

Model-based EAs

Fernando Lobo

Model based EAs

- Learn a model from a population of selected (good) solutions.
- Use the model to generate new solutions.
- Combine ideas of Machine Learning with Evolutionary Computation.

Model based EAs

- First model-based EAs were based on learning probabilistic models.
- More recently non-probabilistic models are also being used.

EDAs / PMBGAs

- Two different names:
 - PMBGAs → Probabilistic Model Building Genetic Algorithms.
 - EDAs → Estimation of Distribution Algorithms
- The term EDA is used more often.

EDAs / PMBGAs

- Don't use explicit variation operators such as crossover and mutation.
- Instead, learn a probabilistic model from the population.
- and use the model to generate new solutions.

Different kind of models

- Some models are very simple.
 - Good for solving simple problems
- Other models are more complex.
 - Good for solving more complex problems.

Today ...

- We look at the simple GA from a different perspective
- Present two EDAs
 - compact GA (uses a simple prob. model)
 - extended compact GA (uses a more complex model)

A different view on the SGA

- Initially there's roughly half ones and half zeros in the population.
- Selection gives preference to better individuals.
- Uniform crossover decorrelates bit positions.

During a GA run...

- For every bit position:
 - alleles compete with each other
 - at the end, one of them wins

1	0	0	0
1	1	0	1
0	1	1	1
1	1	0	0
0	0	1	0
0	1	1	1
1	0	0	0
1	0	0	1

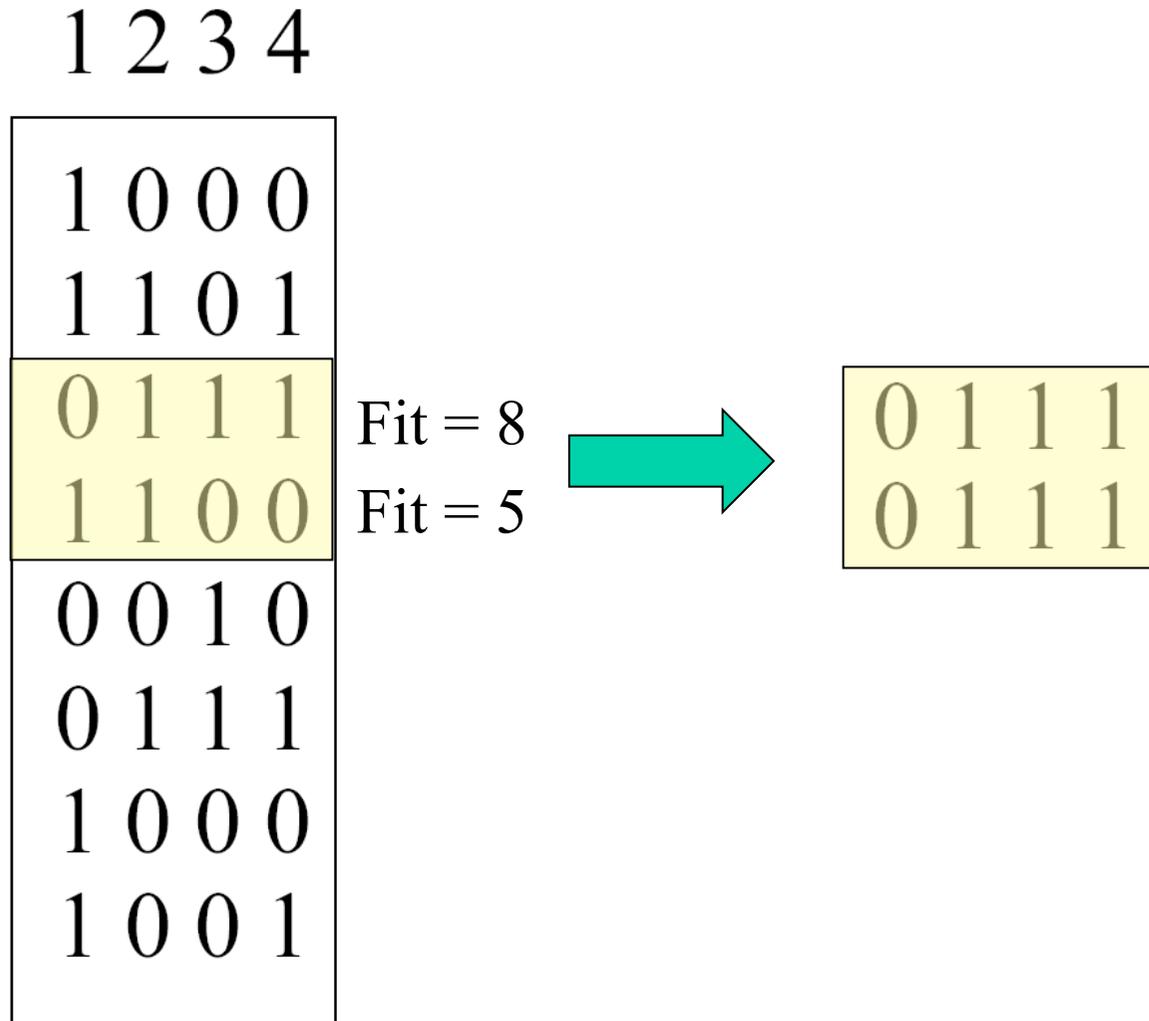
Selection

- Selection gives preferences to better individuals.
- ... but not always do so for better alleles.

