Meta-Evolved Empirical Evidence of the Effectiveness of Dynamic Parameters

Brian W. Goldman
Natural Computation Laboratory
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, U.S.A.
bwgxc9@acm.org

Daniel R. Tauritz
Natural Computation Laboratory
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, U.S.A.
dtauritz@acm.org

ABSTRACT

Traditional evolutionary algorithms (EAs) are powerful problem solvers that have several fixed parameters which require tuning. An increasing body of evidence suggests that the optimal values of some, if not all, EA parameters change during the course of executing an evolutionary run. This paper investigates the potential benefits of dynamic parameters by applying a Meta-EA to evolving optimal dynamic parameter values for population size, offspring size, n in n-point crossover, Gaussian mutation's step size, bit flip mutation's mutation rate, parent selection tournament size, and survivor selection tournament size.

Each parameter was optimized both as the only dynamic parameter, and with all parameters dynamic. The most effective two parameters when acting independently were also allowed to optimize in tandem. The results were compared with a Meta-EA tuned EA using static parameters on the DTrap, NK, Rastrigin, and Rosenbrock benchmark problems. Results support that all tested parameters have the potential to improve solution fitness by changing dynamically, and using multiple dynamic parameters was more effective than using each independently.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Meta-EA, Dynamic Parameters, Parameter Control

1. INTRODUCTION

The already time consuming process of parameter tuning to receive optimal results in an Evolutionary Algorithm (EA) increases combinatorially when considering dynamically changing parameter values. A number of efforts have been made to determine parameter values for each stage of an EA. Mutation has received the most attention [6], but work has also been done in population size [1, 2], offspring size [4], recombination [3], parent selection [7] and survival selection [2]. Some work has also been done controlling multiple parameters at once [5, 8]. In most instances, the focus of parameter control research has been to remove or reduce an EA's dependency on a priori tuning, or on creating algorithms to modify or predict parameter settings, not necessarily to determine optimal settings. In [1], a hand-tuned EA using a static population size was compared to a Meta-EA evolved dynamic population size. While the dynamic population size obtained significantly better results, the difference in tuning methods and the usage of generations instead of evaluations for tuning makes the implications unclear.

2. METHODOLOGY

The following operators were chosen to be made dynamic: population size, offspring size, n in n-point crossover, Gaussian mutation's step size, bit flip mutation's mutation rate, parent selection tournament size, and survivor selection tournament size. These operators were selected because they use parameters that allow for straightforward dynamic modification and they are commonly used in EAs. Each meta-individual's genome represented *key* parameter values to use at evenly spaced intervals of evaluations. To determine the value of a parameter between keys, linear interpolation was used. By using a small number of *key*s, the Meta-EA can achieve a significant decrease in search space complexity without a significant loss in expressive power. To ensure fair comparison, a Meta-EA using a single *key* for each parameter was used to determine the best static configuration and all non-dynamic parameters employed these values.

The fitness of a meta-individual was determined by the average final best fitness of the set of EAs using that meta-individual. To compensate for meta-evolving six different parameter combinations on four benchmark problems, the number of Meta-EA runs and evaluations used were low, and each meta-individual was given 30 runs of an EA using a maximum of 5,000 evaluations. The DTrap, NK, Rastrigin and Rosenbrock benchmark functions were employed to test the effectiveness of dynamic parameters. They were chosen to represent the binary separable, binary inseparable, real-valued separable multimodal, and real-valued inseparable classes of problems, respectively.

Every parameter was tested using both two and five *key*s, to allow for experiments with low search space versus high control. As a higher number of *key*s can replicate fewer *key*s – by using identical key values sequentially – the worst fitness any number of *key*s should be able to obtain is equal to the best value found by a lower number of *key*s. As such, if the best dynamic parameter settings received worse results than static, the reason is due to search space complexity, because in an exhaustive search dynamic would at least find a way to mimic static. Additional testing was also performed using the two best *key* settings in tandem, as well as using two *key*s for every parameter at once.

3. RESULTS & DISCUSSION

The average and standard deviation of the best final fitness found by each configuration on each problem is given in Table 1. Experiments are labeled using the name of the parameter made dynamic and the number of *keys* used, with *Combined* referring to the combination of the best two. Paired T-Tests were used to compare all experiments with using the best found static configuration. To ensure the quality of the static configuration, the hand-tuned parameters found in [1] were compared with the Meta-EA tuned parameters found here, showing the Meta-EA's configuration to receive significantly better results in half as many evaluations.

