# Compact Genetic Algorithm

## and more

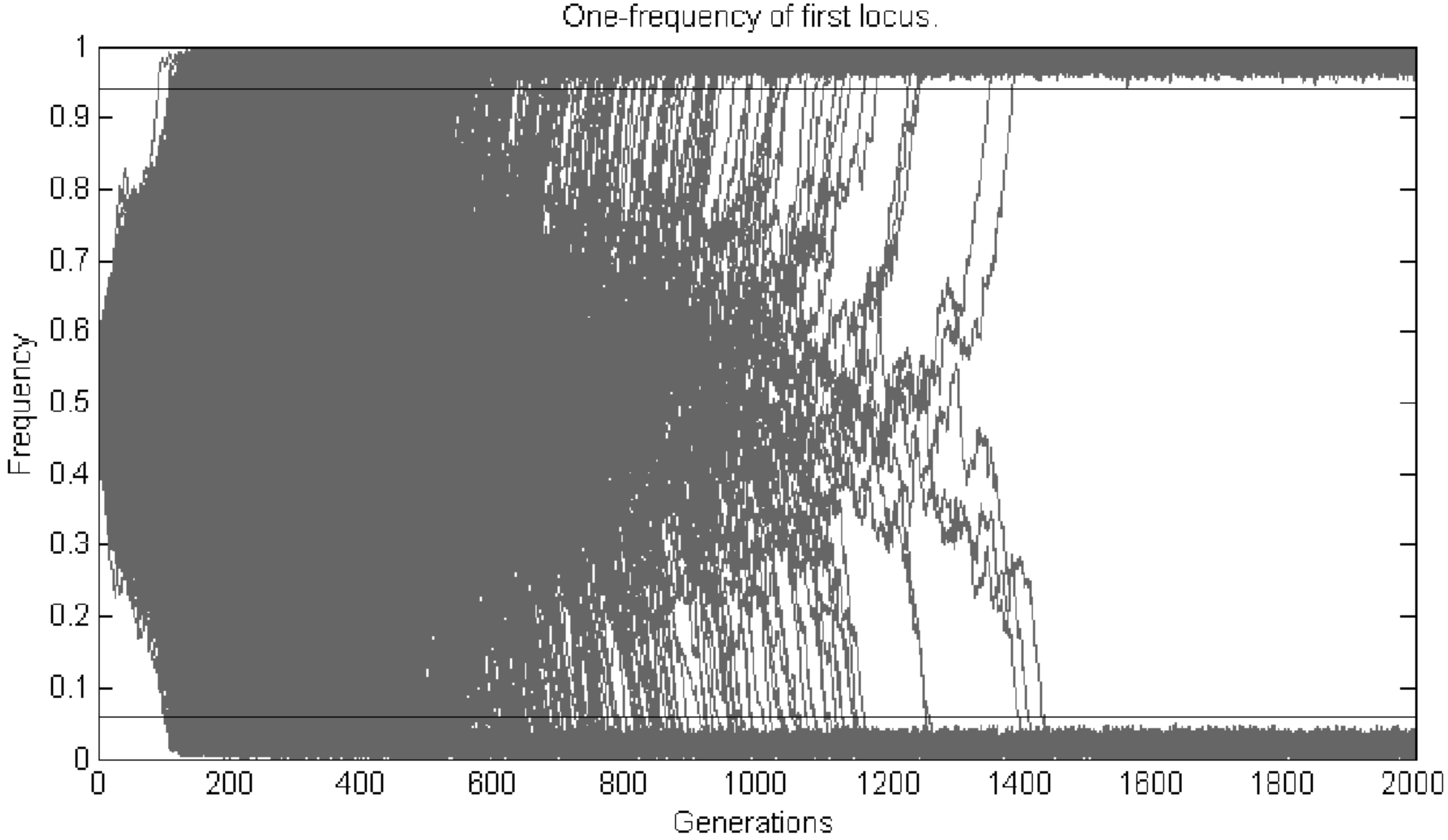
BY ŁUKASZ STAFINIAK

# Using Gene Fixation to Learn About Epistasis

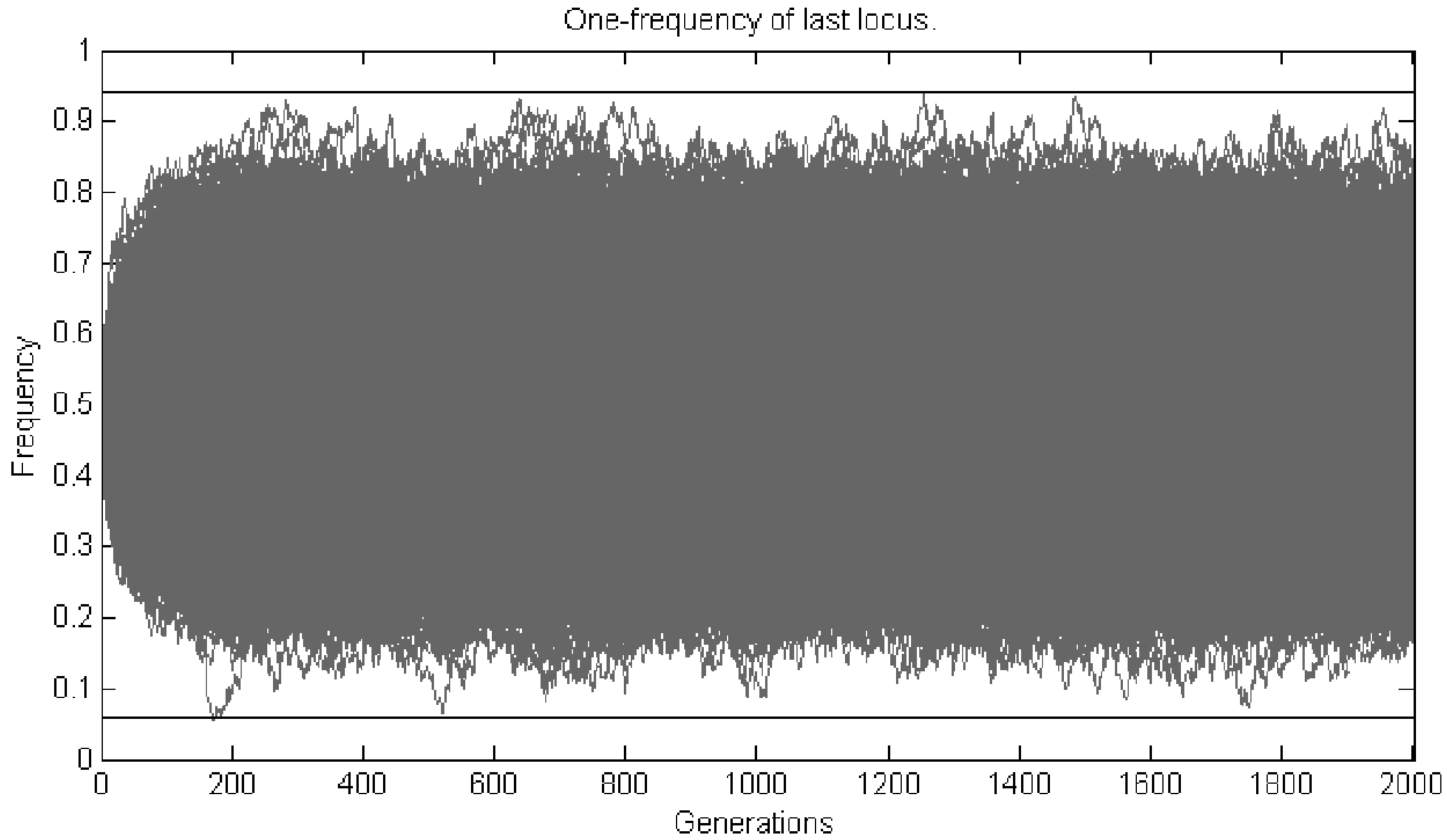
Say we have a stochastic fitness function returning $\mathcal{N}(\delta, \ \sigma)$ if $x_{i_1} \oplus \ldots \oplus x_{i_k} = 1$, and $\mathcal{N}(-\delta, \sigma)$ otherwise ( $\oplus$– exclusive or). We can use the SGA to find the set $\{i_1, \ldots, i_k\}$, with efficiency linear in the number of genes $l$. Compare this with the "naive" $O(l^k)$ approach of approximating marginal distributions (we need to check at least $k$-tuples).

[After Keki Burjorjee]

Frequency of an epistatic gene $x_{i_a}, a \in \{1,4\}$:



