How to Get More from Your Model: The Role of Constructive Selection in Estimation of Distribution Algorithms

J.R. Caldwell
University of Southampton
University Road
Southampton SO17 1BJ, UK
J.R.Caldwell@soton.ac.uk

R.A. Watson
University of Southampton
University Road
Southampton SO17 1BJ, UK
R.A.Watson@soton.ac.uk

ABSTRACT

Model-building optimisation methods aim to learn the structure underlying a problem and exploit this to direct the exploration of solutions. This generally interleaves two processes: Generating samples (from the model), and updating the model (using selected samples). In most estimation of distribution algorithms (EDAs), e.g. BOA, selection is used only in the latter, to determine which samples are retained for updating the model. In contrast, other evolution-inspired algorithms (such as rHN-G and MACRO) use selection differently - within the process that generates samples from the model. It has been hypothesised that this 'constructive selection' process can facilitate optimisation that other EDAs cannot but this has not been previously shown. Here we investigate these distinctions using constraint optimisation problems with a very simple modular structure. We find that a simple constructive selection method (rHN-g) can solve these problems in time polynomial in the problem size whereas other methods, such as BOA, require exponential time. We confirm that this result arises not from any difficulty in acquiring an accurate model but because of how samples are generated given the model. This suggests that by using constructive selection other EDAs could exploit the models they learn more efficiently to solve otherwise unsolvable problems.

CCS CONCEPTS

•**Theory of computation → Evolutionary algorithms;**

KEYWORDS

Constructive selection; Filter selection; Model building optimisation; Estimation of distribution algorithm; Multi-scale search; Bayesian optimisation algorithm; Restart Hopfield network with generative association; Genetic algorithms; Time-Complexity

ACM Reference format:
J.R. Caldwell and R.A. Watson. 2017. How to Get More from Your Model: The Role of Constructive Selection in Estimation of Distribution Algorithms. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076049

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '17 Companion, Berlin, Germany
© 2017 Copyright held by the owner/author(s). 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076049

1 INTRODUCTION

Model-building optimisation methods can learn how to decompose an optimisation problem into nearly-separable sub-problems without *a priori* knowledge of the underlying problem structure. Identifying such modularity aims to exploit the familiar idea of separating a problem into smaller, simpler sub-problems, solving these sub-problems and then re-combing these solutions to solve the original problem. This forms the basis of the building block hypothesis [2, 4].

Here we show that using selection during the generation of samples from the learnt model results in a significant efficiency gain over the more familiar use of selection which is to filter out candidate solutions. We introduce the terms of filter selection (the conventional approach) and constructive selection:

EDAs with Filter selection: The model is used to generate complete solutions without any input from selection. Selection is then used to determine whether that solution is good enough to be used to update the model.

Constructive selection: The model and selection are used together to generate solutions. Specifically, selection is used (within an iterative process) to assess whether a model-informed modification to a solution improves its fitness. After generating the samples the model is updated accordingly.

Filter selection is the more familiar type of selection used in all genetic algorithms and Estimation of Distribution Algorithms (EDAs). EDAs generate new candidate solutions probabilistically from the model and apply a filter to determine whether it is then used to update the model (using selection after generation) [3]. Constructive selection is an idea developed in recent work by Watson and colleagues in the development of multi-scale search algorithms (MSS). They differ to EDAs, namely in the generation of samples where partial solutions are probabilistically generated and selection is used to determine if the partial solution is incorporated into the candidate solution (using selection during generation). The restart Hopfield Network with Generative associations (rHN-G) [8] is a simple example of constructive selection.

Here we are focused on the performance difference between filter selection and constructive selection. As such, the Bayesian Optimisation Algorithm (BOA) [6] is selected as the example for filter selection due to its widely acknowledge performance on difficult model-building optimisation problems [1, 5, 7]. Whilst rHN-G's performance is basic and limited, the fact it performs constructive selection enables it to outperform a sophisticated algorithm for optimisation problems with hierarchical modular structure. Such a problem is the idealised nearly-decomposable constraint optimisation problem: the modular constraint (MC) problem [8]. The

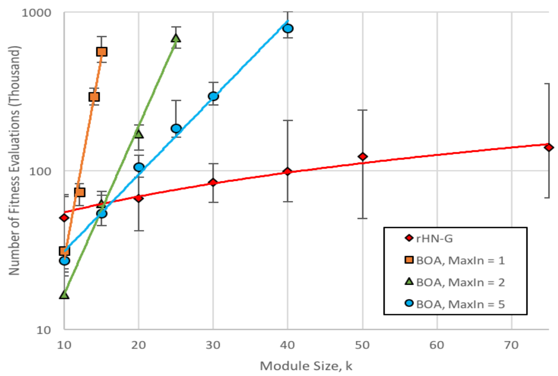

Figure 1: Fitness evaluations required to solve the MC problem. The error-bar shows the range of values from the 50 independent runs. The solid lines are trend lines calculated from the average data points. Maxin represents the number of incoming edges into a node in the Bayesian graph. BOA results use an exponential fit and the rHN-G result uses a linear fit.

