Journal of Heuristics, (Submitted, under review), 1–3 (1998)

A Response to "On method overfitting"

IAN P. GENT ipg@cs.strath.ac.uk
Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, United Kingdom

Abstract. If special-purpose methods can be overfitted to benchmarks, so can general methods.

Keywords: bin packing, heuristics, empirical study of algorithms, benchmark problems

I am very pleased to have the opportunity to respond to 'On method overfitting' (Falkenauer.) The main issue in dispute is the importance of work which uses simple and perhaps overfitted methods to solve benchmark problems. Falkenauer concentrates on my solution of open instances of his 'uniform' class of bin packing benchmarks using manual and simple computer methods [4], and questions its usefulness. I argue that such work is important precisely because it shows that the problems are overfittable. If special-purpose methods can be overfitted to benchmarks, so can general methods if the set of instances remains fixed and limited. The best response is to lessen the importance attached to existing benchmarks, while developing new ones without known simple methods of solution.

There are a number of points on which there is little or no dispute, not least that Falkenauer deserves credit for constructing the benchmarks and similarly the OR Library [1] for making them available world-wide. Their availability has certainly helped other researchers, including me. However, I believe that a benchmark repository should make every effort to maintain the current best known solutions of its problems. It is therefore a disservice to the community that, at time of writing, the OR Library still gives the impression that some of Falkenauer's problems are still open. Not only have I solved them, but so has J.M. Valério de Carvalho (to appear) independently, and I believe before me.

On one point Falkenauer has identified a mistake in my own paper. I believe he is correct that the dominance criterion is effective on instances with many objects per bin. Therefore my own sentence starting "Indeed, it could even be argued" (Gent 1998, page 302) is wrong and I regret that the paper went to press including it.

A large part of Falkenauer's paper criticises the methods I used to solve his benchmark problems as overfitted to the particular problems. There is no need to defend myself against this criticism. I never claimed any general value to the methods I used. Instead, I wrote that "this paper is not ...intended to show the superiority of my methods" (Gent 1998, page 301) and "Rather than make extravagant but unjustified claims for the methods used to solve these problems, ..." (Gent 1998, page 302). Another point on which no defence is necessary is that some of my solutions were obtained by hand. The points I made about the benchmarks were based on the simplicity of the methods used to solve them rather than the hardware used. The key point, and there is no argument about this, is that my methods, while certainly special-purpose, were also very simple.

Falkenauer's arguments against my methods are flawed, although these technicalities are unimportant compared to a basic agreement that my methods were simple and special-purpose. He argues that their success is due to items having integer size and the fact that items often fill bins exactly. All bin packing problems have integer item sizes, by the definition of problem SR1 given by Garey and Johnson (1979). Nor do my methods rely on full bins: each bin in turn is filled as full as possible, and this often leaves gaps. Finally, he states that where bins are filled incorrectly on the first pass, "Gent's procedure gives us little indication on how to proceed from there" (Falkenauer.) In fact, I described the repair methods used manually when this situation arose in solving the last uniform problem.

Falkenauer accuses me of claiming benchmarks solvable by simple methods to be useless. Of course easy benchmarks are not useless. If one method solves all hard problems, while another solves all those in the same time and all easy ones too, the second is clearly to be preferred. Alternatively, methods well tuned to hard problems can be overkill when it is only necessary to solve easy problems.

One general point remains at issue, namely the value of benchmark instances solvable by simple methods. I wrote: "the fact that these problems are amenable to solution by very simple methods must bring into doubt any conclusions based on the perceived hardness of the problems" (Gent 1998, page 301.) Falkenauer's claim is that "such benchmarks represent reasonably difficult BPP [Bin Packing Problem] instances ... provided a *general* BPP method is being tested" (Falkenauer.)

A thorny problem, perhaps an intractable one, arises if Falkenauer's claim is to be accepted, but he does not address it. Given a general method, how do we know it does not take advantage of the special structure of particular benchmarks? For example, suppose I had invented a stochastic search algorithm for bin packing. I might well have implemented the heuristic methods described in [4] to generate starting positions in search, and this would have led to spectacularly good run times on Falkenauer's uniform problems. This illustrates precisely the contribution my work made for bin packing researchers: in the future researchers can assess whether or not a given method is likely to have an undue advantage on these test instances. But my hypothetical case is a clear cut one. Chance features of algorithms might lead to them having advantages of no general importance. Many innocent and incidental choices in implementation run the risk of accidentally taking advantage of the particular structure of a problem. Assessing whether this has happened is problematic. It is safest to assume that the benchmarks are easy and to assign a low weight to them when comparing algorithms.

The approach I have just recommended is mirrored by two extensive studies of benchmark problems in the evolutionary algorithms and machine learning communities. Whitley, Rana, Dzubera and Mathias (1996) give advice on constructing test data for evolutionary algorithms. Of particular relevance, they write that "Evolutionary algorithms are best applied where where simpler methods fail" (Whitley *et al.* 1996, page 274.) Their general guidelines on the construction of test suites contain the advice to include problems where evolutionary algorithms (or other complex heuristic methods) outperform simpler methods. Also, they advise that test suites should be open ended: fixed test suites are not necessarily suitable for hypothesis-driven testing as recommended by Hooker (1995). The machine learning community has learnt from Holte's (1993) demonstration of the ability of very simple classification methods to perform well on the large and widely used UCI Machine Learning Repository [8]. Conscientious researchers now use a mixture of old and new benchmarks, as well as often testing large numbers of benchmarks [6]. Holte believes that a repository such as at UCI is still valuable as long as it continually grows to avoid the danger of the community overfitting its algorithms to a fixed collection [6].

The comments of Whitley *et al.* and Holte are consistent with my own arguments. Old benchmarks should be retained while new ones are continually added. Work showing that simple methods, however special-purpose, can solve existing benchmarks should not be despised because the methods used are overfitted. Instead, it should be welcomed precisely because it shows that overfitting is possible. The community which complacently accepts a fixed set of benchmarks is the one in most danger of overfitting its general methods.

Let me end on a lighter note. This work arose out of having the hubris to try to solve an open combinatorial problem with pen, paper, and a dram of the water of life. To my surprise, I succeeded. My advice is:

Don't be afraid of open problems: you never know, you might solve them.

References

1. J. E. Beasley. 1990. Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society* 41:1069–1072.
2. E. Falkenauer. On method overfitting. *Journal of Heuristics.*
3. M. R. Garey and D. S. Johnson. 1979. *Computers and intractability : a guide to the theory of NP-completeness.* W H Freeman.
4. I. P. Gent. 1998. Heuristic solution of open bin packing problems. *Journal of Heuristics* 3(4):299–304.
5. R. C. Holte. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11:63–91.
6. R. C. Holte. 1998. Personal communication.
7. J. N. Hooker. 1995. Testing heuristics: We have it all wrong. *Journal of Heuristics* 1:33–42.
8. C. J. Merz and P. M. Murphy. 1998. UCI repository of machine learning databases. Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.
9. J. M.Valério de Carvalho. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research.* To appear in volume on Combinatorial Optimization: Recent Advances in Theory and Practice.
10. D. Whitley, S. Rana, J. Dzubera and K. E. Mathias. 1996. Evaluating evolutionary algorithms. *Artificial Intelligence* 85:245–276.

Acknowledgments

I am a member of the APES research group at Leeds and Strathclyde Universities (http://www.cs.strath.ac.uk/~apes). I thank Rob Holte, Phil Kilby, Patrick Prosser, Paul Shaw, Judith Underwood, and Toby Walsh for discussions influencing this paper.