Combining Competent Crossover and Mutation Operators: A Probabilistic Model Building Approach

Cláudio Lima[1], Kumara Sastry[2], David Goldberg[2], and Fernando Lobo[1]

[1]Department of Electronics and Computer Science Engineering
University of Algarve

[2]Illinois Genetic Algorithm Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign

Overview

- Motivation
- Background
  - Crossover vs Mutation
  - Extended Compact GA
  - BB-wise Mutation Algorithm
- Probabilistic Model-Building Hybrid GA
- Experiments
- Conclusions
- Extensions

Motivation

- EDAs outperform simple GA
- EDAs solve decomposable problems within a low-order polynomial number of function evaluations
- However, for large-scale problems, even a low-order polynomial number of FE can be very demanding…

Motivation

- Efficiency-enhancement techniques
  - **Hybridization**
  - Time Continuation
  - Parallelization
  - Evaluation-Relaxation
- Use Model information to perform local search => **Building-Block-wise Local Search**
- A more general Hybridization

Background

- Crossover versus Mutation
- Extended Compact Genetic Algorithm
- Building-Block-wise Mutation Algorithm

Crossover *vs* Mutation

- Assumption of knowledge of the problem decomposition (Sastry and Goldberg, 2004)
- For deterministic decomposable problems:
  - Mutation outperforms Crossover
  - Speedup of O(sqrt(k)*log(m))
- For decomposable problems with additive Gaussian noise:
  - Crossover outperforms Mutation
  - Speedup of O(sqrt(k)*m / log(m))

Extended Compact GA

- Estimation of distribution ≡ Linkage learning
- Estimate the distribution of the population using Marginal Product Models (MPMs)
  - For a **linear** problem: $[x_1][x_2][x_3][x_4][x_5][x_6][x_7][x_8][x_9][x_{10}][x_{11}][x_{12}]$
  - For an **order-3** additively decomposable problem: $[x_1x_2x_3][x_4x_5x_6][x_7x_8x_9][x_{10}x_{11}x_{12}]$
  - For an **order-4** additively decomposable problem: $[x_1x_2x_3x_4][x_5x_6x_7x_8][x_9x_{10}x_{11}x_{12}]$

Which Model is Better?

- Metric: Minimum Description Length (MDL)
  - Preference for **accurate** and **simpler** models

- Search for the model that minimizes the storage needed to represent the population
  - Overall Complexity =

Model Complexity + Population Compression Complexity

$$C_m = \log_2(n+1)\sum_{i=1}^{m}(2^{k_i}-1)$$

$$C_p = n\sum_{i=1}^{m}\sum_{j=1}^{2^{k_i}} -p_{ij}\log_2(p_{ij})$$

Model Building

- In each generation, after selection, perform a greedy search for the best MPM
- Start with the **simplest model**: $[x_1][x_2][x_3]...[x_{l-1}][x_l]$
- Consider **all possible merges of two subsets** and choose the one that leads to a **lower overall complexity**
- Stop when no further improvement is possible

eCGA steps

Extended Compact Genetic Algorithm **(eCGA)**

(1) Create a random population of n individuals.

(2) Evaluate all individuals in the population.

(3) Apply s-wise tournament selection [5].

(4) Model the selected individuals using a greedy MPM search procedure.

(5) Generate a new population according to the MPM found in step 4.

(6) If stopping criteria is not satisfied, return to step 2.

BB-wise Mutation Algorithm

- Induces good neighborhoods as linkage groups (Sastry & Goldberg, 2004)
- Use linkage learning procedures developed for selectorecombinative GAs
- Mutation: Bit-wise => Building-Block-wise
- Search: Hillclimbing => Deterministic or Random

BB Neighborhood

Consider a order-3 decomposable problem with 4 subfunctions (BB partitions):

BB1	BB2	BB3	BB4
101	011	**001**	110

Solution A → 101 011 **001** 110

Neighborhood of Solution A within BB partition #3 →

101 011 **010** 110
101 011 **011** 110
101 011 **100** 110
101 011 **101** 110
101 011 **110** 110
101 011 **111** 110
101 011 **000** 110

How can we get the best BB in partition #3? Evaluate all the neighborhood and choose the best individual

Extended Compact Mutation Algorithm

Extended Compact Mutation Algorithm **(eCMA)**

(1) Create a random population of n individuals and evaluate their fitness.