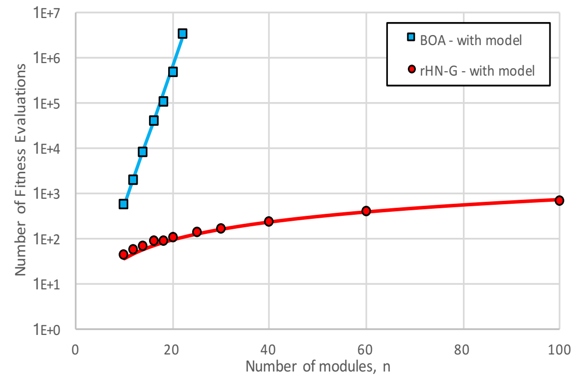

Figure 2: Average number of fitness evaluations required to solve the MC problem given BOA and rHN-G have been supplied with the problem structure. BOA scales exponentially (2^n) and rHN-G scales as $n \log n$.

problem is chosen to specifically highlight the performance difference between the selection methods: precisely what constructive selection can do and filter selection cannot. Thus, we argue, by using constructive selection it is possible to exploit the learnt information in a more efficient manner for this class of problem.

2 RESULTS

Fig. 1 presents the average number of fitness evaluations required to solve the MC problem. The superior performance of rHN-G allowed us to quantify results for significantly larger problem sizes than for BOA. For all the BOA experiments the number of fitness evaluations required scales exponentially with module size even when increasing the complexity of the model. The efficiency gain seen in BOA by increasing MaxIn is misleading as the time-complexity of model induction is not included. The results show that by using constructive selection in replacement of filter selection provides a significantly more efficient algorithm. The problem structure is not difficult to learn for both algorithms, however the way in which the information is exploited by selection determines whether the problem can be solved or not.

To further illustrate this, we separate the components of the time-complexity from model induction and generating samples. Both algorithms are provided with the model that identifies modules but we do not provide information on how to put the modules together. This effectively makes the algorithms the same: the only difference is the selection method. Learning the inter-module dependencies is very difficult but rHN-G does not need to learn the inter-module connections to solve the problem: constructive selection puts the modules together correctly and easily.

Fig. 2 presents the results for BOA and rHN-G given the model. It is clear from this figure that even given the correct model of the modular structure BOA is still unable to solve the problem in polynomial time. BOA required an exponential fit, specifically $0.5(2^n)$, whereas rHN-G only required $1.1(n \log n)$. This result confirms that the efficiency saving seen between rHN-G and BOA is a result of using constructive selection over filter selection.

3 CONCLUSION

Here we introduce the term of constructive selection to differentiate from the conventional (filter) use of selection. We have found that using constructive selection in model-building optimisation methods results in a reduction in time-complexity required to solve a modular constraint optimisation problem from exponential, using filter selection, to polynomial, using constructive selection. We verify that this result is caused by differences in the ability to generate good samples from the model and not from differences in model induction complexity.

ACKNOWLEDGMENTS

We acknowledge financial support from the EPSRC CDT in Next Generation Computational Modelling grant EP/L015382/1. This work is partially supported by Templeton project 60501.

REFERENCES

[1] Uwe Aickelin, Edmund K Burke, and Jingpeng Li. 2007. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society* 58, 12 (2007), 1574–1585.

[2] David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[3] Mark Hauschild and Martin Pelikan. 2011. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation* 1, 3 (2011), 111–128.

[4] John H Holland. 1975. Adaptation in natural and artificial systems. (1975).

[5] Martin Pelikan and David E Goldberg. 2003. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Genetic and Evolutionary Computation Conference*. Springer, 1271–1282.

[6] Martin Pelikan, David E Goldberg, and Erick Cantú-Paz. 1999. BOA: The Bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*. Morgan Kaufmann Publishers Inc., 525–532.

[7] Roberto Santana, Pedro Larrañaga, and Jose A Lozano. 2008. Protein folding in simplified models with estimation of distribution algorithms. *IEEE transactions on Evolutionary Computation* 12, 4 (2008), 418–438.

[8] Richard A Watson, Rob Mills, and Christopher L Buckley. 2011. Transformations in the scale of behavior and the global optimization of constraints in adaptive networks. *Adaptive Behavior* (2011), 1059712311412797.