AN ANALYSIS OF SOME PACKING ALGORITHMS

M. R. Garey and R. L. Graham
Bell Telephone Laboratories, Inc.
J. D. Ullman
Princeton University

A potentially useful approach to evaluating heuristic techniques involves the determination of their worst-case deviation from the optimal solution. We describe the results of applying such an analysis to a variety of heuristic algorithms for one-dimensional bin packing, closely bounding their worst-case performance.

INTRODUCTION

The following abstract problem occurs in a variety of computer science and operations research contexts. We are given a set of objects O_i with O_i having weight $\alpha_i > 0$, $1 \leq i \leq r$. We have at our disposal an unlimited supply of boxes B_j, each with a maximum capacity of w units of weight. It is desired to assign all the objects to the minimum number N_0 of boxes subject to the constraint that the total weight assigned to any box can be no more than w.

In this formulation the problem takes the form of a typical loading or packing problem [3]. However, it is not difficult to show that it is also a special case of the one-dimensional cutting stock problem [5, 6, 7], the assembly-line balancing problem [1], and multiprocessing with deadlines [4, 8, 9], as well as occurring in certain memory allocation algorithms and table formatting schemes [11].

At present, no efficient (in the sense of Edmonds [2]) algorithm is known for the solution of this problem, and, in fact, many people strongly suspect that none exists. However, a number of heuristic algorithms have been suggested for producing good solutions. In this paper we briefly describe some recent results dealing with the worst-case behavior of some of these algorithms.

SOME HEURISTIC ALGORITHMS

By identifying each object O_i with its weight α_i, we can think of the problem as being one in which the *weights* α_i are packed into the boxes B_j. Consider the following algorithm for loading the α_i which we shall call the "first-fit" algorithm. For a given arrangement (or *list*) $L = (\alpha_1, \ldots, \alpha_r)$, the weights α_k are successively assigned in order of increasing k, each to the box B_j of lowest index into which it can validly be placed. The number of boxes thus required will be denoted by $N_{FF}(L)$, or just N_{FF}, when the dependence on L is suppressed.

If L is chosen so that $\alpha_1 \geq \cdots \geq \alpha_r$ then the first-fit algorithm using this list is called the "first-fit decreasing" algorithm and the corresponding $N_{FF}(L)$ is denoted by N_{FFD}.

Instead of first-fit, one might instead assign the next α_k in a list L to the box whose resulting unused capacity is minimal. This is called the "best-fit" algorithm and N_{BF} will be used to denote the number of boxes required in this case. The corresponding definitions of "best-fit decreasing" and N_{BFD} are analogous to first-fit decreasing and N_{FFD}.

One of the first questions which arises concerning these algorithms is the extent by which they can ever deviate from N_0. For N_{FF} this worst-case behavior is given by the following result.

Theorem ([11, 5]): For any $\varepsilon > 0$, if N_0 is sufficiently large then

$$N_{FF}/N_0 < 17/10 + \varepsilon \quad . \tag{1}$$

The 17/10 in (1) is best possible.

An example for which $N_{FF}/N_0 = 17/10$ is given by packing the 37 weights: 10 of weight 51; 10 of weight 34; 3 of weight 16; 7 of weight 10; and 7 of weight 6 into boxes of capacity $w = 101$. If the increasing list $L = (6, \ldots, 51)$ is used then $N_{FF}(L) = 17$. On the other hand if the decreasing list $L' = (51, \ldots, 6)$ is used then we find

$$N_{FF}(L') = N_0 = 10 \quad .$$

In fact, for any $\varepsilon > 0$, examples can be given [5] with $N_{FF}/N_0 \geq 17/10 - \varepsilon$ and N_0 arbitrarily large. It appears, however, for N_0 sufficiently large, that N_{FF}/N_0 is strictly less than 17/10. These examples also show that $N_{BF}/N_0 \geq 17/10 - \varepsilon$.

In order for the ratio N_{FF}/N_0 to achieve relatively large values, it is necessary for some of the α_i to be relatively large. This is stated precisely in the following result.

Theorem ([5]): Suppose

$$\frac{\max \alpha_i}{w} \leq \alpha$$

Then for any $\varepsilon > 0$, if N_0 is sufficiently large then

$$\frac{N_{FF}}{N_0} - \varepsilon \leq \begin{cases} 17/10 & \text{for } \alpha > 1/2, \\ 1 + \lfloor \alpha^{-1} \rfloor^{-1} & \text{for } 0 < \alpha \leq 1/2.^{*} \end{cases} \tag{2}$$

The right-hand side of (2) cannot be replaced by any smaller function of α.

It is conjectured† that the preceding theorem also holds when N_{FF} is replaced by N_{BF}. This is

---

*Where $\lfloor x \rfloor$ denotes the greatest integer $\leq x$.

†Added in proof: This has recently been established by A. Demers.

known to be true for $\alpha \leq 1/2$.

As one might suspect, N_{FFD}/N_0 cannot differ from 1 by as much as N_{FF}/N_0 can. This is shown in the following result.

Theorem ([5]): For any $\varepsilon > 0$, if N_0 is sufficiently large then

$$\frac{N_{FFD}}{N_0} < 5/4 + \varepsilon \tag{3}$$

Examples can be given [5] for which N_0 is arbitrarily large and $N_{FFD}/N_0 = 11/9 = N_{BFD}/N_0$. It is conjectured* that the term 5/4 in (3) can be replaced by 11/9; this has been established [5] for some restricted classes of α_i.