(2) Apply s-wise tournament selection [5].

(3) Model the selected individuals using a greedy MPM search procedure.

(4) Choose the best individual of the population for BB-wise mutation.

(5) For each detected BB partition:

- (5.1) Create $2^k - 1$ unique individuals with all possible schemata in the current BB partition. Note that the rest of the individual remains the same and equal to the best solution found so far.
- (5.2) Evaluate all $2^k - 1$ individuals and retain the best for mutation in the other BB partitions.

A Probabilistic Model Building Hybrid GA

- The probabilistic model of eCGA is used for two distinct purposes:
  1. Effective recombination of BBs that provide rapid global-search capabilities
  2. Effective search in the BB neighborhod that locally provides high-quality solutions
- The key idea is to obtain the benefits from both approaches

Joining Both...

- Learn the model
- Perform BB-wise local search on the best individual of the population
- Update the model parameters (frequencies) according with the BB-wise mutated individual
  - **Increase** the BB instances frequencies of the **mutated individual** by **s**
  - **Decrease** the BB instances frequencies of the **previous best solution** by **s**

Updating the Model

Model found: $[x_1x_2x_3][x_4x_5x_6][x_7x_8][x_9]$

Best individual of the population 000 011 01 0
Mutated version of the best individual 111 111 00 0

$p(x_1x_2x_3)$		$p(x_4x_5x_6)$		$p(x_7x_8)$		$p(x_9)$	
...	...	...	...	...	...	...	...
000	0.11 – s/n	**011**	0.18 – s/n	**01**	0.23 – s/n	**0**	0.44
...	...	...	...	...	...	...	...
111	0.03 + s/n	**111**	0.02 + s/n	**00**	0.31 + s/n	...	...
...	...	...	...	...	...	...	...

Hybrid eCGA

***Hybrid Extended Compact Genetic Algorithm* (heCGA)**

(1) Create a random population of n individuals.

(2) Evaluate all individuals in the population.

(3) Apply s-wise tournament selection [5].

(4) Model the selected individuals using a greedy MPM search procedure.

(5) Apply BB-wise mutation to the best individual.

(6) Update the frequencies of the MPM found on step 4 according to the BBs instances present on the mutated individual:

- (6.1) Increase the BB instances frequencies of the mutated individual by s.
- (6.2) Decrease the BB instances frequencies of the previous best individual by s.

(7) Generate a new population according to the updated MPM.

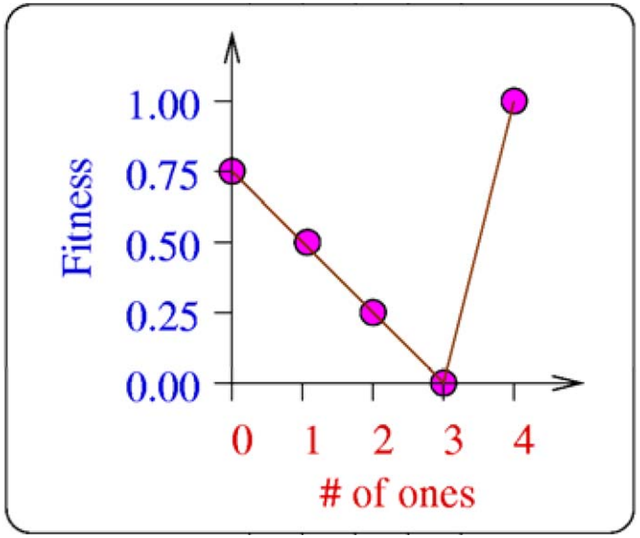
(8) If stopping criteria is not satisfied, return to step 2.

