Controlling the Tradeoff between Time and Quality by Considering the Reproductive Potential of Offspring

Eddy Parkinson
School of Computer Science
Adelaide, SA 5005, Australia
edward.parkinson
@adelaide.edu.au

Adam Ghandar
School of Computer Science
Adelaide, SA 5005, Australia
adam.ghandar@
adelaide.edu.au

Zbigniew Michalewicz *
School of Computer Science
Adelaide, SA 5005, Australia
zbyszek@cs.adelaide.edu.au

Andrew Tuson
School of Informatics,
City University
London EC1V OHB, UK
a.tuson@soi.city.ac.uk

ABSTRACT

To improve evolutionary algorithm performance, this paper proposes a strategy to aid ascent and to help avoid premature convergence. Rapid increases in population fitness may result in premature convergence and sub optimal solution. A thresholding mechanism is proposed which discards child solutions only if their fitnesses are either too bad, in which case they are discarded, nor too good, in which case they pose the danger of premature convergence. This strategy is evaluated using two combinatorial optimization problems: the classic TSP benchmark and the more constrained vehicle routing problem (VRP) benchmark. The idea offers a relatively straight forward method for adding value by improving both runtime or solution quality. We consider a stochastic hill climber and a population based heuristic (an evolutionary algorithm).

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods

General Terms

Logistics, Optimization

Keywords

Vehicle Routing Problem, Threshold, Meta-heuristic, Reproductive Potential

1. INTRODUCTION AND BACKGROUND

The typical evolutionary search revolves around fitness increases over several generations. We present evidence that large increases in initial fitness can lead to *premature convergence* and it is possible to improve final solution fitness by rejecting them.

We consider setting an upper and lower bound on the fitness of new solutions such that new solutions falling outside these bounds are rejected. The bounds are able to be fixed, self adaptive, or deterministically set over the progress of the search. This is shown in the Figure 2 which shows a deterministic setting of the window of acceptable offspring quality during the search. Threshold have been examined in [3] [2].

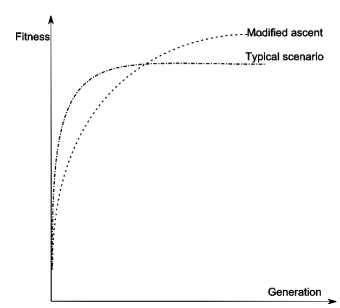

Figure 1: Threshold Heuristic that prevents very rapid jumps in initial fitness

Meta-heuristics select some solutions and discard others, but there looks to be a gap in our understanding, for example: Lower quality solutions (solutions with a low fitness) are discarded by *Simulated Annealing* [5] and *Tabu Search* [4], but their strategies are quite different. Lower quality solutions are always discarded by both meta-heuristics, either at the end of the search or the beginning of the search. Simulated Annealing discards many lower quality solutions at the end of the search and Tabu Search does almost the exact opposite, discarding them at the beginning. Lower quality solutions are kept or discarded, but to some extent the decision appears to be arbitrary rather than as a result of scientific analysis.

Meta-heuristics appear to ignore small improvement strategies Meta-heuristics do not appear to include a strategy of

*Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland,
and Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland

Copyright is held by the author/owner(s).
GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

preferring small improvements. Some meta-heuristics prefer large improvements over small improvements. Other meta-heuristics treat all improved solutions as being equal, but non appear to use a strategy of preferring small improvements. More recent meta-heuristics such as Ant Colonies Optimisation [1] and Co-Evolutionary Optimisation [6] continue the tradition of preferring large improvements. It looks like the strategy of preferring small improvements is not considered by meta-heuristics, this suggests there is a gap in our knowledge.

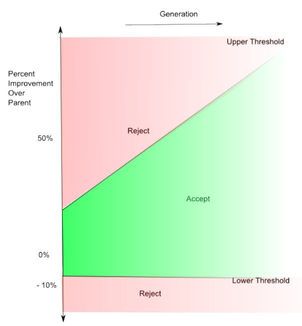

Figure 2: Estimating the potential of offspring to increase overall performance through an acceptable solution quality range (set relative to the parents.) This acceptable range is changed deterministically during the search process (in this case small improvements are favoured early in the search and, progressively, larger improvements are allowed later.

2. EXPERIMENT METHODOLOGY AND RESULTS

Many meta-heuristics are based on iterative improvement, they create improved solution after improved solution. Meta-heuristic iterative improvement can be divided into finding a new best solution and avoiding local optima. A meta-heuristic, in theory, is able to reject solutions that lead to local optima, but working out which solutions to reject can be problematic. To help with this problem the methodology deliberately separates the issue of avoiding local optima from the issue of finding a new best solution, these issues are referred to as Improvement Isolation and Improvement Preference respectively.

Improvement Isolation, both faster and better quality

The lower threshold (see Figure 2) produced a valuable result that is a little counter intuitive. Intuition would suggest that a lower threshold that rejects fewer solutions would improve final solution quality at the cost of longer execution time, and this matched the results up to a point. But the results for the VRPTW showed that setting the threshold too low had a lose-lose impact, the final solution quality was reduced as well producing a longer execution time.

Two threshold variables were tested, solution distance and route slack time, both produced the same positive result and improved both speed and final solution quality. If the threshold was too low or too high then both speed and quality were damaged. This looks to be a simple way to produce a small improvement in solution quality while at the same time making the algorithm faster. This was true for all the Solomon VRPTW problem types with both the EA and hill climber implementations.

Improvement Preference The assumption that large improvements are generally benifitial does not match the results described in this paper. The results shows the same pattern 6 times. The results show that for the VRPTW the heuristic methods that prefer small improvements overtake those that prefer large improvements. There was a gain of 3at the cost of a 10 second (30%) increase in execution time.

When the amount of slack time in the route was used to guide the search, gradually reducing the amount of slack time in the route produced the best results. Selecting child solutions that reduced the slack time in the route by a small amount consistently produced better quality final solutions. Although the percentage gain was well below 1

Both genetic algorithm and hill climber algorithms followed the improvement preference patterns described above.

Conclusion

With both VRPTW and TSP, the research results showed the pair of thresholds produced improvements in run-time and/or quality. The pair of threshold were able to improve final solution quality by a few percent and in some cases also reduce algorithm run time. The results show the following patterns:

1. Rejecting large jumps in initial fitness at the beginning of the search, improved final solution quality with VRPTW and TSP problem.

2. Accepting only large jumps in initial fitness at the beginning of the search produced the fastest results with the VRPTW.

3. Rejecting all solution modifications below a certain threshold improved both final solution quality and algorithm run-time.

2.1 Acknowledgements

This work was partially funded by the ARC Discovery Grant DP0985723 and by grants N 516 384734 and NN519 578038 from the Polish Ministry of Science and Higher Education (MNiSW).

3. REFERENCES

[1] M. Dorigo and L. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1):53 –66, Apr. 1997.

[2] G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86 – 92, 1993.

[3] G. Dueck and T. Scheuer. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *J. Comput. Phys.*, 90(1), 1990.

[4] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):pp. 671–680, 1983.

[6] M. A. Potter and K. A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.*, 8:1–29, March 2000.