pp. 207-227.
7. Zhang, G, Cai, X, Wong, C. K. (2000), „Lineartimeapproximation algorithms for bin packing“, Operations Research Letters, Vol. 26, No. 5, pp. 217222.
8. Železnik, N, et al. (2004), "Program razgradnje NEK in odlaganja NSRAO in IJG" ("Program of NPP Krško Decommissioning and SF & LILW Disposal"), Study, No. T-1123-03, Agency for Radwaste Management, Ljubljana, Slovenia, and Agency for Hazardous Waste, Zagreb, Croatia.
9. Žerovnik, G, Snoj, L, Ravnik, M. (2009), „Optimization of Spent Nuclear Fuel Filling in Canisters for Deep Repository“, Nuclear Science and Engineering, Vol. 163, No. 2, pp. 183-190.
10. Žerovnik, G, Žerovnik, J. (2011), „Constructive heuristics for the canister filling problem“, Central European Journal of Operations Research, Vol. 19, No. 3, pp. 371-389.

Appendix

While FFD, RFF, KP and ZCW algorithms are well known, the Alg1, Alg2 and Alg3 first appear in a very recent publication, and therefore we give a brief outline here.

Generic Algorithm

```
  % $C_{max}$      .....   capacity of bins
  % K         .....   cardinality constraint
  % $R_i$        ....    the set of items assigned to the bin i
  % m,C,auxi  ....    auxiliary var then iables
quicksort X;
m := 0;
while $X \neq \emptyset$ do begin
     set aux := $\emptyset$; C := Cmax;
     for k: = K  down to 1 do begin
        x := CHOOSE(X);  (*)
        if x is defined  then X := X \{x}; C := C - x ; aux := aux $\cap$ {x}  endif;
    endfor;
    m := m + 1;
    $R_m$ = aux;
endwhile;
return R = {$R_1$,$R_2$, ... ,$R_m$}.
```

The algorithms differ only in implementation of the function CHOOSE.
CHOOSE in **Alg1**:
CHOOSE(X) := x, the largest element x of X that satisfies $x \leq C/k$;

CHOOSE in **Alg2**:
if k = K **then** CHOOSE(X) := x, the largest element of X that satisfies $x \leq C$;
if k < K **then** CHOOSE(X) := x, the largest element of X that satisfies $x \leq C/k$;

CHOOSE in **Alg3**:
if k > Floor(K/2) **then** CHOOSE(X) := x, a random element of X that satisfies $x \leq C$;
if $k \leq$ Floor(K/2) **then** CHOOSE(X) := x, the largest element of X that satisfies $x \leq C/k$;

Remark: Clearly if CHOOSE is simply
CHOOSE(X) := x, where x is the largest element of X that satisfies $x \leq C$;
we have the algorithm which puts each element in the first bin in which there is enough room. If K is large enough, this variant is equivalent to the FFD Algorithm.

About the authors

Maja Remic received the B.S. degree in computer and information science from University of Ljubljana in 2011. Currently, she is a software developer for a company in Ljubljana. Author can be contacted at ***maja.remic@siol.net***

Gašper Žerovnik received the B.S. degree in physics and Ph.D. degree in nuclear engineering from University of Ljubljana in 2007 and 2012. Currently he is research assistant at the Jožef Stefan Institute, Ljubljana, and part time teaching assistant at the Faculty of Energy Technology, University of Maribor. His main research interest is development and use of analytical and numerical methods, mainly in reactor physics with possible broader applications. Author can be contacted at ***gasper.zerovnik@ijs.si***

Janez Žerovnik received the B.S. and M.S. degree in mathematics from University of Ljubljana, Slovenia in 1982 and 1987. In 1992 he received Ph.D. degree in Computer Science from University of Ljubljana and in 1994 Ph.D. degree in mathematics from Technical University Graz, Austria. Currently, he is a Professor of mathematics at the Faculty of Mechanical Engineering at University of Ljubljana and part time researcher at the Institute of Mathematics, Physics and Mechanics. His research interests include discrete mathematics, in particular graph theory and its applications in computer science, operational research, mathematical chemistry, and elsewhere. Author can be contacted at ***janez.zerovnik@fs.uni-lj.si***