The Annals of Probability
1992, Vol. 20, No. 4, 2164–2167

BOOK REVIEW

E. G. Coffman, Jr., and George S. Lueker, *Probabilistic Analysis of Packing and Partitioning Algorithms*. Wiley, New York, 1991, xiv + 192 pages, $46.95.

Review by Mark Jerrum

University of Edinburgh

Asked to provide an application of probability theory to computer science, a probabilist would no doubt respond with an example from the field of performance evaluation, where applications of queuing theory and the like to the analysis of real-time systems are by now well known. In these examples, probability enters through the interaction of a "reactive" program with an environment which behaves stochastically.

But suppose that instead of reactive programs we consider simple "transducers": programs that accept an input, compute for some time and eventually provide an output. Since the designers of computer hardware usually take great pains to ensure that the behaviour of their products is entirely deterministic, it might be imagined that the scope for applying probability theory to computer programs viewed as transducers is distinctly limited. Therefore, it may come as a surprise that the probabilistic analysis of algorithms has been a thriving area of theoretical computer science for at least fifteen years. The stochastic element may take one of two forms, which might be termed internal and external randomisation.

Internal randomisation arises when the model of computation is extended to include the potential to make random choices. Using such a model, one can describe and analyse *randomised algorithms*, whose execution proceeds stochastically. (In implementations of randomised algorithms, true random choice is replaced by some approximation based on pseudorandom number generators, since, as we have noted, real computers are deterministic.) Interest in randomised algorithms was sparked in the mid-1970s by the discovery of fast randomised tests for primality. Since then, many further applications of randomised algorithms have been found, particularly in computational algebra and geometry, and combinatorial enumeration. It would be moving away from the point to describe these applications in detail here, and the reader is directed instead to the excellent lecture notes of Prabhakar Raghavan (1990) or the survey paper of Welsh (1983), which although missing the more recent developments has the advantage of wider availability.

External randomisation arises when a deterministic algorithm is presented with problem instances drawn from a specified probability distribution. Quantities such as the execution time of the algorithm or the quality of the solution it produces become random variables of the input, and the aim is now to make

Received March 1992

probabilistic statements about these random variables. The motivation for introducing external randomisation is the following. Under a certain plausible assumption, the *theory of NP-completeness* can be used to show that many problems of practical interest are computationally intractable. However, the consequences of the theory of NP-completeness are statements of the general form: "Any algorithm solving problem X will necessarily run for a long time on *some* instances." Postulating a probability distribution on problem instances may provide a kind of escape from the tyranny of NP-completeness, provided, as is often the case, problem X turns out to be "easy on average." Although average case results can be interesting from a theoretical point of view, it is easy to overestimate their practical significance, as in real life the input distribution is seldom known, and even the assumption that it exists may be invalid.

Coffman and Lueker's book concerns itself with external randomisation in the context of various partitioning problems. A typical instance of such a problem is created by selecting numbers $X_1, X_2, \dots, X_n$ independently at random from a certain probability distribution. Two specific examples of partitioning problems are the *makespan scheduling problem* and the *one-dimensional bin-packing problem*. The former is motivated by scheduling tasks of known duration on an m-processor machine, and asks for a partition of the multiset $S = \{X_1, \dots, X_n\}$ into m blocks, such that the maximum block-sum is minimised. The latter problem is in some sense dual and asks for a partition of S into the smallest number of blocks such that no block-sum exceeds a predetermined bound. (We can imagine the items in S being packed into a number of "bins" of equal capacity.)

All the partitioning problems considered in the book, including the two described above, are NP-complete. This observation prompts the search for efficient heuristics that produce nearly optimal solutions with high probability. (Note that an assumed probability distribution on problem instances is crucial here.) An example of such a heuristic is the so-called *first fit decreasing* (FFD) heuristic for one-dimensional bin-packing. For this, the elements of S are first permuted into nonincreasing order, $X_1' \geq X_2' \geq \cdots \geq X_n'$, and the bins into which they are to be packed are given sequence numbers $1, 2, 3, \dots$. Then, setting i in turn to $1, 2, \dots, n$, the element X_i' is packed into the *lowest*-numbered bin that will accommodate it. To quantify the effectiveness of a heuristic such as FFD, one compares the number of bins used by the heuristic with the number of bins used by an optimal packing algorithm. In the case of FFD, with bins of unit size and items drawn from the uniform distribution on $[0, 1]$, it turns out that the ratio of these two numbers tends to 1 as n tends to infinity.