Block selection

- Two individuals compete.
- The winner replaces the loser.
- Similar to tournament size 2.

Example



What happens to the allele frequencies?

- Winning alleles increase by $1/\text{popsize}$.
- Losing alleles decrease by $1/\text{popsize}$.
- Ties remain the same.

Example

1	2	3	4
1	0	0	0
1	1	0	1
0	1	1	1
1	1	0	0
0	0	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Fit = 8

Fit = 5



0	1	1	1
0	1	1	1

0s increase by $1/8$ on position 1

1s increase by $1/8$ on positions 3 and 4

Compact GA

(Harik, Lobo, Goldberg, 1997)

- Population is represented by a probability vector of size L
- Each element of the vector denotes the frequency of 1's in a GA's population for the corresponding bit position.

Simple GA

1 2 3 4

1	0	0	0
1	1	0	1
0	1	1	1
1	1	0	0
0	0	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Compact GA

1 2 3 4

$5/8$	$4/8$	$3/8$	$4/8$
-------	-------	-------	-------

Compact GA

- Vector is initialized with 0.5 at every position.
- Individuals are sampled from the vector.
- Vector is updated towards the winning alleles by $+ 1/N$

Compact GA

- Each update represents a small step in the action of the GA.
- Inspiration comes from the Gambler's Ruin Model for population sizing (Harik et al., 1996).

Compact GA

- Compact GA is equivalent to Simple GA with uniform crossover.
- But population can be represented with less memory.
 - cGA: $L * \log_2(N+1)$ bits
 - sGA: $L * N$ bits

Related algorithms

- PBIL (Baluja, 1994)
- UMDA (Mühlenbein & Paß, 1996)

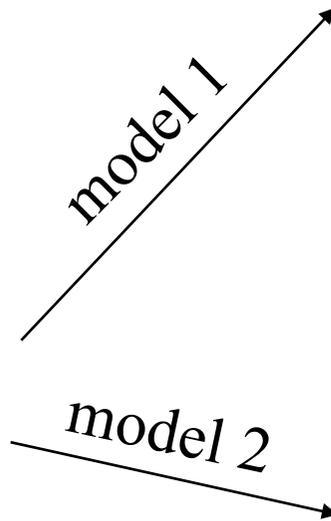
Extended Compact GA

(Harik, 1999)

- It's possible to have probabilistic models that capture dependencies among bit positions.

Example: 1 population, 2 models

	1	2	3	4
1	1	0	0	0
2	1	1	0	1
3	0	1	1	1
4	1	1	0	0
5	0	0	1	0
6	0	1	1	1
7	1	0	0	0
8	1	0	0	1



{1}	{2}	{3}	{4}
5/8	4/8	3/8	4/8

{1,3}		{2}	{4}
11	0	1	4/8
10	5/8	0	4/8
01	3/8		
00	0		

Which model is better?

- ECGA uses the MDL principle to choose the better one.

Minimum Description Length principle

- Use the model that minimizes the storage needed to represent the population.
- Includes storage of the model + storage of the population according to the model
- Simplest explanation is best
 - (simplest means shortest description length)

Model Complexity: $\log_2(N + 1) \sum_i (2^{S_i} - 1)$

Population's
complexity
according to model: $N \sum_i Entropy(M_i)$

S_i : size of the i^{th} subset of genes

M_i : marginal distribution of the i^{th} subset of genes

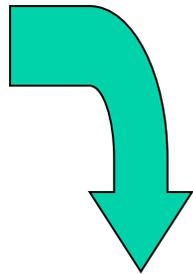
Entropy

- Entropy tells the randomness of a distribution.
- Can be used to detect patterns in data.
- Can be interpreted as the average number of bits needed to represent an element at random from that distribution:

$$\sum -p_i \log_2(p_i)$$

Example:

Entropy of the marginal
distribution {1,3}



	{1,3}	{2}	{4}
11	0	1 4/8	1 4/8
10	5/8	0 4/8	0 4/8
01	3/8		
00	0		

$$-(5/8) \log_2(5/8) - (3/8) \log_2(3/8) \approx 0.954$$

Means that an element drawn at random from this distribution needs on average 0.954 bits (rather than 2 bits if all outcomes were equally likely)

