

# A Steady-State Genetic Algorithm with Resampling for Noisy Inventory Control\*

Steven Prestwich<sup>1</sup>, S. Armagan Tarim<sup>2</sup>, Roberto Rossi<sup>1</sup>, and Brahim Hnich<sup>3</sup>

<sup>1</sup> Cork Constraint Computation Centre, University College, Cork, Ireland  
s.prestwich@cs.ucc.ie, r.rossi@4c.ucc.ie

<sup>2</sup> Department of Management, Hacettepe University, Turkey  
armagan.tarim@hacettepe.edu.tr

<sup>3</sup> Faculty of Computer Science, Izmir University of Economics, Turkey  
brahim.hnich@ieu.edu.tr

**Abstract.** Noisy fitness functions occur in many practical applications of evolutionary computation. A standard technique for solving these problems is fitness resampling but this may be inefficient or need a large population, and combined with elitism it may overvalue chromosomes or reduce genetic diversity. We describe a simple new resampling technique called Greedy Average Sampling for steady-state genetic algorithms such as GENITOR. It requires an extra runtime parameter to be tuned, but does not need a large population or assumptions on noise distributions. In experiments on a well-known Inventory Control problem it performed a large number of samples on the best chromosomes yet only a small number on average, and was more effective than four other tested techniques.

## 1 Introduction

In many real-world applications of Genetic Algorithms (GAs) and other Evolutionary Computation algorithms, the fitness function is *noisy*: that is, the fitness of a chromosome cannot be computed directly but must be averaged over a number of samples. Examples include the learning of randomised games such as Backgammon, human-computer interaction, and simulation problems for which we wish to evolve a robust plan. The standard deviation of the sample mean of a random variable with standard deviation  $\sigma$  is  $\sigma/\sqrt{n}$  where  $n$  is the number of samples, so a large number of samples may be needed for very noisy fitness functions.

Several techniques for handling fitness noise in EAs are surveyed in [4,13]: the use of sampling to obtain an average fitness reduces noise; increasing the population size makes it harder for an unfit chromosome to displace a fitter one

---

\* S.A. Tarim and B. Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027. R. Rossi is supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain-Driven Research (CTVR) and Grant No. 05/IN/I886.

(a point also made by [10]) and can be viewed as a form of implicit averaging; and *rescaled mutation* samples distant points in the search space then moves a small distance toward them. [5] propose regression to estimate the fitness of neighbouring chromosomes. [1] vary sample rates across both chromosomes and generations in a generational GA. [18] record fitness levels in a search history, and use a stochastic model of fitness levels to locate new points in the search space. [3] use a threshold selection heuristic for accepting chromosomes. [17] adapt the sampling rate to different regions of the search space, a technique they call *dynamic resampling*. [19] use a Bayesian approach to sampling called Optimal Computing Budget Allocation, which assumes normally distributed noise.

A popular approach is to use a *Noisy Genetic Algorithm* (NGA) which computes the fitness of each chromosome by averaging over a number of samples [9,11,14,15]. Following [1] we shall refer to this as *static sampling*, and refer to this algorithm as NGAs. NGAs wastes considerable time evaluating unpromising chromosomes, but it can be improved by linearly increasing the number of samples with search time, starting from a low value [21,27]. We shall refer to this as *incremental sampling* and the resulting algorithm as NGAi. However, though NGAs and NGAi have been used to solve real problems, they may not be the most efficient approach. It is pointed out in [22] that a reduction in noise is not necessary for *every* chromosome, only for the *best* ones. Of course, this entails discovering which are the best chromosomes without performing a large number of samples, but poor chromosomes might become apparent after just a few samples.

An alternative technique is to *resample* chromosome fitness: that is, some chromosomes are allowed to survive for more than one generation, and their fitness is periodically recomputed to refine the estimate. Various heuristics may be used to decide when to discard a chromosome. [22] experiments with averaging over a small number of samples, and guiding resampling by a statistical test which assumes Gaussian noise but is considered to be robust under non-Gaussian noise. [12] uses the standard deviation of the fitness to correct for its noise, again under assumptions on noise distribution. Resampling and the common heuristic of *elitism* do not always combine well. [6] show that, with an elitist GA, the probabilistic method of [12] is inferior to a resampling approach. [2] show that, in Evolutionary Strategies that allow fitness values to survive for more than one generation, failure to resample can lead to systematic overvaluation of chromosomes. [8] found that, when applying co-evolutionary learning to the noisy task of learning how to play Backgammon, more sampling can have a bad effect on the learning besides incurring overhead. It causes less fit chromosomes to be pruned more quickly which reduces genetic diversity too drastically, especially with small populations. Despite these drawbacks, resampling and elitism have been successfully combined. [25] describe an extension of the Simple (generational) GA that maintains a list of the fittest solutions found so far, while increasing the number of samples as search proceeds as in NGAi; they also increase the population size during search.

Another successful resampling elitist GA is the *Kalman-extended Genetic Algorithm* (KGA) [23], designed for problems whose fitness is both noisy and