COMBINATORICA 1(4) (1981) 349—355

BIN PACKING CAN BE SOLVED WITHIN $1+\varepsilon$ IN LINEAR TIME

W. FERNANDEZ de la VEGA

C. N. R. S., 54 Bd. Raspail
Paris 75 006, France*

and

G. S. LUEKER**

Dept. Info. Comp. Sci., Univ. of California
Irvine, CA 92 717 U.S.A.

Received 22 May 1981

For any list L of n numbers in $(0, 1]$ let L^* denote the minimum number of unit capacity bins needed to pack the elements of L. We prove that, for every positive ε, there exists an $O(n)$-time algorithm S such that, if $S(L)$ denotes the number of bins used by S for L, then $S(L)/L^* \leqq 1+\varepsilon$ for any L provided L^* is sufficiently large.

1. Introduction

In the bin-packing problem, we are given a list L of real numbers between 0 and 1 which we wish to assign to unit capacity bins in such a way that no bin receives numbers totalling more than 1 and a minimum number of bins is used. Let us denote this minimum by L^*. For any (heuristic) bin-packing algorithm S, let $S(L)$ denote the number of bins used for the input list L and $R_s(k)$ the maximum ratio $S(L)/L^*$ for any list with $L^*=k$. The asymptotic performance ratio of S, denoted by R_S^∞, is defined as $\overline{\lim}_{k\to\infty} R_s(k)$. The problem of finding an optimal packing can involve the solution of the NP-complete partition problem [8] (see also [3, p. 226]) and hence is likely to be computationally intractable. Many papers have been devoted to the study of heuristic algorithms with guaranteed bounds on performance [4, 5, 6, 11]. For a long time the best polynomial algorithm was the so-called First-Fit-Decreasing algorithm which has asymptotic performance ratio 11/9. Only quite recently an improved algorithm was found by A. C. Yao [11]. Even more recently, Garey and Johnson [4] have announced their "Modified First-Fit-Decreasing" algorithm which is stated to have performance ratio $\leqq 71/60$. A. C. Yao asks if there exists a limit to the best asymptotic performance ratio for a polynomial algorithm. In this paper we answer this question in the negative, showing that, for every positive ε, there is a *linear time* algorithm S for which $R_S^\infty \leqq 1+\varepsilon$. We also obtain a linear time algorithm for a generalization of bin-packing, namely the d-dimensional bin-packing problem of [2], again for every ε, with asymptotic performance ratio $\leqq d+\varepsilon$. The best algorithm described in [2]

* Mailing address: 4 bis rue Wulfran Puget, Marseille 13 008, France

** The work of this author was supported by NSF Grant MCS 79—04997.

AMS subject classification (1980): 68 C 25; 68 E 05, 90 C 10

for this problem, which is a generalization of the First-Fit-Decreasing algorithm, has guaranteed performance ratio $d+(1/3)$ for every input. Hence, choosing sufficiently small ε, our algorithm brings an improvement for large inputs. We refer to ([11], theorem 7) for a result, too long to be stated here, concerning lower bounds for the complexity in a particular model of computation of obtaining any performance ratio better than d.

2. Terminology

Besides the usual format of the data in a bin-packing problem which is assumed to be presented as a list $L=x_1x_2\ldots x_n$ of numbers in $(0, 1]$ (we shall represent lists as words in the one dimensional case since we use concatenation), we shall use also the notion of a multiset in which some elements may appear more than once. For instance, to a list L whose elements achieve m distinct values $y_1, \ldots, y_m$, we shall associate the multiset M_L denoted as in [9]

$$M_L = \{n_1 \cdot y_1, \ldots, n_m \cdot y_m\},$$

with the meaning that, for each i, n_i denotes the number of terms of L equal to y_i. We shall say that n_i is the multiplicity of y_i in M_L.

For any list L we denote its length by $|L|$. We denote by $V(L)$ the sum of the elements of L.

Let X be a bin used in a packing. By the type of X we shall mean the multiset of numbers which are assigned to X. For a given packing problem, determined by a list L whose elements take m distinct values $y_1<y_2<\ldots<y_m$, we can represent the types by vectors in N^m, the vector $T=(t_1, \ldots, t_m)$ with $\sum_{i=1}^{m} t_i y_i \leqq 1$, representing the type of a bin which contains, for each i, exactly t_i numbers equal to y_i. We shall say that the type T is saturated if we have $y_1+\sum_{i=1}^{m} t_i y_i > 1$.

