Parameter Control: Strategy or Luck?

Giorgos Karafotias
VU University
Amsterdam, Netherlands
g.karafotias@vu.nl

Mark Hoogendoorn
VU University
Amsterdam, Netherlands
m.hoogendoorn@vu.nl

A.E. Eiben
VU University
Amsterdam, Netherlands
a.e.eiben@vu.nl

ABSTRACT

Parameter control mechanisms in evolutionary algorithms (EAs) dynamically change the values of the EA parameters during a run. Research over the last two decades has delivered ample examples where an EA using a parameter control mechanism outperforms its static version with fixed parameter values. However, very few have investigated why such parameter control approaches perform better. In principle, it could be the case that using different parameter values alone is already sufficient and EA performance can be improved without sophisticated control strategies. This paper investigates whether very simple random variation in parameter values during an evolutionary run can already provide improvements over static values.

Categories and Subject Descriptors

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

Keywords

Evolutionary algorithms; parameter control; methodology

1. INTRODUCTION

When setting up an evolutionary algorithm (EA) one needs to define the values for its parameters. Inappropriate values can degrade performance but the question whether a parameter value is appropriate is far from trivial as different phases in an evolutionary run could require different values. There are two options [3]: (1) find fixed parameter values that seem to work well across the entire run (*parameter tuning*), or (2) find a suitable control strategy to adjust the parameter during a run (*parameter control*). In literature, a variety of control strategies have been shown to outperform their static counterparts (e.g. [6] and [10]), and many have acknowledged that dynamically adjusting parameters is a very good idea (see e.g. [7]).

The motivation for this paper comes from perceiving control strategies as 'intelligent variations' applied to parameter values and observing that performance benefits are generally attributed to the 'intelligence' and not the 'variation' in itself. Some authors have made weak hints in this direction, see the next Section, however, none have performed a rigorous analysis. The goal of this paper is to investigate whether (non-intelligent) 'variation' alone is sufficient to improve EA performance. To this end, we implement a few simple random methods to vary parameter values during the run of an EA and investigate their impact on a set of benchmark problems.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

2. MOTIVATION AND RELATED WORK

It is widely accepted that parameter control is preferable over static parameters because different values are needed at different stages of the evolution. Also, information about the fitness landscape accumulated during the search can be used to improve parameter values in the later phases of the run [2].

In many of the studies introducing parameter control strategies, the controller's value is evaluated by comparing to the equivalent EA with static parameters. Usually no further investigation is carried out as to how exactly the parameters are varied and to what extent the performance gain is a result of the specific control strategy or the mere fact that parameters simply change. The idea that simply changing the values of a parameter, regardless of how, can result in better performance has been hinted in some previous work. In [9], Spears showed that a GA with random operator selection has similar performance with the self-adaptive GA meaning it is just the availability of multiple operators that improves performance and not self-adaptation. Randomized values are used in [4] to set the parameters of different islands for a distributed EA with the rationale that, at each moment during the search process, there will be at least one parameter configuration that will favor further advance.

Apart from the methodological issues raised by the possibility that mere variation of parameters can be beneficial, it is also a fundamental question; an inherent value in changing parameter values would provide a solid justification and motivation for research in parameter control. The only study we are aware of that directly addresses this question is found in [1], where deterministic schedules are used to investigate if there is an intrinsic advantage in having dynamic population sizes.

In this paper we attempt to answer the question whether variation of parameter values by itself (with no intelligence, purpose or strategy) can have a positive effect on the performance of an evolutionary algorithm. We believe that a theoretical approach would be impossible or oversimplifying and prefer an experimental approach as will be described in the following section.

3. EXPERIMENTS AND RESULTS

The purpose of the experiments is to determine if there can be an intrinsic merit in the mere variation of parameters (with no particular method/strategy) in terms of performance gain for the EA. To assess the effect of parameter variation isolated from the effect of an "intelligent" control strategy we use the most naive approach possible, i.e. random variation. Keeping all other factors identical, we compare the performance of an EA when its parameters are fixed during the whole search and when its parameters vary according to some random distribution. To show the difference between the random variation and a non-random (but certainly not sophisticated) variation approach, a sine function is used which facilitates sequences of increase and decrease of values.

We are not trying to establish as a general truth that parameter variation will by itself lead to better performance but rather to determine if it can be possible to observe better performance as a result of only the availability of multiple parameter values regardless of any control strategy. Furthermore, we do not propose random variation as a parameter control method.

We use a $(\mu + \lambda)$ Evolution Strategy with n-point crossover, gaussian mutation and tournament selection for both parents and survivors. Its parameters are the population size μ, the generation gap g, the number of crossover points n, the mutation step size σ and the tournament sizes k_p and k_s. We use a set of standard test functions: Ackley, Rastrigin, Rosenbrock, Schaffer, Bohachevsky, Griewangk and Shekel.

As a first step, the ES is tuned for every test function separately (all six parameters are tuned concurrently with one tuning process per problem). We use Bonesa [8], a state-of-the-art parameter tuner for real valued parameters. This step results in seven parameter vectors $\vec{p}_i$, one for each problem f_i.

