Noisy Combinatorial Optimisation by Evolutionary Algorithms

Aishwaryaprajna
School of Computer Science, University of Birmingham
Birmingham, United Kingdom
axx784@cs.bham.ac.uk

Jonathan E. Rowe
School of Computer Science, University of Birmingham
Birmingham, United Kingdom
j.e.rowe@cs.bham.ac.uk

ABSTRACT

We investigate the effectiveness of a set of evolutionary algorithms on noisy combinatorial optimisation problems. Despite some of these having polynomial runtime bounds for noisy OneMax, we find that in practice they are not able to solve this problem in reasonable time, with the exception of the Paired Crossover EA, and UMDA. We further study the performance of these two algorithms on noisy versions of SubsetSum and Knapsack.

CCS CONCEPTS

• **Computing methodologies → Search methodologies**; **Randomized search**; *Artificial intelligence*;

KEYWORDS

Noisy combinatorial optimisation, Gaussian noise, Expected runtime

ACM Reference Format:
Aishwaryaprajna and Jonathan E. Rowe. 2019. Noisy Combinatorial Optimisation by Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321955

1 INTRODUCTION

Optimisation in the presence of noise has received considerable treatment in the case of continuous optimisation, and a large number of heuristic approaches have been explored [5, 8]. However, until recently, there were fewer studies for combinatorial problems in the context of noise (see [2] for a survey).

In this paper, we are interested in whether any of the algorithms with good theoretical runtimes for noisy OneMax would be capable of solving combinatorial problems with added noise in practice. We proceed in two stages. First we will experimentally compare a collection of algorithms on noisy OneMax and noisy Linear problems, to see which can find solutions within a reasonable amount of time (to be defined below). Second, we will take those algorithms which pass this first test, and see how well they handle noise in two combinatorial problems: SubsetSum and Knapsack.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321955

Table 1: Noisy problems

Problem	Noisy variant
OneMax(x)	$\sum_{i=1}^{n} x_i + N(0,\sigma)$
WeightedLinear(x)	$x \cdot \mathbf{w} + N(0,\sigma)$
SubsetSum(x)	$\lvert\theta - x \cdot \mathbf{w}\rvert + N(0,\sigma)$
KnapsackV1(x)	$x \cdot \mathbf{p} + N(0,\sigma)$ if $x \cdot \mathbf{w} \le \theta$; $\theta - x \cdot \mathbf{w} + N(0,\sigma)$ otherwise.
KnapsackV2(x)	$x \cdot \mathbf{p} + N(0,\sigma)$ if $x \cdot \mathbf{w} + N(0,\sigma) < \theta$; $\theta - x \cdot \mathbf{w} + N(0,\sigma)$ otherwise.

Table 2: Algorithms studied

Algorithm	Parameter settings
(1+1)-EA	mutation rate $= 1/n$
Mutation-Population [3]	$\chi = 1, \lambda = \sigma^2 \log n$
Compact GA (cGA) [4]	$K = 7\sigma^2\sqrt{n}(\log n)^2$
PBIL [1]	$\lambda = 10n, \eta = 0.05$
UMDA [6]	$\lambda = 20\sqrt{n}\log n, \mu = \lambda/2$
PCEA [7]	$\mu = 10\sqrt{n}\log n$

2 EXPERIMENTAL METHOD

The problems studied are defined on a search space of bit strings of length n. The problems are described in Table 1. We let $N(0,\sigma)$ be a random number drawn from a normal distribution with mean zero, and standard deviation σ. We denote by $\mathbf{w}$, $\mathbf{p}$ and θ the weights, profits, and threshold, respectively. We investigate the performance of a range of evolutionary algorithms, as given in Table 2. The parameter settings given are in line with recommendations of what theoretical results we could find in the context of noisy optimisation.

3 RESULTS

The expected runtime of PCEA has lower theoretical bounds than the other algorithms, so for the first stage (looking at OneMax and WeightedLinear) we have allowed each algorithm to have twice the number of fitness evaluations PCEA requires to find the optimum. For these problems, we set $n = 100$. The results are depicted in Figure 1 and 2.

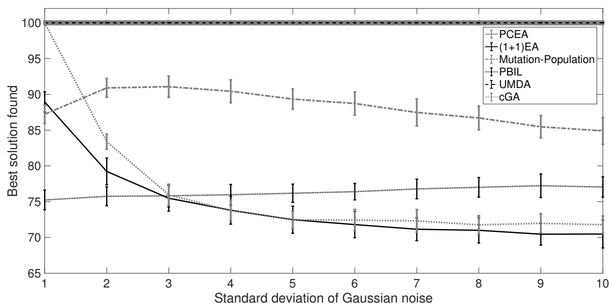

Figure 1: Comparison of the algorithms while solving the noisy OneMax for small noise levels