We note that to each packing of a list L there corresponds in an obvious way a multiset of types of bins. If $\mathcal{M}$ is a multiset of types of bins corresponding to a packing of L and $M=M_L$ is the multiset of numbers corresponding to L, we shall say, with a slight abuse of language, that $\mathcal{M}$ is a packing of M.

3. The main theorem

Theorem 1. *For any positive ε there is a bin-packing algorithm S with asymptotic performance ratio $R_S^\infty \leqq 1+\varepsilon$, working in time $C_\varepsilon + Cn\log(1/\varepsilon)$, where C_ε depends only on ε and C is an absolute constant.*

The essential idea of the proof is to reduce (in linear time) bin-packing within $1+\varepsilon$ to a restricted problem in which the number of distinct values of the numbers to be packed is bounded and moreover these values are bounded from below. Hence the number of possible types of bins is bounded. We describe a nearly optimal algorithm for the restricted problem in section 3.1 and perform the reduction in section 3.2.

3.1. A restricted version of bin-packing.

For $\eta \in (0, 1]$ and any positive integer m, let RBP (η, m) denote the following problem.

INPUT: A multiset $M=\{n_1 \cdot x_1, \ldots, n_m \cdot x_m\}$ with values $x_1, \ldots, x_m$ in $[\eta, 1]$. We define the size of the input to be $n=\sum_{i=1}^{m} n_i$.

OUTPUT: A multiset of types of bins corresponding to an optimal packing of M.

Let M^* denote the cardinality of this multiset (taking into account the multiplicities). Hence $M^*=L^*$ where L is any list for which $M_L=M$. We observe that given such a multiset, it is trivial to define a packing of any such list L using a set of bins with types given by this multiset and that this can be done in linear time.

We shall say that an algorithm solves RBP (η, m) within an additive constant C if, when applied to any admissible input M, it finds a multiset of types of bins corresponding to a packing of M and with cardinality not exceeding M^*+C. We shall now prove the following

Fact. *For any given η and m,* RBP (η, m) *can be solved within an additive constant in time independent of n.*

Proof. Let η and m be given. We note first that, for any admissible input M for RBP (η, m), the number of possible types of bins is bounded by a function $\tau(\eta, m)$. In fact, setting $k=\left\lfloor \frac{1}{\eta} \right\rfloor$, we can take

$$\tau(\eta, m) = \binom{m+k}{k}$$

which is the number of ways one can choose $m+1$ nonnegative integers which add to k. (The first m numbers are item counts and the last one is for padding the sum out to k.)

Let $q=q(M) \leqq \tau(\eta, m)$ denote the actual number of types. Let these types be arranged in some order and let $T_i=(t_1^i, \ldots, t_m^i)$ denote the vector representing the ith type. Consider the following linear integer program, to which we shall refer as program P1.

$$\textit{Minimize} \sum_{i=1}^{q} a_i$$

subject to

$$(1) \qquad \sum_{i=1}^{q} a_i t_j^i \geqq n_j \quad j = 1, \ldots, m$$

and

$$a_i \geqq 0 \qquad i = 1, \ldots, q.$$

From a solution of this program, we can, by deflating perhaps some types, obtain a solution $(a_1^*, \ldots, a_q^*)$ for which the constraints (1) become equalities. Hence converting the constraints into equalities does not change the value of the program (it certainly can't increase it) and $(a_1^*, \ldots, a_q^*)$ defines in an obvious way an optimal multiset of types. Now if $(\alpha_1, \ldots, \alpha_q)$ is a solution of the ordinary linear program

P2 corresponding to P1 then $(\alpha_1', \ldots, \alpha_q')$, where $\alpha_i' = \lceil \alpha_i \rceil$, $i=1, \ldots, q$, satisfies the constraints of P1. From the multiset defined by $(\alpha_1', \ldots, \alpha_q')$ we can again obtain, by deflating perhaps some types, a multiset of types $\{a_1^{**} \cdot T_1, \ldots, a_q^{**} \cdot T_q\}$ corresponding to a packing of M and we have

$$\sum_{i=1}^{q} a_i^{**} \leqq \sum_{i=1}^{q} \alpha_i' \leqq \sum_{i=1}^{q} \alpha_i + q \leqq \sum_{i=1}^{q} a_i^* + q.$$

The inequality between the first and last expressions means that the multiset of types $\{a_1^{**} \cdot T_1, \ldots, a_q^{**} \cdot T_q\}$ solves RBP (η, m) on the instance M within the additive constant* q. To find $(a_1^{**}, \ldots, a_q^{**})$ amounts essentially to solving the ordinary linear program P2.

