BNAIC 2005, The 17th Belgian-Dutch Conference on Artificial Intelligence, Katja Verbeeck, Karl Tuyls, Ann Nowe, Bernard Manderick and Bart Kuijpers *editors*, Brussels, 17-18 October 2005, pp365–367, Royal Flemish Academy of Belgium for Science and Arts

EVOLVING BENCHMARKS

W. B. Langdon

Department of Computer Science, University of Essex, Colchester CO4 3SQ, UK

Abstract

Genetic programming (GP) is used to evolve global optimisation test problems. These automatically generated performance metrics are used to show strengths and weaknesses of Particle Swarm Optimization (PSO) and Differential Evolution (DE). Knowledge gained will help when choosing maximisers (and their tuning parameters) and in research into new search tools (which might include hyperheuristics).

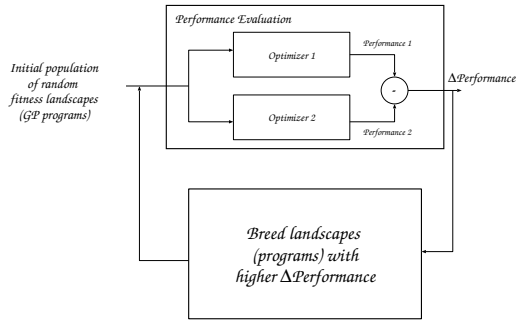
1 Introduction

There are many optimisation algorithms for which we do not have an adequate understanding of why certain parameter settings, or certain variations, perform better or worse than others. We suggest an experimental method to augment deep mathematical analysis which automatically finds test cases which demonstrate the strength and weaknesses of modern search heuristics [1]. It can be used to contrast any pair of optimisers. Section 3 gives a compressed demonstration: comparing Particle Swarm Optimisation (PSO) and Differential Evolution (DE).

The conventional approach is to study the performance of algorithms on a standard suite of problems, attempting to find the reasons behind relative success or failure. We turn this idea on its head: instead of studying the performance of two optimisers on a standard problem in the hope of finding an informative difference, we evolve (using genetic programming) new problems that maximise the difference in performance between the optimisers. Thus the underlying strengths and weaknesses of each optimiser are exaggerated and thereby revealed.

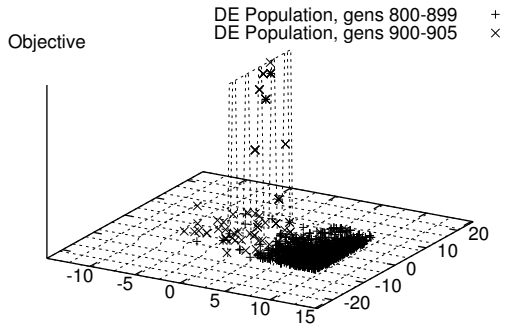
2 Method

We use tinyGP to evolve problems on which one search technique performs radically better or worse than another. We begin with a GP population in which each individual represents a problem landscape that can be searched by each of the two techniques. The fitness of an individual is the difference between the performances of the two techniques on the function represented by it. Each optimiser is run 5 times for each fitness evaluation. With this approach, GP will tend to evolve benchmark problems where one technique outperforms the other.

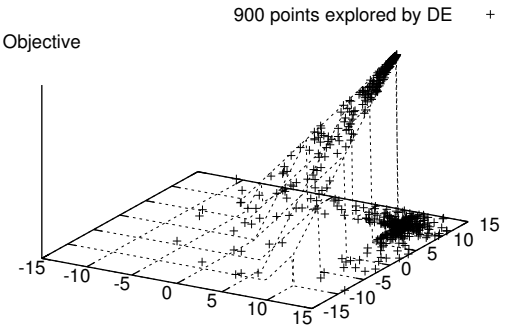


GP runs both optimisers on evolved benchmark. Fitness of benchmark is their difference.

GP steady state population 1000. Fitness is difference in optimiser performance, it is re-evaluated every time parents are selected using binary tournaments. 10% crossover or 2% mutation chance per tree node. Primitives: $+ - \times$ x, y and 100 constants.
PSO 30 particles, 1000 generations, no constriction or friction, particle speed $\leq \pm 10$.
DE used Rainer Storn's parameter settings: pop 20, 1000 gens, 90% crossover, F 0.8, "DEBest2Bin" strategy.



Landscape in which 76% of runs PSO finds optima before DE. In this run the DE population (+) converges toward one edge of the legal region. Later DE finds the spike (×).



Evolved landscape where DE finds optimum faster than PSO. + show points explored by DE. The optimum is small and PSO finds it in only 12% of runs.

3 PSO beats DE (left) *and* DE beats PSO (right)

In this run (left), after 900 generations DE finds the spike by chance and then rapidly climbs its peak (generation 905). PSO seems to have fewer problems stumbling into it by chance.

DE (right) behaves like "lemmings". I.e. it tends to waste many fitness evaluations in the infeasible region just over the cliff. [2] suggests that global optima of constrained problems are often at the boundary between a smoothly varying feasible region and an infeasible region (where a constraint is violated). I.e. cliff edges may be common in constrained problems and so *DE might not perform well on constrained optimisation problems.* The global optimum occupies $2.5\ 10^{-7}$ of the search space. This target proves to be too small for PSO, which seldom finds it. *This shows a weakness of the* standard *PSO: the particles are unable to home in on "narrow" global optima.*

Other results [1] suggest *DE has limited ability to move large distances once the population is tightly clustered.* It gives examples of *DE deceptive landscapes.* Also it suggest *a PSO without constriction or friction can not focus its search for long.* If the optimum is not found early, it is unlikely to be found later. This is the opposite of a GA, which tends to focus its search in later generations, rather than expand it.

This idea is widely applicable. The essential ingredient is a way of comparing the optimisers. Given this, it could be extended to higher dimensions and multi-objective optimisers. We could even consider extensions to more than two optimisers, although some form of population or fitness sharing, archive, demes or niches, might be needed. Indeed a co-evolutionary approach would allow a continuous range of optimisers to be tested. Either parameters or even the basic form of the optimiser, might be adapted. However it seams unlikely, indeed counter to the "no free lunch" theorems, that one could automatically find an optimiser that is easily able to solve every problem GP could throw at it. Perhaps a more serious problem is we might lose the simplicity of the technique and/or the evolved benchmarks. This would not help us to understand sophisticated search techniques.

4 Conclusions

Theoretic analysis of evolutionary algorithms in general, and PSO and DE in particular, is very hard. GP can help our understanding, by forcing alternative techniques to compete inside computers (rather than scattered across the pages of diverse conferences and journals) it can readily produce examples which illustrate their comparative strengths and weaknesses.

References

[1] William B. Langdon and Riccardo Poli. Evolving problems to learn about particle swarm and other optimisers. In CEC-2005, volume 1, pages 81–88, 2-5 September 2005. IEEE Press.

[2] Marc Schoenauer and Zbigniew Michalewicz. Evolutionary computation at the edge of feasibility. In *PPSN IV*, *LNCS 1141*, pages 245–254, 1996. Springer.