Experiments

- **m** concatenated deceptive trap functions of **order-k**

$$f_{trap}(u) = \begin{cases} 1 & \text{if } u = k \\ 1 - d - u * \frac{1-d}{k-1} & \text{otherwise} \end{cases}$$

u - # of 1s in the string
k – size of the trap function
d – fitness signal



For our experiments, k=4 and d=1/k=0.25

Experiments

- Problem 1: Deception

$$f_d(X) = \sum_{i=0}^{m-1} f_{trap}(x_{ki}, x_{ki+1}, \ldots, x_{ki+k-1})$$

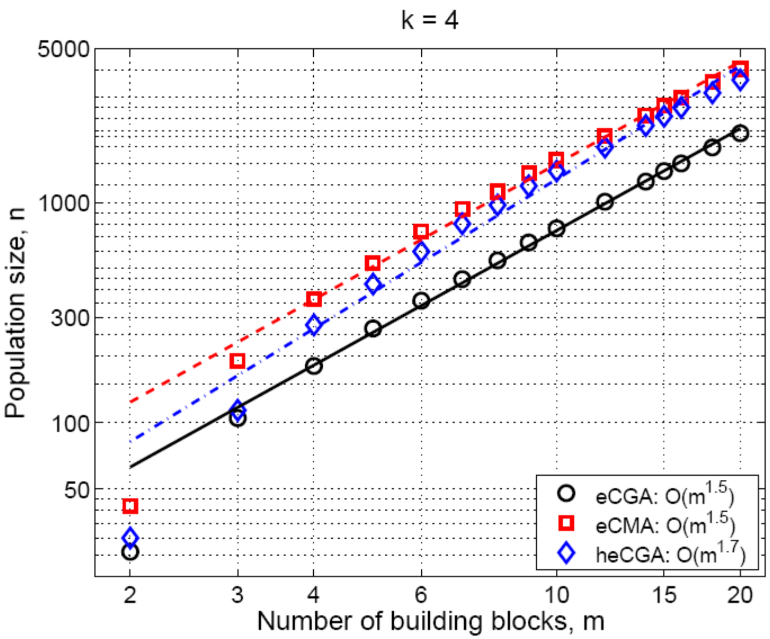
- Problem 2: Deception + Scaling

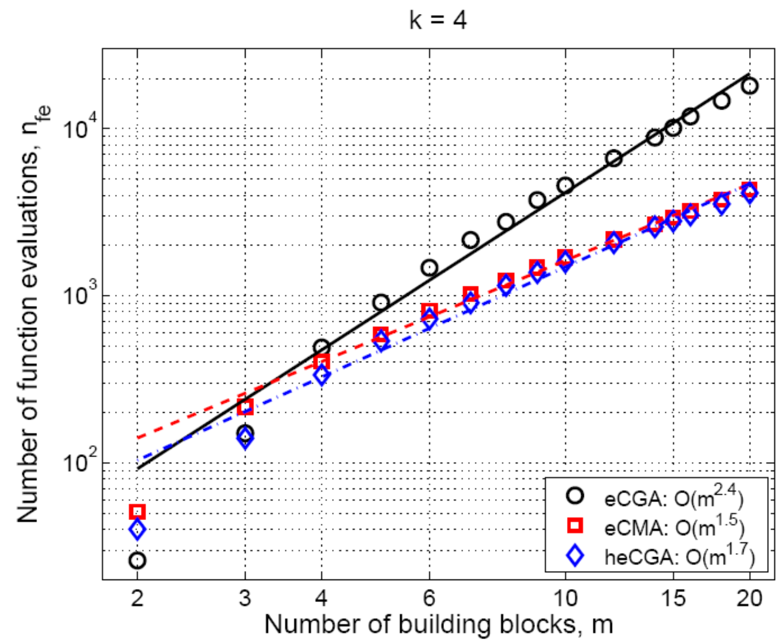
$$f_{ds}(X) = \sum_{i=0}^{m-1} 2^i f_{trap}(x_{ki}, x_{ki+1}, \ldots, x_{ki+k-1})$$

