

Ant Colony Optimization Beats Resampling on Noisy Functions

Tobias Friedrich

Timo Kötzing

Francesco Quinzan

Andrew M. Sutton

Hasso Plattner Institute
University of Potsdam
Potsdam, Germany

ABSTRACT

Despite the pervasiveness of noise in real-world optimization, there is little understanding of the interplay between the operators of randomized search heuristics and explicit noise-handling techniques such as statistical resampling. Ant Colony Optimization (ACO) algorithms are claimed to be particularly well-suited to dynamic and noisy problems, even without explicit noise-handling techniques.

In this work, we empirically investigate the trade-offs between resampling and the noise-handling abilities of ACO algorithms. Our main focus is to locate the point where resampling costs more than it is worth.

Keywords

Evolutionary Algorithm; Genetic Algorithm; Crossover; Estimation of Distribution Algorithm; Ant Colony Optimization; Robustness; Noise

1. INTRODUCTION

In many practical optimization problems, the objective function has some kind of stochastic component that arises out of different factors. In these scenarios, the direct evaluation of the objective function is not as reliable, and optimization algorithms must employ some kind of noise-handling strategy. The most common type of noise-handling strategy is statistical resampling. In this strategy, an algorithm estimates the true value of a function at a point by repeatedly sampling the value at that point to increase the signal to noise ratio. This approach comes at a computational cost, as the extra function evaluations must count toward the total run time of the algorithm.

A large number of papers [1, 2, 4–8] have observed that Ant Colony Optimization (ACO) algorithms are particularly well-suited for solving dynamic and noisy problems. ACO algorithms do not explicitly keep a population of solutions in memory, but instead construct a sequence of *pheromone values* that represents a probability distribution over the search

space. This approach appears to make them particularly robust in a changing, noisy environment [3].

ACO algorithms do not employ resampling directly, but instead rely on the robustness of the constructed distribution to somehow “filter” any noise. The impact of resampling and implicit distribution-building mechanisms of ACO algorithms is not clear. Obviously, resampling can gain a more accurate estimate of the true objective function, but at the cost of many extra function evaluations. The quality of the estimate depends on the number of samples and the underlying noise model. On the other hand, it is also not clear to what degree the distribution-building approach of ACO can help in the presence of noise, with or without using restarts.

2. CONTRIBUTION

In this work we take a first look at the interplay between statistical resampling and implicit noise-handling that arises from the cooperative distribution-building mechanisms of ACO algorithms. We empirically compare the performance of the $(\mu + 1)$ -EA (a mutation-only evolutionary algorithm), the $(\mu + 1)$ -GA (a steady-state genetic algorithm employing crossover) and λ -MMASib (an ant colony optimization algorithm). We investigate the trade-off between resampling and implicit noise-handling ability of each of these algorithms.

Our testbed is the fitness function $\text{OneMax} + \mathcal{N}(0, \sigma^2)$, which is a simple unimodal function with posterior noise. All algorithms have local parameters that influence their run time. We study the dependence of the parameter on the amount of noise σ and empirically determine for each algorithm the optimal parameter setting depending on σ .

We are then able to compare the algorithms with optimal parameter settings depending on the level of noise. We observe a strict hierarchy how well the algorithms can deal with noise (cf. Figures 1 and 2): from worst to best this is $(\mu + 1)$ -EA, $(\mu + 1)$ -GA, and λ -MMASib. The λ -MMASib scales most gracefully with increasing noise. When optimal parameters are chosen, we observe the empirical run time of all algorithms to have a polynomial, and we interpolate the degrees of these polynomials in Figure 2.

We study the optimal number of samples for a given noise level (cf. Figures 3–4). With optimal resampling we observe improved run times for the $(\mu + 1)$ -EA and $(\mu + 1)$ -GA, which scaled least gracefully with noise. However, even with optimal resampling both of them are still much worse than the λ -MMASib (cf. Figure 1).

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’16 Companion, July 20–24, 2016, Denver, CO, USA.

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4323-7/16/07.

DOI: <http://dx.doi.org/10.1145/2908961.2909039>

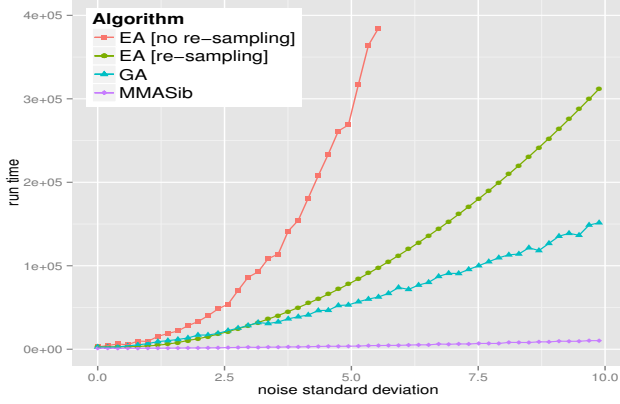


Figure 1: Number of fitness evaluations (run time) for $(\mu+1)$ -EA with optimal resampling, and $(\mu+1)$ -EA, $(\mu+1)$ -GA, λ -MMASib with no resampling, for a given posterior noise standard deviation.