On all four problems tested, using a single dynamic parameter was able to outperform the best found static configuration, in many cases to a statistically significant degree. The exceptions to statistical significance are NK and Rosenbrock, were the lack of time to converge is likely responsible for the high standard deviation. Despite the significantly larger search space, using two parameters together was able to outperform using either independently on DTrap, Rastrigin and Rosenbrock. Furthermore, allowing all parameters to change dynamically resulted in the best results of all experiments on Rosenbrock.

None of the experiments run resulted in final best meta-individuals who mimicked simpler dynamic settings. Also, when using dynamic parameters in tandem, the best *key* settings found for each was different than using the parameters alone. This implies that the best methods for parameter control are highly dynamic and highly interdependent. Finally, each parameter tested received the best or second best results on at least one of the problems, meaning that to achieve best performance, all parameters may need to be allowed to change dynamically.

Keys	Fitness	P-Value
4-Bit Concatenated DTrap, 100 Traps		
Combined	0.838 (0.013)	0
Parent Selection 5	0.837 (0.014)	0
Offspring 5	0.831 (0.012)	0
All	0.814 (0.008)	0.373
Static	0.813 (0.009)	1
NK Landscapes, N=20, K=3		
Population 2	0.766 (0.027)	0.325
Mutation Rate 5	0.765 (0.027)	0.383
Combined	0.764 (0.026)	0.464
Static	0.763 (0.024)	1.000
All	0.762 (0.026)	1.000
Rastrigin, N=20, A=10		
Combined	-8.278 (2.487)	0.002
Offspring 5	-8.816 (2.590)	0.011
Survival Selection 5	-9.538 (2.284)	0.063
Static	-10.769 (3.590)	1
All	-13.376 (4.471)	1
Rosenbrock, N=20, A=100		
All	-31.186 (31.716)	0.005
Combined	-39.435 (56.788)	0.023
Mutation Step Size 2	-55.181 (78.896)	0.138
Recombination 2	-63.263 (66.377)	0.217
Static	-79.545 (89.301)	1

Table 1: Fitness of dynamic parameters compared with static

4. CONCLUSION & FUTURE WORK

When using a Meta-EA to evolve parameter settings, a single dynamic parameter was found to be more effective than using strictly static parameters and all of the tested dynamic parameters were more effective than static on at least one problem. Using two dynamic parameters was better than using one dynamic parameter on all problems except NK, and on Rosenbrock the most effective configuration allowed all parameters to change dynamically together. The more a parameter changed, the better fitness it was able to achieve. The optimal control trends also changed depending on how other parameters were changing concurrently.

While running a lower number of evaluations should be indicative of longer runs, experiments using more common evaluation counts for the EA will allow for more direct interpretation. Since the minimum quality for any dynamic configuration should be to at least mimic the best less dynamic configuration, enough meta-evolution to reach this point should be performed.

To increase generality of results, problems involving permutation based solutions and real world problems should be attempted. Further experimentation, including parameters using more *key*s, can determine if there are any common dynamic parameter trends that can be used as the basis for control strategies.

5. REFERENCES

[1] J. Cook and D. Tauritz. An Exploration into Dynamic Population Sizing. In *Proceedings of GECCO 2010*. Portland, Oregon, USA, 2010.

[2] A. Eiben, M. Schut, and A. deWilde. Is Self-Adaptation of Selection Pressure and Population Size Possible? In *Proceedings of PPSN IX*, pages 900–909, 2006.

[3] J. Gomez. Self Adaptation of Operator Rates in Evolutionary Algorithms. In *Proceedings of GECCO 2010*, pages 162–173. Springer Berlin, Heidelberg, 2004.

[4] A. Nwamba and D. Tauritz. Futility-Based Offspring Sizing. In *Proceedings of GECCO 2009*, pages 1873–1874, 2009.

[5] G. Papa. Parameter-less Evolutionary Search. In *Proceedings of GECCO 2008*, pages 1133–1134. Atlanta, GA, USA, 2008.

[6] J. Smith. Modeling GAs with Self-Adaptive Mutation Rates. In *Proceedings of GECCO 2001*, pages 599–606, 2001.

[7] E. Smorodkina and D. Tauritz. Toward Automating EA Configuration: the Parent Selection Stage. In *Proceedings of IEEE CEC*, pages 63–70, 2007.

[8] F. Vafaee, W. Xiao, P. Nelson, and C. Zhou. Adaptively Evolving Probabilities of Genetic Operators. *Machine Learning and Applications*, pages 292–299, 2008.