Approximation Algorithms for Extensible Bin Packing

E. G. Coffman, Jr.* George S. Lueker†

Abstract

In a variation of bin packing called *extensible bin packing*, the number of bins to use is specified as part of the input, and bins may be extended to hold more than the usual unit capacity. The cost of a bin is one if it is not extended, and the size if it is extended. The goal is to pack a set of items of given sizes with minimum cost. Adapting ideas in [6, 7, 3], we give a fully polynomial asymptotic approximation scheme for this extensible bin packing. We also show that the additive constant in the approximation bound cannot be eliminated unless $P = NP$.

1 Introduction

Consider the following variation of the bin-packing problem, which we call *extensible bin-packing*. We are given a list of n positive numbers $x_1, x_2, \ldots, x_n$, specifying the sizes of n different items, and a positive integer m giving a number of bins. We take the nominal capacity of each bin to be one. Let the bins be denoted by $b_1, b_2, \ldots, b_m$, and let the *level* of b_j, written $\ell(b_j)$, be the total size of the items packed into b_j. We will allow bins to be filled beyond their nominal capacity, and define the *cost* of a bin b_j to be $\max\bigl(1, \ell(b_j)\bigr)$. We are to assign each item to one of the bins, possibly filling some bins past their capacity, so as to minimize the total cost of the m bins, i.e., so as to minimize the objective function $\sum_{j=1}^{m} \max\bigl(1, \ell(b_j)\bigr)$.

In [2] it is shown that this problem is NP-hard, and that the LPT algorithm gives a 13/12 approximation.

If the number m of bins is fixed, then there is a simple dynamic programming solution in which the states correspond to m-tuples specifying the levels in each bin. Since this is *ex-benevolent* as defined in [8], it follows from the results of [8] that there is a fully polynomial time approximation scheme for the problem.

For the case in which m is varying, a polynomial time approximation scheme is presented in [1], as a special case of their more general theorem.

Here we show that the changes in this problem from the usual bin-packing problem do not prevent us from applying the methods of [6, 7, 3] to achieve a *fully* polynomial time asymptotic approximation scheme (FPTAAS) with varying m.

2 A Fully Polynomial Asymptotic Approximation Scheme

As in [2, 1], it is not hard to show that we can use reductions to make some simplifying assumptions. In particular, we can assume each item is of size at least $x_i \geq \frac{\epsilon}{1+\epsilon}$ and size at most 1, and that the sum of the item sizes is at most $2m$, without significantly affecting the difficulty of the problem. Under these assumptions one can easily show that there is always an optimum packing that does not pack any bin to a level of 3 or more.

We begin by rounding the items to sizes of the form $\lfloor \frac{(1+\epsilon)^k}{\epsilon} \rfloor \epsilon^2$. After rounding, let there be n_k items of size s_k, $1 \leq k \leq N$. Make a list of all possible ways of packing a bin to a level of at most about 3 with sizes chosen from the s_k; we call each of these ways of packing a bin a *configuration*. Let C_{ik} be the number of items of size s_k in the ith configuration, $1 \leq i \leq M$. Note that the cost of packing a bin according to the ith configuration can be written as $\max\bigl(1, \sum_{k=1}^{N} C_{ik} s_k\bigr)$, so we have the integer program

$$\text{(2.1)} \qquad \text{minimize} \quad \sum_{i=1}^{M} z_i \max\Bigl(1, \sum_{k=1}^{N} C_{ik} s_k\Bigr)$$

$$\text{subject to} \quad \sum_{i=1}^{M} z_i C_{ik} \geq n_k, \quad \text{for} \quad k = 1, 2, \ldots, N$$

$$\text{(2.2)} \qquad \text{and} \quad \sum_{i=1}^{M} z_i \geq m,$$

where the z_i are nonnegative integers telling us the number of instances of the ith configuration to be used. (Note that the differences introduced by considering extensible bin packing instead of ordinary bin packing are that the objective function changes from simply $\sum_{i=1}^{M} z_i$ to that shown in (2.1), and the extra constraint (2.2) is introduced.)