- Problem 3: Deception + Noise

$$f_{dn}(X) = f_d(X) + \mathrm{G}(0, \sigma_N^2)$$

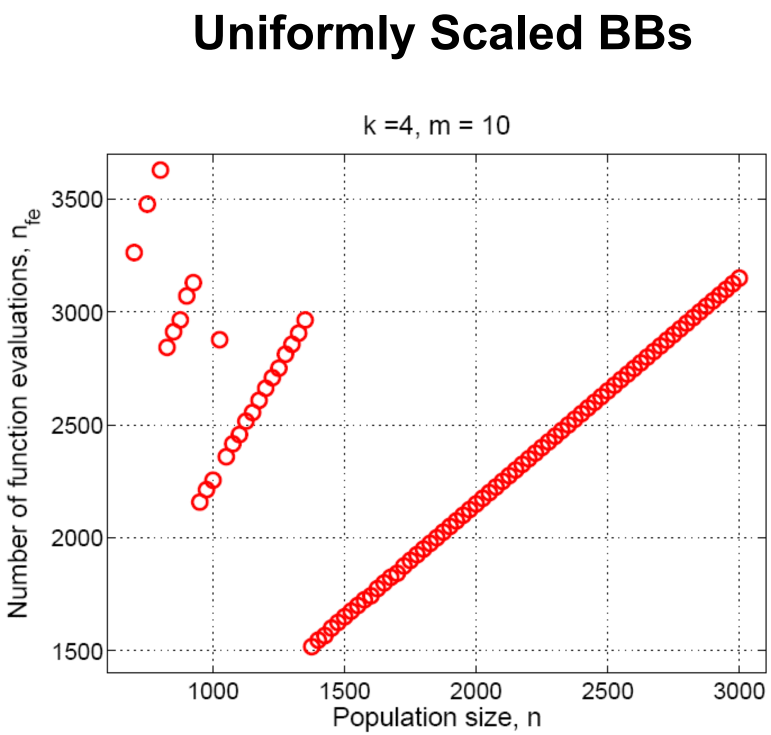
Uniformly Scaled BBs





k = 4

Behaviour of Hybrid eCGA



- **More Flexible**: can solve the problem within a bigger range of pop size than eCMA
- 4 lines, 4 different hitting times (# gen.)

Uniformly Scaled BBs

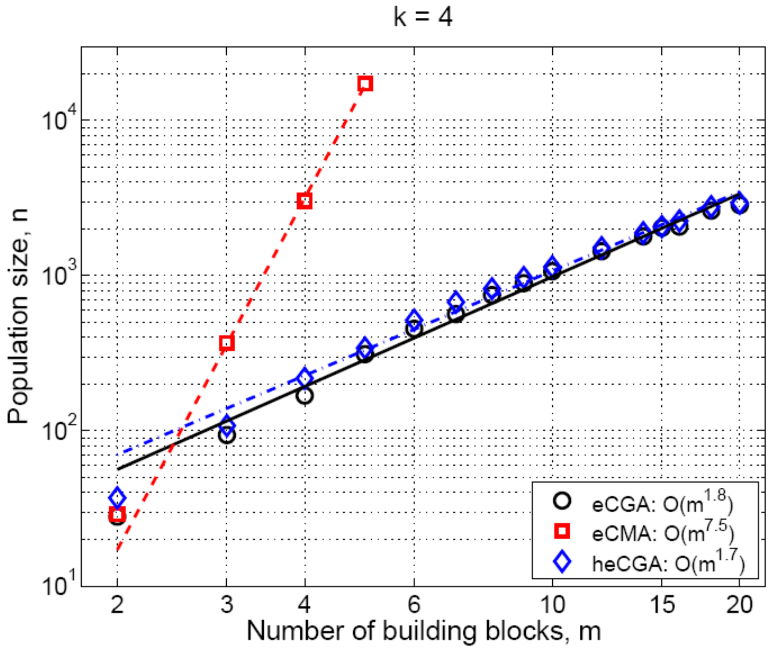
- eCGA needs smaller populations, but takes more FEs than eCMA and heCGA