## Frequency of a noncoding gene $x_a, a \notin \{i_1, ..., i_4\}$:



# Compact Genetic Algorithm

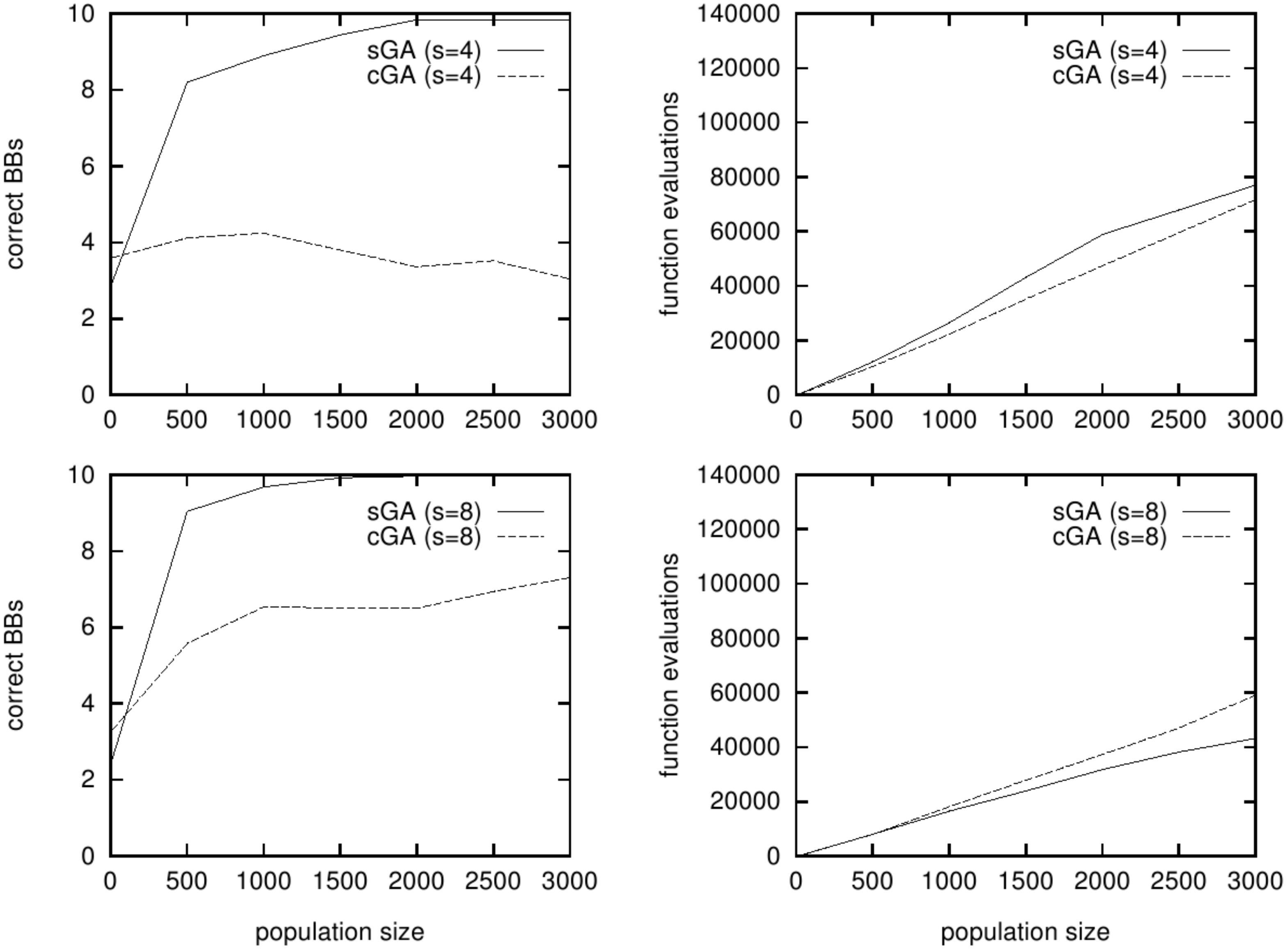
- Basic cGA with step $\frac{1}{n}$ is equivalent to (U)SGA with population size $n$, steady state, tournament size 2 selection, for "simple" problems.

- This cGA consumes only $\log_2 n$ bits per gene, compared to $n$ bits of SGA.

- Tournament size $k$ cGA: generate $k$ individuals, and then perform $k-1$ updates ("duels") against the best one.

- Under uniform crossover, order-$k$ BB has survival probability $2^{1-k}$. For cGA, $P(\text{survival of } H) = \prod_{i \in H} p_i$, starting with $p = \frac{1}{2}$ this gives $2^{-k}$.

- In order to protect BBs, we need to strenghten selectional pressure. For survival prob. $p$ we need tournament size $\frac{1}{p}$, two times worse for cGA. (And it still turns out worse.)



10 times 3-trap fitness function, $2^{1-k} = \frac{1}{4}$.

# Extended Compact Genetic Algorithm

- The goal is linkage learning, or building block identification.
- SSGA with uniform crossover is roughly equivalent to CGA, which is an order-1 probabilistic optimization algorithm, only remembering marginal probabilities for variables = genes; CGA works under assumption of independence of the variables.
- SGA with one-point crossover [often] works better because programmers tend to code related genes close together (partial problem-specific linkage knowledge).[sometimes worse - slower]
- The choice of a good distribution is equivalent to linkage learning.
- Extensions of CGA first used order-2 methods: two-variable dependency approximations of population distribution; exact estimation would just resample the population.

- ECGA uses Occam razor not to overfit the distribution estimation to a particular population; it uses Minimum Description Length principle: the model size+the population compression ratio resulting from a distribution.

- ECGA's class of distributions are Marginal Product Models: assuming independence of **disjoint** groups of variables and giving full (marginal) distribution within a group; the MDL criterion computes

  - model complexity $= \log N \sum_i 2^{|s_i|}$

  - compressed population complexity $= N \sum_i H(X_{s_i})$

  where $H(X_{s_i}) = H(p(X_{s_i}))$ is the entropy of a group $s_i$ of variables

- ECGA uses greedy search by trying to merge each pair of groups starting with singletons; the MDL criterion detects dependencies between groups

- selection introduces dependencies across boundaries of groups of MPM from which the population was generated

- with enough time for population analysis, ECGA can be extended from MPM to bayesian network learning

# Sources

- *Two Remarkable Computational Competencies of The Simple Genetic Algorithm* Keki Burjorjee
- *The Compact Genetic Algorithm* Georges R. Harik, Fernando G. Lobo, David E. Goldberg
- *Linkage Learning via Probabilistic Modeling in the ECGA* Georges Harik