Since both the number of variables and the number of constraints of P2 have bounds depending only on η and m, this can be done in time depending only on these parameters. ∎

3.2. Converting bin-packing within $1+\varepsilon$ into restricted bin-packing.

Let $\varepsilon > 0$ be given and let $\varepsilon_1 = \varepsilon/(\varepsilon+2)$ denote the root of $(1+\varepsilon_1)(1-\varepsilon_1)^{-1} = 1+\varepsilon$. We set $m = \lceil \varepsilon_1^{-2} \rceil$. Let $L = x_1 \ldots x_n$ be the list to be packed and let l denote the number of x_i's smaller than ε_1. We define h by $n - l = mh + r$, $r \leqq m-1$, and suppose first $h > 0$. Let $L_1 = K_0 y_1 K_1 \ldots y_m K_m R$ be any list obtained by rearranging L in such a way that:

(i) K_0 contains the l terms $< \varepsilon_1$,
(ii) $|K_1| = |K_2| = \ldots = |K_m| = h-1$,
(iii) for $i = 1, \ldots, m-1$ each term x in K_i satisfies $y_i \leqq x \leqq y_{i+1}$,
(iv) each term x in $K_m R$ satisfies $x \geqq y_m$.

These conditions amount exactly to saying that, if L' is the list obtained by sorting L in non-decreasing order, then y_i is the term of rank $l + (i-1)h + 1$ in L' and, writing $L' = Y_0 y_1 Y_1 \ldots y_m Y_m$, that the sub-lists $K_0, K_1, \ldots, K_m R$ are arbitrary rearrangements of $Y_0, Y_1, \ldots, Y_m$ respectively. Hence to find a list L_1 fulfilling the forementioned conditions amounts essentially to finding the y_i's.

Using the fact that the "selection problem" which consists in finding the kth largest term of a list of length l can be solved in time $O(l)$ [1], it can be shown that all the y_i's can be found in time $O(n \log_2 m)$, hence in time $Cn \log(1/\varepsilon)$ where C is an absolute constant.

We require now the following definition. For any two lists of same length $L = x_1 \ldots x_n$ and $K = y_1 \ldots y_n$ we shall say that L dominates K and write $L \succ K$ or $K \prec L$ if we have $x_i \geqq y_i$, $i = 1, \ldots, n$. It is clear that $L \succ K$ implies $L^* \geqq K^*$.

Consider the lists $L_2 = K_0 y_1^h \ldots y_m^h R$ and $L_3 = K_0 y_2^h \ldots y_m^h 1^h R$. Clearly we have $|L_2| = |L_3| = n$ and $L_2 \prec L_1 \prec L_3$. Since from a packing of L_3 we can get immediately a packing of L_1 (hence a packing of L) using the same number of bins, if we show on one hand that we can indeed obtain in linear time a packing of L_3 using no more than $(1-\varepsilon_1)^{-1} L_3^* + C$ bins and, on the other hand, that the inequality $L_3^* \leqq L_2^*(1+\varepsilon_1)$ holds, then it follows that our packing of L uses less than

* D. Johnson pointed out to us that we can replace here q by m since P2 will always have a solution with at most m non-zero α_i's.

$(1+\varepsilon_1)(1-\varepsilon_1)^{-1}L_2^*+C \leqq (1+\varepsilon)L^*+C$ which is asymptotic to $(1+\varepsilon)L^*$ by the definition of ε_1 and the trivial inequality $L_2^* \leqq L^*$, and we shall be done.

Now the lists L_2 and L_3, thought of as multisets, differ only in that, in going from L_2 to L_3, we change h terms equal to y_1 into as many terms equal to 1. Hence, given a packing of L_2, we can get in an obvious way a packing of L_3 using no more than h additional bins. This implies $L_3^* \leqq L_2^*+h$ and, since $L_2^* \geqq mh\varepsilon_1 \geqq \frac{h}{\varepsilon_1}$, we get $L_3^* \leqq L_2^*(1+\varepsilon_1)$ which was the second assertion to be proved.