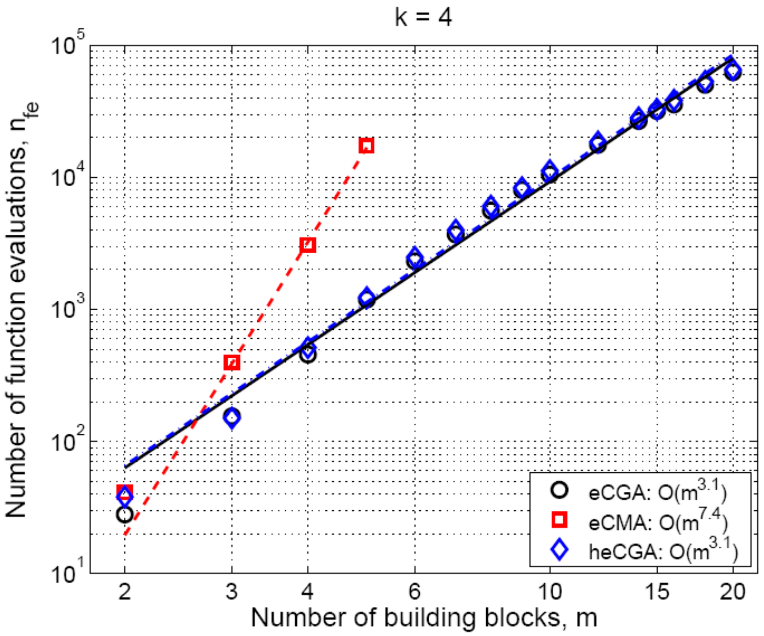
  1. BBs discovery in a progressive way
  2. Mixing time