In order to determine the effect of variation we add some random variation to the tuned vectors $\vec{p}_i$. At each generation, parameter values are drawn from a distribution (gaussian or uniform). For each parameter, a separate distribution is used; the "centers" of these distributions are the tuned values:

- *gaussian*: for problem i, values for parameter j are drawn from a normal distribution $N(\vec{p}_i(j), d \cdot \vec{p}_i(j))$
- *uniform*: for problem i, values for parameter j are drawn uniformly from $[\vec{p}_i(j) - \frac{w}{2}, \vec{p}_i(j) + \frac{w}{2}], w = d \cdot \vec{p}_i(j)$

Several width coefficients d are tried. Separate runs are made with each parameter varied alone and all parameters varied together. For 4 out of 7 problems and for 9 out of 49 combinations of problem and parameter, there exists some kind of variation that leads to significantly better performance.

For a more rigorous test, we use Bonesa to find good values for the standard deviation of the gaussian distribution. For problem i, values for parameter j are drawn from $N(\vec{p}_i(j), \sigma_i^j)$ with every σ_i^j derived through a search process by Bonesa (one tuning process per problem concurrently tuned the deviations of all six parameters). Due to time limitations, this experiment was performed only for the Ackley function. For all deviations the tuning process converged to values far from zero (except for g). This indicates that some variation is beneficial.

Finally, we make a *fair* comparison between the performance of the ES using static parameters and its performance using varying values. Since the static values were derived by tuning, the settings that determine the varying values must be calibrated as well. An identical tuning process (using Bonesa with the same budget of algorithm tests) is performed. Except for the normal and uniform random distributions, a sine wave able to generate sequences of increasing and/or decreasing values is used as well. For each variation mechanism, the following settings are tuned:

- *gaussian*: for each problem i and each parameter j, the mean m_i^j and standard deviation σ_i^j
- *uniform*: for each problem i and each parameter j, the minimum l_i^j and the width w_i^j of the range
- *sine*: for each problem i and each parameter j, the amplitude A_i^j, frequency f_i^j, angular frequency ω_i^j and phase ϕ_i^j

For 3 out of 7 problems naive variation yielded significantly better results. The sine wave was the winner for the Shekel function; in this case all parameters (except μ) are varied by a rapid oscillation, within a certain range, that also resembles random.

4. CONCLUSIONS AND FUTURE WORK

In this paper we put forward the assumption that random variation, without intelligence or strategy, can improve EA performance simply by making multiple parameter values available to the evolutionary process. To test this hypothesis we performed three experiments where the effect of randomly varying the parameter values was examined. Results showed that it is possible to improve the performance of an EA by randomly changing its parameters.

There are two implications of these findings. First, there is intrinsic gain in the variation of parameters and this provides a motivation for parameter control in general. Second, they raise an important issue in methodology. It has been common practice to evaluate a parameter controller by performing a comparison to the equivalent EA with static parameters. However, such a comparison does not necessarily show that the controller is good. A complete evaluation of a control mechanism should also include an analysis of how the parameters are varied during a run and a comparison to "naive" variation of the same parameters as a baseline benchmark.

Future work will focus on making a comparison between sophisticated parameter control approaches and the random variation approach presented in the experimental part of this paper to investigate the differences between the two in terms of performance.

5. REFERENCES

[1] J. Costa, R. Tavares, and A. Rosa. An experimental study on dynamic random variation of population size. In *Systems, Man, and Cybernetics, 1999. IEEE International Conference on*, volume 1, pages 607–612, 1999.

[2] K. A. De Jong. Parameter setting in EAs: a 30 year perspective. In Lobo et al. [5], pages 1–18.

[3] A. Eiben, Z. Michalewicz, M. Schoenauer, and J. Smith. Parameter control in evolutionary algorithms. In Lobo et al. [5], pages 19–46.

[4] Y. Gong and A. Fukunaga. Distributed island-model genetic algorithms using heterogeneous parameter settings. In *IEEE Congress on Evolutionary Computation*, pages 820–827, 2011.

[5] F. Lobo, C. Lima, and Z. Michalewicz, editors. *Parameter Setting in Evolutionary Algorithms*. Springer, 2007.

[6] B. McGinley, J. Maher, C. O'Riordan, and F. Morgan. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *Evolutionary Computation, IEEE Transactions on*, 15(5):692 –714, 2011.

[7] Z. Michalewicz and M. Schmidt. Parameter control in practice. In Lobo et al. [5], pages 277–294.

[8] S. Smit and A. E. Eiben. Multi-problem parameter tuning using bonesa. In J. Hao, P. Legrand, P. Collet, N. Monmarché, E. Lutton, and M. Schoenauer, editors, *Artificial Evolution*, pages 222–233, 2011.

[9] W. M. Spears. Adapting crossover in evolutionary algorithms. In *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 367–384. MIT Press, 1995.

[10] Y.-Y. Wong, K.-H. Lee, K.-S. Leung, and C.-W. Ho. A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 7:506–515, 2003.