To prove the first put $Q=y_2^h \dots y_m^h 1^h$ so that $L_3=K_0QR$. We begin by putting each element of R in a bin using $|R| \leqq \varepsilon_1^{-2}$ bins. Next, our RBP algorithm with parameters η and m, applied to the multiset M_Q, will find in time $\leqq C_\varepsilon$ (since η and m depend only on ε) a multiset of types of bins of total cardinality $\leqq Q^*+\tau(\eta, m)$ into which we can pack Q in linear time. Moreover, there is room left in these bins for any multiset of numbers each $<\varepsilon_1$ whose sum does not exceed the quantity $S=Q^*(1-\varepsilon_1)-V(Q)$. Now we argue according to the relative values of S and of the sum $V(K_0)$ of the elements which remain to be packed.

If $V(K_0) \leqq S$, no extra bin is needed. In this case we have obtained a packing of L_3 in at most $Q^*+\tau(\eta, m)+\varepsilon_1^{-2} \leqq L_3^*+C$ bins, with $C=\tau(\eta, m)+\varepsilon_1^{-2}$.

If $V(K_0)>S$, then we first fill up the bins used for Q with elements of K_0 totalling at least S and we are left with elements totalling at most $V(K_0)-S$ which can be packed in no more than $(V(K_0)-S)(1-\varepsilon_1)^{-1}$ extra bins. In this case the total number of bins used is at most

$$Q^*+C+\big(V(K_0)-S\big)(1-\varepsilon_1)^{-1} = \big(V(K_0)+V(Q)\big)(1-\varepsilon_1)^{-1}+C \leqq L_3^*(1-\varepsilon_1)^{-1}+C$$

using the trivial inequality $L_3^* \geqq V(K_0)+V(Q)$.

Hence in both cases we found, as desired, a packing of L_3 using at most $L_3^*(1-\varepsilon_1)^{-1}+C$ bins. In the case $h=0$ that we have left aside, the list L has at most $m-1$ terms greater than ε_1 and packs trivially in no more than

$$m+V(L)(1-\varepsilon_1)^{-1} \leqq L^*(1+\varepsilon)$$

bins for sufficiently large L^*. This concludes the reduction and the proof of Theorem 1. ∎

4. An application to the d-dimensional case

In the generalized bin-packing problem discussed in [2] and [11], $L=(\vec{x}_1, \dots, \vec{x}_n)$ is a list of d-dimensional vectors ($d \geqq 1$), with each component of the vectors in the interval $(0, 1]$. It is required to pack these vectors into a minimum number of bins, which we again denote by L^*, in such a way that the sum $\vec{v}$ of vectors in any bin has each component $v_i \leqq 1$. We have not succeeded in extending our proof for the one dimensional case to any higher dimension. A basic obstruction comes here from the fact that there is no natural order defined on $[0, 1]^d$ for any $d>1$. We can only obtain, by a rather direct use of our one dimensional algorithm, the following result

Theorem 2. *For any $\varepsilon>0$ there is a linear time algorithm which solves the generalized d-dimensional bin-packing problem within $d+\varepsilon$.*

Proof. Let J_i denote the set of indexes of the vectors in L for each of which the largest component is the ith one (in case of ties, an arbitrary choice is made within the set of indexes of the largest components of the considered vector). Let L_i denote the list of the values of the ith component of the vectors with indexes in J_i. Clearly a packing of L_i defines a generalized packing of these vectors. Now, for any fixed ε, we can, using our one dimensional packing algorithm, pack L_i within $1+\varepsilon$ in linear time. Hence we can pack L in a number of bins not greater than $(1+\varepsilon)\sum_{i=1}^{d} L_i^*$ also in linear time. On the other hand the vectors with indexes in J_i do not pack in less than L_i^* bins and this implies

$$L^* \geqq \max\{L_1^*, \ldots, L_d^*\} \geqq d^{-1} \sum_{i=1}^{d} L_i^*.$$

Hence our packing achieves an accuracy ratio $\leqq (1+\varepsilon)d \leqq d+\varepsilon'$ which is what we wanted. ▮

5. Concluding remarks

We list some problems for future research.