As usual, relax this to a linear program and take the dual. Let $\mathcal{C}$ be the set of vectors given by the rows of matrix C, so that each element of $\mathcal{C}$ specifies a configuration, $\vec{s} = (s_1, s_2, \ldots, s_N)$, and $\vec{n} = (n_1, n_2, \ldots, n_N)$.

*Columbia University.

†University of California, Irvine.

Then we can write the dual program as

$$\text{maximize} \ \ tm + \vec{n}\cdot\vec{u}$$

$$\text{subject to} \ \ t + \vec{m}\cdot\vec{u} \le \max(1, \vec{m}\cdot\vec{s}) \ \ \text{for} \ \ \vec{m}\in\mathcal{C}$$

$$\text{and} \ \ t, u_i \ge 0.$$

(For the ordinary bin packing problem we would have simply "*maximize* $\vec{n}\cdot\vec{u}$*, subject to* $\vec{m}\cdot\vec{u}\le 1$ *for* $\vec{m}\in\mathcal{C}$ *and* $u_i \ge 0$.")

The corresponding separation problem "*given* $(t, u_1, u_2, \ldots, u_N)$*, either determine that this vector is feasible, or find a vector* $\vec{c}$ *such that for all feasible* $(t', u'_0, u'_1, \ldots, u'_N)$*, we have* $(t, u_1, u_2, \ldots, u_N)\cdot\vec{c} > (t', u'_0, u'_1, \ldots, u'_N)\cdot\vec{c}$," is somewhat more complicated than the knapsack problem we obtain for ordinary bin packing. Nonetheless, it can again be solved using dynamic programming, by developing a table giving, for each possible value of l, the maximum value of $\vec{m}\cdot\vec{u}$ subject to the constraint that $\vec{m}\cdot\vec{s} = l\epsilon^2$. Applying [4, (6.5.15)Theorem] we obtain a basic solution for the primal linear program. Rounding this leads to a feasible solution A satisfying $A \le (1+\epsilon)\mathrm{OPT} + O_\epsilon(1)$. The entire process uses time polynomial in n and $1/\epsilon$.

We close by noting that constructing a fully polynomial time approximation scheme (FPTAS) for extensible bin packing (note the absence of the word "asymptotic," i.e., we are now requiring $A \le (1+\epsilon)\mathrm{OPT}$) with a variable number of bins is at least as hard as constructing a fully polynomial approximation scheme for $P||C_{\max}$ with a variable number of processors, and hence impossible unless $P = NP$ [5]. To see this, note that for a given set of sizes and a given number m of bins (or processors), and with a bin size of B, if the extensible bin packing optimum cost exceeds m then the makespan exceeds B. On the other hand, if the extensible bin packing optimum cost is at most $m+\epsilon$, then the makespan is at most $B(1+\epsilon)$. Thus using binary search on B we could convert an FPTAS for extensible bin packing into an FPTAS for $P||C_{\max}$.

Note that the question of whether a problem admits a FPTAAS is fairly sensitive to the problem encoding. For example, suppose we were to modify the extensible bin-packing problem by letting the bin size be a parameter of the problem instance, instead of fixing it at one. Then a scaling argument, as in [5], would show that the additive term could be made negligible; thus even an FPTAAS would be impossible unless $P = NP$.

References

[1] Noga Alon, Yossi Azar, Gerhard J. Woeginger, and Tal Yadid. Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, 1:55–66, 1998.

[2] Paolo Dell'Olmo, Hans Kellerer, Maria Grazia Speranza, and Zsolt Tuza. A 13/12 approximation algorithm for bin packing with extendable bins. *Information Processing Letters*, 65:229–233, 1998.

[3] Martin Grötschel, Lásló Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[4] Martin Grötschel, Lásló Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Algorithms and Combinatorics 2. Springer-Verlag, 1988.

[5] Dorit S. Hochbaum and David B. Shmoys. Using dual approximation algorithms for scheduling problems: Theoretical and practical results. *Journal of the ACM*, 34(1):144–162, January 1987.

[6] Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 312–320, 1982.

[7] Frank D. Murgolo. An efficient approximation scheme for variable-sized bin packing. *SIAM Journal on Computing*, 16(1):149–161, 1987.

[8] Gerhard J. Woeginger. When does a dynamic programming formulation guarantee the existence of an FPTAS? In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 820–829, 1999.