

# 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

## GAs with noisy fitness

In many real-world GA applications the fitness function is *noisy*: it can't be computed directly but must be averaged over a number of samples. For example learning randomised games such as Backgammon, human-computer interaction, and finding a robust plan in simulation problems. The standard deviation of sample mean of  $n$  samples of a random variable is  $\sigma/\sqrt{n}$ , so many samples may be needed for very noisy fitness functions.

Several techniques for handling fitness noise in EAs have been reported:

- increasing population size to make it harder for an unfit chromosome to displace a fitter one
- *rescaled mutation*: sampling distant points in the search space then moving a small distance toward them
- regression to estimate the fitness of neighbouring chromosomes
- varying sample rates across both chromosomes and generations in a generational GA
- recording fitness levels in a search history, and using a stochastic model of fitness levels to locate new points in the search space
- using a threshold selection heuristic for accepting chromosomes

- *dynamic resampling*: adapting the sampling rate to different regions of the search space
- *Optimal Computing Budget Allocation*: a Bayesian sampling method
- *Noisy GA*: average fitness over a fixed number of samples (**static sampling**)
- Noisy GA but linearly increasing the number during the run (**incremental sampling**)
- *resampling*: allow some chromosomes to survive, and periodically refine their fitness estimates; aims to save effort by sampling only the more promising chromosomes
- *Kalman-extended Genetic Algorithm*: steady-state GA with resampling using 1 sample each time (**Kalman sampling**)

## **This work**

We aim to find a simple resampling strategy that:

- can be used with a steady-state GA such as GENITOR
- does not assume any noise properties and can handle high noise
- does not require a large population
- resamples fit chromosomes many times to avoid overvaluation (high **reliability**)
- on average uses only a few samples per chromosome (low **wastefulness**)

As a test-bed we use a well-known Stochastic Inventory Control problem: the *replenishment cycle policy*  $(R^n, S^n)$  with stationary demands, fixed delivery cost, linear holding and stockout costs, 100 time periods. Model “no order” as a special allele NULL.

## The algorithm

We use basic GENITOR with random parent selection, and standard uniform crossover applied with a crossover probability 0.5. Population size is 30. Mutation: in our inventory problem, each gene can take any of 100 integer values or NULL; we select NULL with probability 0.5, otherwise randomly select one of the 100 values. We assume that at least  $U$  samples are required to obtain a sufficiently reliable fitness estimate. We use a large value:  $U = 1000$ .

Pseudo-code for our resampling GENITOR, called GASGA:

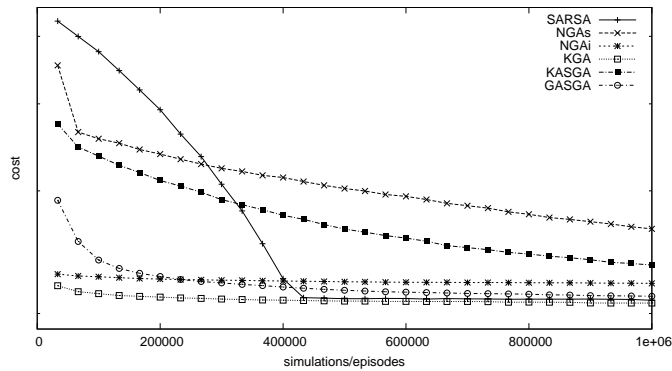
```
GASGA( $S, P, U$ )
  create population of size  $P$ 
  evaluate population using  $S$  samples
  while not(termination condition)
    select two parents
    breed one offspring  $O$ 
    evaluate  $O$  using  $S$  samples
    replace least-fit chromosome by  $O$ 
    select fittest chromosome  $F$  with #samples  $< U$ 
    re-evaluate  $F$  using  $S$  samples
  output fittest chromosome
```

## Results

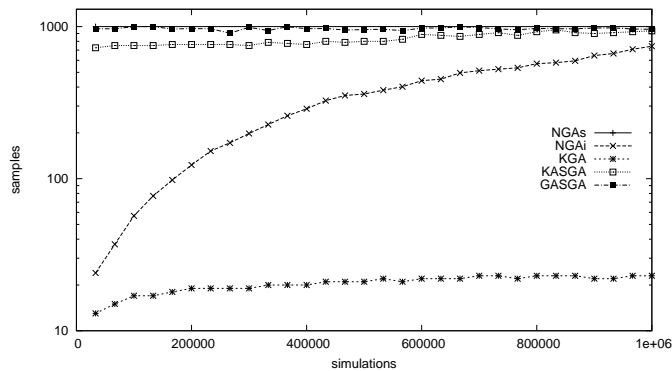
We compare five strategies, plus SARSA( $\lambda$ ) (the problem can be modelled as a POMDP with state=period, action=order level or NULL, reward=-cost):

- static sampling
- incremental sampling  $U$  during the run
- Kalman sampling
- Kalman sampling averaged over  $S$  samples
- GASGA: our main contribution

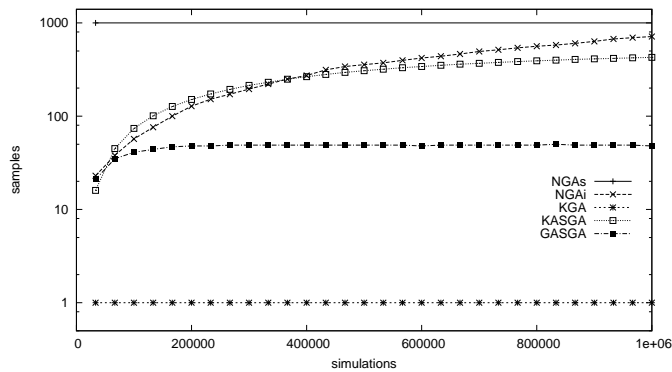
### fitness:



### reliability:



### wastefulness:



GASGA is a simple heuristic approach to the problem of resampling very noisy fitness functions. It has high fitness and reliability, and low wastefulness; all other tested GAs are inferior in at least 1 dimension. SARSA( $\lambda$ ) has high fitness, but reliability and wastefulness are undefined.

We found even better results by hybridising a GA with SARSA:

S. D. Prestwich, S. A. Tarim, R. Rossi, B. Hnich.

A Cultural Algorithm for POMDPs from Stochastic Inventory Control.

5th International Workshop on Hybrid Metaheuristics (to appear).