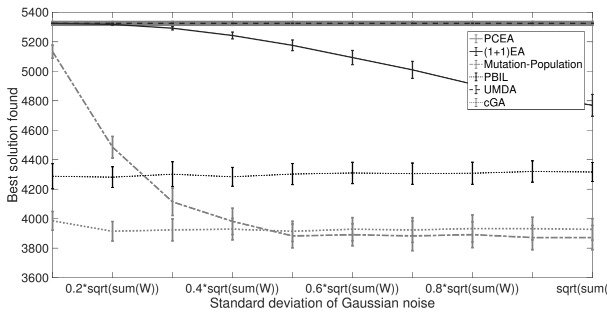

Figure 2: Comparison of the algorithms while solving the noisy WeightedLinear for small noises

Having determined that only UMDA and PCEA are effective at solving noisy linear problems, the second stage of the experiment is to compare their performance on the noisy versions of two combinatorial problems (SubsetSum and Knapsack). For these experiments the algorithms are run until the populations converge. The results are depicted in Figures 3, 4, and 5,

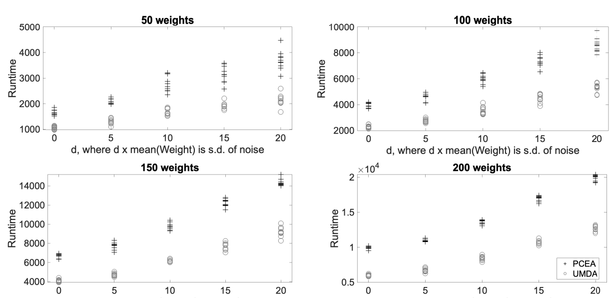

Figure 3: Comparison of runtime of UMDA (circles) and PCEA (crosses) while solving instances of the noisy SubsetSum problem.

4 CONCLUSIONS

We have empirically studied a range of evolutionary algorithms on a set of noisy problems. Both PCEA and UMDA handle noise well

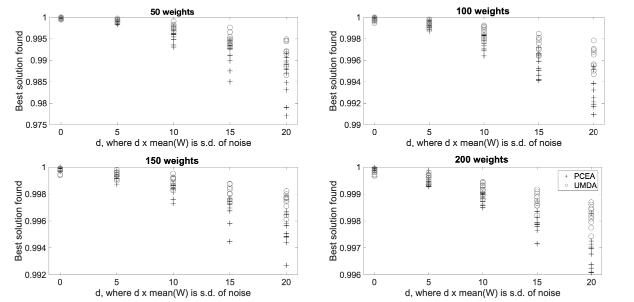

Figure 4: Solution quality of UMDA (in circles) and PCEA (in crosses) while solving instances of noisy KnapsackV1

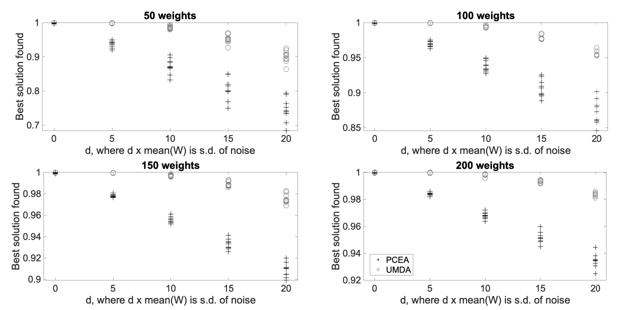

Figure 5: Solution quality of UMDA (circles) and PCEA (crosses) while solving the NoisyKnapsackV2

on both the simple test problems, and on the noisy combinatorial problems we have studied. The other algorithms have not been able to cope with noise, within the same function evaluation budget.

REFERENCES

[1] Shumeet Baluja. 1994. *Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning.* Technical Report. Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science.

[2] Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J Gutjahr. 2009. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing* 8, 2 (2009), 239–287.

[3] Duc-Cuong Dang and Per Kristian Lehre. 2015. Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*. ACM, 62–68.

[4] Georges R Harik, Fernando G Lobo, and David E Goldberg. 1999. The compact genetic algorithm. *IEEE transactions on evolutionary computation* 3, 4 (1999), 287–297.

[5] Yaochu Jin and Juergen Branke. 2005. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation* 9, 3 (2005), 303–317.

[6] Heinz Mühlenbein. 1997. The equation for response to selection and its use for prediction. *Evolutionary Computation* 5, 3 (1997), 303–346.

[7] Adam Prügel-Bennett, Jonathan Rowe, and Jonathan Shapiro. 2015. Run-Time Analysis of Population-Based Evolutionary Algorithm in Noisy Environments. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII (FOGA '15)*. ACM, New York, NY, USA, 69–75.

[8] Pratyusha Rakshit, Amit Konar, and Swagatam Das. 2017. Noisy evolutionary optimization algorithms – A comprehensive survey. *Swarm and Evolutionary Computation* 33 (2017), 18 – 45.