Although the problems considered in the book are motivated by real-life situations, it is not intended that one should draw serious practical conclusions from the results: as we have remarked, the distribution of problem instances is not known in practice; moreover, the heuristics that are considered are so simple that the temptation to adopt more sophisticated heuristics in real applications would be irresistible. However, the very simplicity of the problem scenarios and heuristics lends theoretical appeal to the results.

To give some idea of the depth to which these heuristics have been analysed, here is a result taken from the book (Theorem 6.3). At the risk of giving a false impression, I have selected it on grounds of surprise. Suppose $X_1, X_2, \dots, X_n$ are selected independently from the uniform distribution on $[0, a]$, where $0 < a \leq 1$. Denote by FFD $(X_1, \dots, X_n)$ the number of unit-sized bins used by the FFD heuristic when packing items $X_1, \dots, X_n$. The expression $\mathrm{FFD}(X_1, \dots, X_n) - \sum_{i=1}^{n} X_i$ gives the total wasted space in the bins under the FFD heuristic and is a random variable. The expectation of this random variable is $\Theta(1)$ if $a \leq \frac{1}{2}$, $\Theta(n^{1/3})$ if $\frac{1}{2} < a < 1$ and $\Theta(n^{1/2})$ if $a = 1$. [The notation $\Theta(f(n))$ stands for a function bounded above and below, for all sufficiently large n, by $c_1 f(n)$ and $c_2 f(n)$, where c_1 and c_2 are positive constants.] The surprise, of course, lies in the existence of discontinuities at $a = \frac{1}{2}$ and $a = 1$.

Coffman and Leuker's book can be divided into three parts. The first is concerned with laying foundations: Here are introduced the tools that have proved useful in the analysis of partitioning problems. These tools include general statistical techniques—conveniently gathered together in a single chapter that can be skimmed by some readers—and specialised results concerning "matching problems" in the plane. The middle part contains detailed analyses of a number of partitioning problems, the goal being to obtain as much information as possible about both the exact solution and the approximate solutions provided by various heuristics.

The final section is shorter and considers two possible extensions of packing to two dimensions. In *strip packing*, rectangles drawn from a specified distribution are to be packed as economically as possible into a semiinfinite strip of unit width. The rectangles are required to be pairwise disjoint and to have their sides aligned with the sides of the strip. The object is to minimise the length of the occupied portion of the strip. In *two-dimensional bin packing*, a set of rectangles drawn from a specified distribution is to be packed into a number of two-dimensional bins, which are squares of unit side. Again, the packing must be disjoint and orthogonal, and the aim is to minimise the number of bins used. It is shown that relatively simple heuristics can produce results that are asymptotically close to optimal.

It will come as no surprise that this book of 192 pages devoted to the probabilistic analysis of partitioning algorithms is a thorough and scholarly work. It is moreover, accurate and clearly written. Aside from a handful of people with a specialist's interest in the topic, who are the potential readers of this book? I think they come from two groups.

Computer scientists working in the area of randomised algorithms will find the book a useful source of analytical techniques. The layout of the book is well suited to their purpose, as the basic statistical methods are separated from the applications and are described in a separate, self-contained chapter.

Probabilists may view the book as an introduction to the probabilistic analysis of algorithms and as an indicator of possible research directions. They too will find the structure of the book convenient. The problem scenarios are simple enough that they have a theoretical appeal independent of their original motivation within computer science. Although the dilettante computer scientist has learnt many statistical tricks, there is surely much for the specialist to contribute. Books such as this one may encourage fruitful collaboration.

REFERENCES

PRABHAKAR RAGHAVAN (1990). Lecture notes on randomized algorithms. Research Report RC 15340, IBM T. J. Watson Research Center, Yorktown Heights, New York.

WELSH, D. J. A. (1983). Randomised algorithms. *Discrete Appl. Math.* **5** 133–145.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF EDINBURGH
EDINBURGH EH9 3JZ
SCOTLAND