(1) We have not given any evaluation for the constant C_ε in the expression of the time required by our algorithm but we remark that it is certainly huge. The number of types can exceed $\binom{m}{k}$ with $k=\lfloor 1/\varepsilon_1 \rfloor$ and $m=\lceil \varepsilon_1^{-2} \rceil$ so that C_ε cannot be bounded by a polynomial in $1/\varepsilon$. We ask: does there exist a family (S_ε) of bin-packing algorithms indexed by the parameter ε, such that $R_{S_\varepsilon}^\infty \leqq 1+\varepsilon$ and the time complexity of S_ε is bounded by a polynomial function of the two variables n and $1/\varepsilon$?

(2) Does there exist an $O(n)$-time bin-packing algorithm S with $R_S^\infty = 1$?

(3) Turning now to lower bounds for the best achievable accuracy, it is not known, as pointed out in [3], if there exists a polynomial time algorithm S with $|S(L)-L^*|$ bounded by a constant independent of L^*. Of course a negative answer to this question can only be conditional on $P \neq NP$.

(4) What is the best possible asymptotic performance ratio for a polynomial algorithm fort he d-dimensional case? A more modest question is: can we improve at all over $R_S^\infty = d$ in polynomial time? Let's mention that it is not possible to improve over the ratio d in linear time using the computational model of Yao (see [11, theorem 7]).

Acknowledgement. We are pleased to express our thanks to Jean François Maurras for valuable information concerning linear programming algorithms.

Note added in proof. It is worth noting that a number of pieces of our analysis have been employed in earlier papers, though it does not seem to have been noticed earlier that a bound of the type obtained here was possible. For example, in [14] the method of bucketing items according to some measure of size, and treating small items specially, was used to obtain approximations for the knapsack problem; see [3, 16] for a survey of methods for approximation algorithms. In [17] a method of attempting to make buckets have about the same number of items by using sampling was used in a sorting algorithm with good expected behavior; see [12] for a survey of algorithms with good expected behavior. [13, 15] considered the application of linear programming to bin packing in a manner similar to that employed here; [13] observes that a linear programming solution.

We are very grateful to David Johnson for bringing some of the earlier research to our attention, and for also making other helpful remarks on an early manuscript.

References

[1] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest and R. E. Tarjan, Time bounds for selection, *J. Comput. Sys. Sci.*, **7** (1973), 448—461.
[2] M. R. Garey, R. L. Graham, D. S. Johnson and A. C. Yao, Multiprocessor scheduling as generalized bin-packing, *J. Combinatorial Theory* A **21** (1976), 257—298.
[3] M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1979.
[4] M. R. Garey and D. S. Johnson, Approximation algorithms for bin packing problems: a survey, *preprint* 1980.
[5] D. S. Johnson, *Near optimal bin packing algorithms*, Ph. D. Th., MIT, Cambridge, Mass., June 1973.
[6] D. S. Johnson, Fast algorithms for bin packing, *J. Comptr. Syst. Sci.* **8** (1974), 272—314.
[7] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey and R. L. Graham, Worst case bounds for simple one-dimensional packing algorithms, *SIAM J. Comptg.* **3** (1974), 299—325.
[8] R. M. Karp, Reducibility among combinatorial problems, in: *Complexity of Computer calculations.* (R. E. Miller and J. W. Thatcher, Eds.) Plenum Press, New York, 1972, 85—103.
[9] D. E. Knuth, *The Art of Computer Programming, Vol. 3, Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.
[10] A. Schönhage, M. S. Paterson and N. Pippenger, Finding the median, *J. Comput. Sys. Sci.* **13** (1976), 184—199.
[11] A. C. Yao, New algorithms for bin packing. *J. ACM* **27**, 2 (Apr. 1980).
[12] J. L. Bentley, Probabilistic analysis of algorithms, *Applied Probability—Computer Science, the Interface,* Boca Raton, Florida, January 1981.
[13] P. C. Gilmore and R. E. Gomory, A linear programming approach to the cutting-stock problem, *Operations Research* **9** (1961), 849—859.
[14] O. H. Ibarra and C. E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *Journal of the ACM* **22** (1975), 463—468.
[15] L. V. Kantorovich, Mathematical methods of organizing and planning production, *Management Science* **6**, 4 (July 1960), 366—422.
[16] S. Sahni, General techniques for combinatorial approximation, *Operations Research* **25**, 6 (1977). 920—936.
[17] B. W. Weide, *Statistical Methods in Algorithm Design and Analysis,* Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania (August 1978); appeared as CMU Computer Science Report CMU—CS—78—142.