- eCMA scales better than eCGA

- heCGA behaves similarly to eCMA

Exponentially Scaled BBs

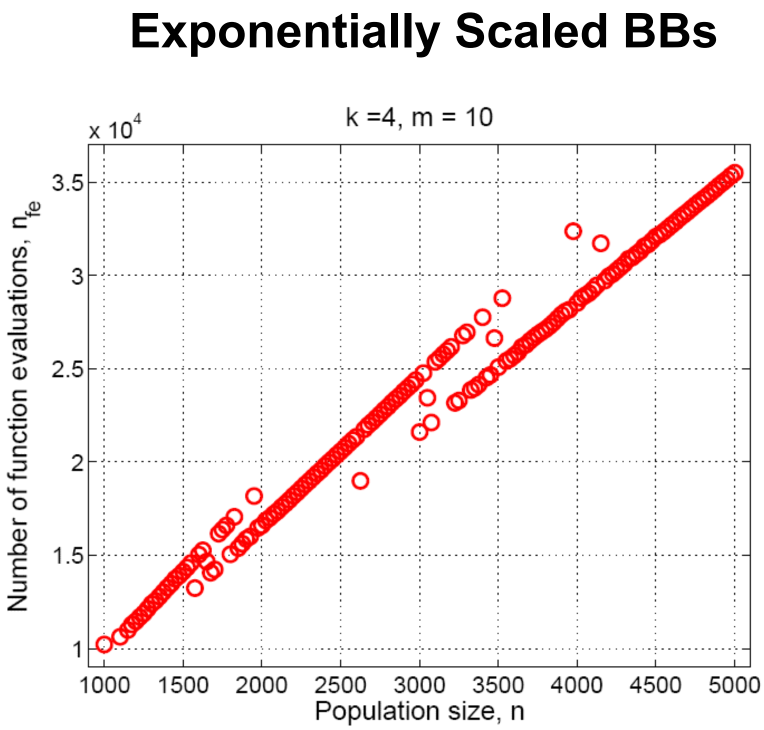




Exponentially Scaled BBs

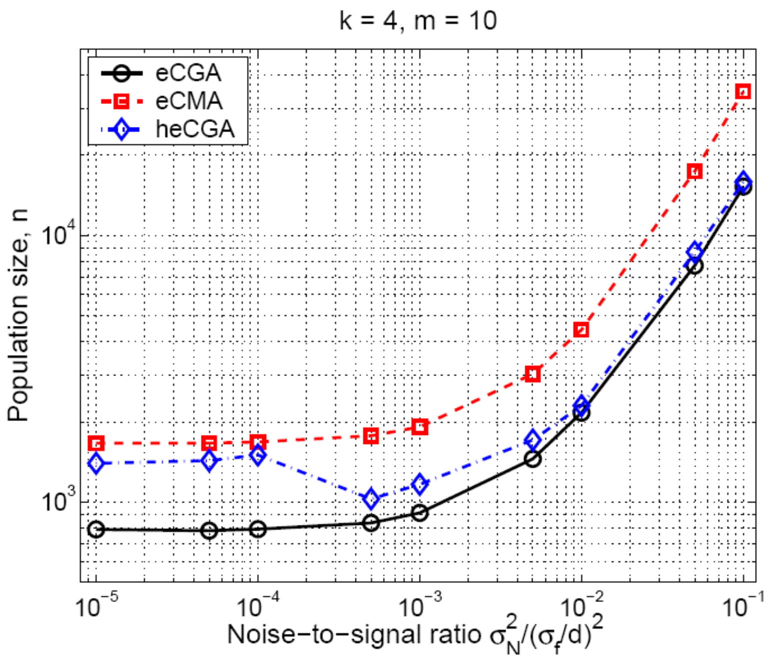
- eCMA needs exponentially pop sizes and NFEs
- Hybrid eCGA performs similar to regular eCGA
- Hybrid eCGA changed his behaviour

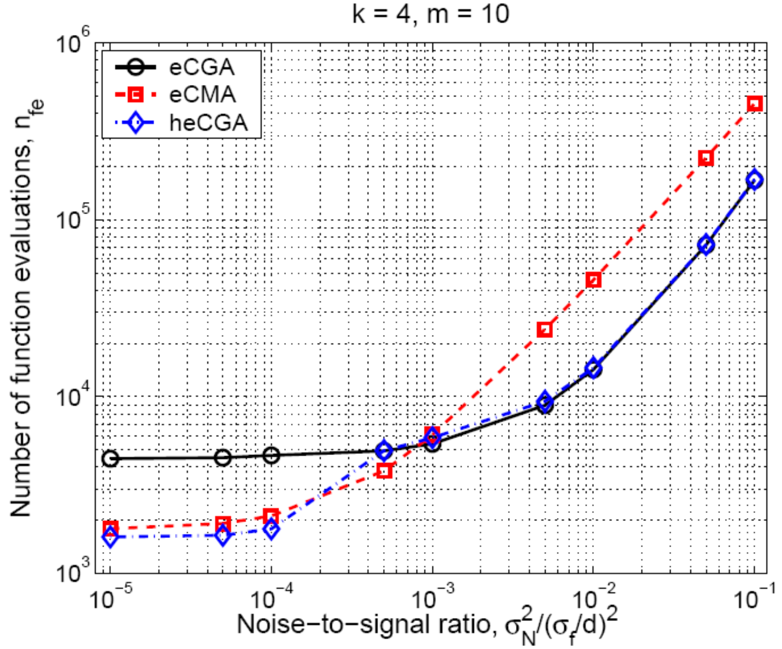
Behaviour of Hybrid eCGA



- Number of FE grows almost linearly with pop size
- Increasing pop size will not reveal much more BB information

BBs with additive Gaussian Noise





BBs with additive Gaussian Noise

- Trasition phase on Hybrid eCGA behaviour
- For small values of noise, similar case to the deterministic function:
  - Hybrid eCGA do similar to eCMA, which is the best
- For larger values of noise:
  - Hybrid eCGA do similar to eCGA, which is the best

Summary and Conclusions

- Integration of BB-wise mutation on eCGA
- Probabilistic Model Building Hybrid GA allows a more general hybridization
- In the absence of domain knowledge, the hybrid eCGA is more robust than either single-operator-based approach

Extensions

- Other hybridization configurations
- Combination with other enhancement techniques for EDAs
- Problems with overlapping BBs
- Application to real-world problems

Acknowledgements

- GECCO reviewers for helpful comments and suggestions

- FCT/MCES under grants POSI/SRI/42065/2001 and SFRH/BD/16980/2004, AFOSR/USAF under grants F49620-00-0163 and F49620-03-1-0129, NSF under grant DMI-9908252, and ITR grants DMR-99-76550 and DMR-0121695.