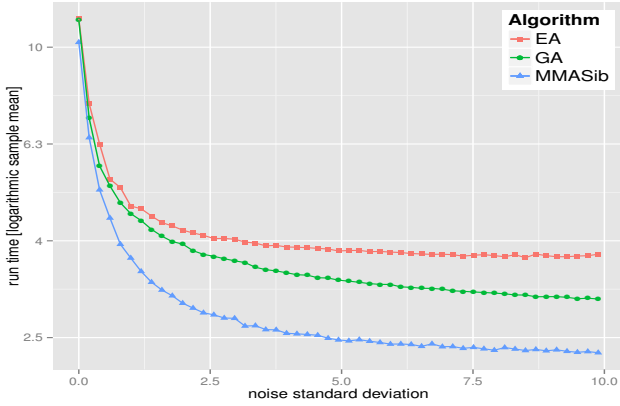


Figure 2: Interpolated polynomial degree of run times for $(\mu+1)$ -EA, $(\mu+1)$ -GA and λ -MMASib for a given posterior noise standard deviation.

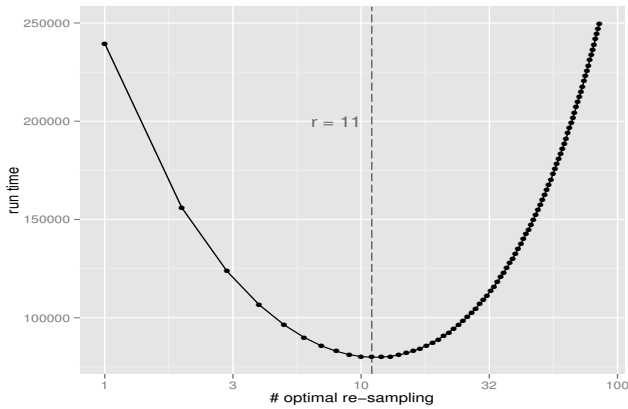


Figure 3: Number of fitness evaluations (run time) for the $(\mu+1)$ -EA, for a given number of resamples. Posterior noise standard deviation is fixed at $\sigma = 5.0$.

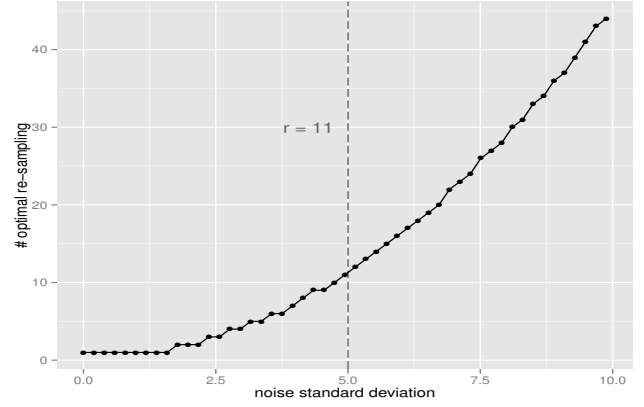


Figure 4: Trend of optimal r for a given posterior noise standard deviation. Note at $\sigma = 5.0$ we see minimum of Figure 3.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE). We would like to thank the anonymous reviewers for their helpful comments.

References

- [1] G. A. Di Caro, F. Ducatelle, and L. M. Gambardella. Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks. In *Reflecting interfaces: the complex coevolution of information technology ecosystems*, pages 185–216. 2008.
- [2] B. Doerr, A. Hota, and T. Kötzing. Ants easily solve stochastic shortest path problems. In *Proc. of GECCO '12*, pages 17–24, 2012.
- [3] M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851 – 871, 2000.
- [4] M. Feldmann and T. Kötzing. Optimizing expected path lengths with ant colony optimization using fitness proportional update. In *Proc. of FOGA '13*, pages 65–74, 2013.
- [5] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton. Robustness of ant colony optimization to noise. In *Proc. of GECCO '15*, pages 17–24, 2015.
- [6] T. Kötzing and H. Molter. ACO beats EA on a dynamic pseudo-boolean function. In *Proc. of PPSN'12*, pages 113–122, 2012.
- [7] A. Lissovoi and C. Witt. MMAS versus population-based EA on a family of dynamic fitness functions. *Algorithmica*, 2015. (To appear).
- [8] D. Sudholt and C. Thyssen. A simple ant colony optimizer for stochastic shortest path problems. *Algorithmica*, 64:643–672, 2012.