The bound in (3) also applies to the ratio N_{BFD}/N_0. This is implied by the following result.

Theorem ([5]): If

$$\frac{\min \alpha_i}{w} \geq 1/5$$

then $N_{FFD} = N_{BFD}$.

The quantity 1/5 above is best possible since for any $\varepsilon > 0$ examples can be given [5] for which

---

*Added in proof: This has recently been established by D. Johnson.

$$\frac{\min \alpha_i}{w} > 1/5 - \varepsilon ,$$

N_0 is arbitrarily large and

(4) $$\frac{N_{FFD}}{N_{BFD}} = 11/10 .$$

In the other direction, examples exist [5] for which N_0 is arbitrarily large and

(5) $$\frac{N_{BFD}}{N_{FFD}} = 10/9 .$$

The quantities 11/10 and 10/9 represent the largest values of the ratios N_{FFD}/N_{BFD} and N_{BFD}/N_{FFD} currently known (for large N_0).

Another algorithm which has been proposed [10] proceeds by first selecting from all the α_i a subset which packs B_1 as well as possible, then selecting from the remaining α_i a subset which packs B_2 as well as possible, etc. Although more computations would usually be required for this algorithm it might be hoped that the number $\bar{N}$ of boxes required is reasonably close to N_0. This does not have to be the case, however, since examples exist [5] for any $\varepsilon > 0$ for which N_0 is arbitrarily large and

(6) $$\frac{\bar{N}}{N_0} > \sum_{n=1}^{\infty} \frac{1}{2^n-1} - \varepsilon$$

The quantity

$$\sum_{n=1}^{\infty} \frac{1}{2^n-1} = 1.606695...$$

in (6) is conjectured to be best possible for this case.

SOME REMARKS

Some of the difficulty in proving many of the preceding results and conjectures seems to stem from the fact that a decrease in the values of α_i may result in an increase in the number of boxes required. For example, if the weights (760,395,305,379,379,241, 200,105,105,40) are packed into boxes of capacity 1000 using the first-fit decreasing algorithm then we find $N_{FFD} = 3$ which is optimal. However, if all the weights are decreased by 1, so that now the weights (759,394,...,39) are packed into boxes of capacity 1000 using the first-fit decreasing algorithm, we have $N_{FFD} = 4$ which is clearly not optimal. In fact, it can happen that N_{FF} can increase when some of the α_i are *deleted*. For example, if the first-fit algorithm is used to pack the weights in the list L = (7,9,7,1,6,2,4,3) into boxes of capacity 13 then $N_{FF}(L) = 3$. If the weight 1 is deleted from L to form L' = (7,9,7,6,2,4,3), then we obtain $N_{FF}(L') = 4$!

A number of interesting open questions remain, in addition to those already mentioned. For example, one could allow boxes of different capacities and study the behavior of $N_{FF}(L)/N_{FF}(L')$ for a fixed set of weights, as a function of the lists L and L', the ordering of the boxes, the distribution of the capacities and weights, etc. It would also be of interest to examine two-dimensional analogues of these problems in view of the applicability of the results [6, 7].

The determination of the worst-case behavior for these algorithms can also be considered as a first step in the analysis of their *expected* behavior. While generally being more useful in day-to-day applications, results of this type have typically been more difficult to obtain. A major problem in this regard is that accurate assumptions regarding the statistics of a "random" problem may be *impossible to obtain.*

As we remarked in the introduction, this paper is intended to serve as a brief summary of some recent results in this area. For a more complete treatment of these topics, the reader should consult [5] and [8].

ACKNOWLEDGMENTS

The work of J.D. Ullman was supported by National Science Foundation Grant GJ-1052.

REFERENCES

[1] Conway, R. W., Maxwell, W. L., and Miller, L. W., *Theory of Scheduling*, Addison-Wesley, Reading, 1967.

[2] Edmonds, J., "Paths, Trees and Flowers," *Canadian Journal of Mathematics, 17* (1965), 449-467.

[3] Eilon, S., and Christofides, N., "The Loading Problem," *Management Science, 17*, 5 (1971), 259-268.

[4] Fulkerson, D. R., "Scheduling in Project Networks," *Proc. IBM Scientific Computing Symp. on Combinatorial Problems*, IBM Corp., New York (1966), 73-92.

[5] Garey, M. R., Graham, R. L., and Ullman, J. D., "Worst Case Analysis of Some Memory Allocation Algorithms," Proc. Fourth Annual ACM Symposium on Theory of Computing, May (1972), 143-150.

[6] Gilmore, P. C., and Gomory, R. E., "A Linear Programming Approach to the Cutting Stock Problem," *Operations Research, 9* (1961), 849-859.

[7] ——, "A Linear Programming Approach to the Cutting Stock Problem II," *Operations Research, 11* (1963), 863-888.

[8] Graham, R. L., "Bounds on Multiprocessing Timing Anomalies and Related Packing Algorithms," *AFIPS Conf. Proc., 40* (1972), 205-217.

[9] ——, "Bounds on Multiprocessing Timing Anomalies," *SIAM Journal of Applied Mathematics, 17*, 2 (1969), 416-429.

[10] Prim, R. C., (Personal communication).

[11] Ullman, J. D., "The Performance of a Memory Allocation Algorithm," Tech. Report No. 100, Computer Science Laboratory, E.E. Department, Princeton University (1971).