Example:

	{1,3}	{2}	{4}
11	0	1 4/8	1 4/8
10	5/8	0 4/8	0 4/8
01	3/8		
00	0		

Model: [1,3][2][4]

$$\text{Model Complexity} = \log_2(9)(3 + 1 + 1) = 15.8$$

$$\text{Compressed Population Complexity} = 8(0.954 + 1 + 1) = 23.6$$

$$\text{Combined Complexity} = 15.8 + 23.6 = 39.4$$

Model building

- At every generation, the ECGA does a greedy search for a model that compresses the population the most.
- Starts with the univariate model.
- Merges all possible subsets and chooses the one that leads to a lower combined complexity.
- Stops when it's not possible to improve any more.

Example:

Start with model [1][2][3][4]

Combined complexity = 44 bits

Merge all possible pairs of subsets...

MPM	Combined Complexity
[1, 2][3][4]	46.7
[1, 3][2][4]	39.8
[1, 4][2][3]	46.7
[1][2, 3][4]	46.7
[1][2, 4][3]	45.6
[1][2][3, 4]	46.7

Best model so far

[1,3][2][4] with 39.8 bits

Again, merge all possible pairs of subsets...

MPM	Combined Complexity
[1, 3, 2][4]	48.6
[1, 3, 4][2]	48.6
[1, 3][2, 4]	41.4

Cannot improve, stop model search.

Use [1,3][2][4]

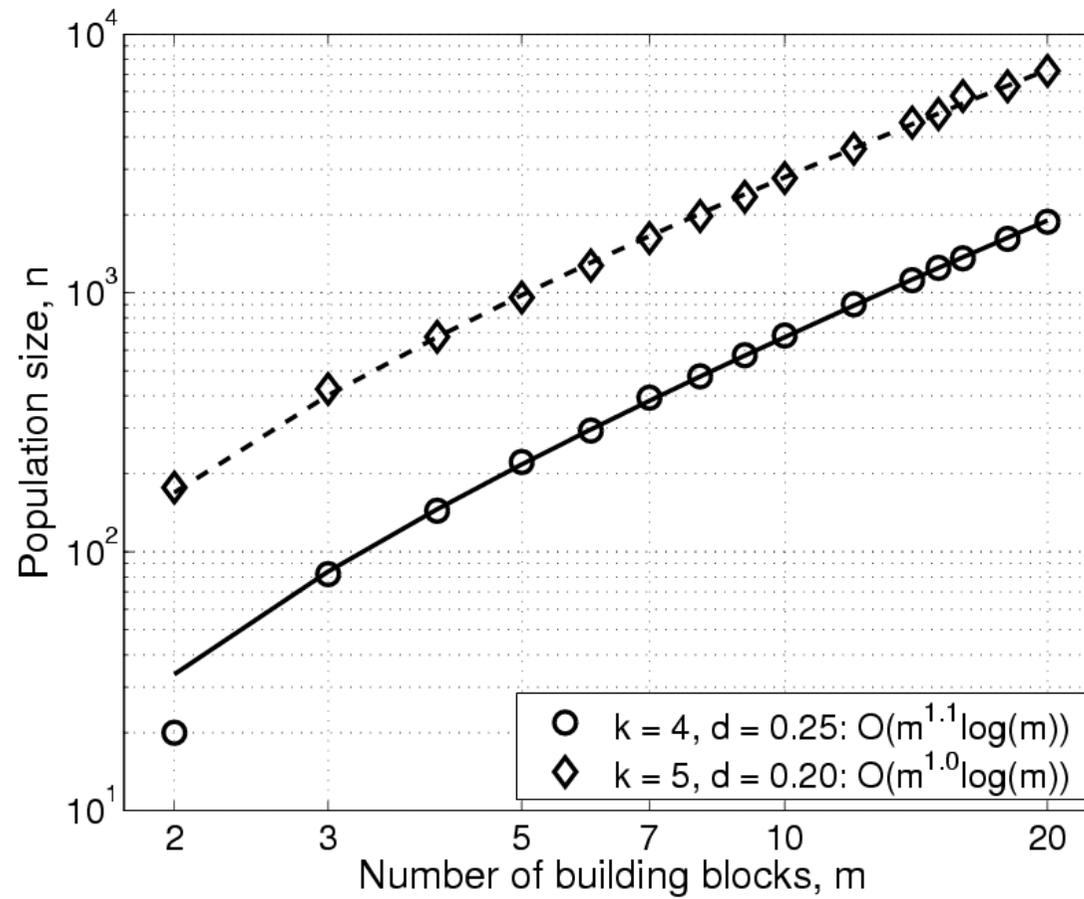
Learning a good model equals learning linkage

- For the next generation new individuals are sampled from the learned model.
- The gene partition corresponds to linkage groups.

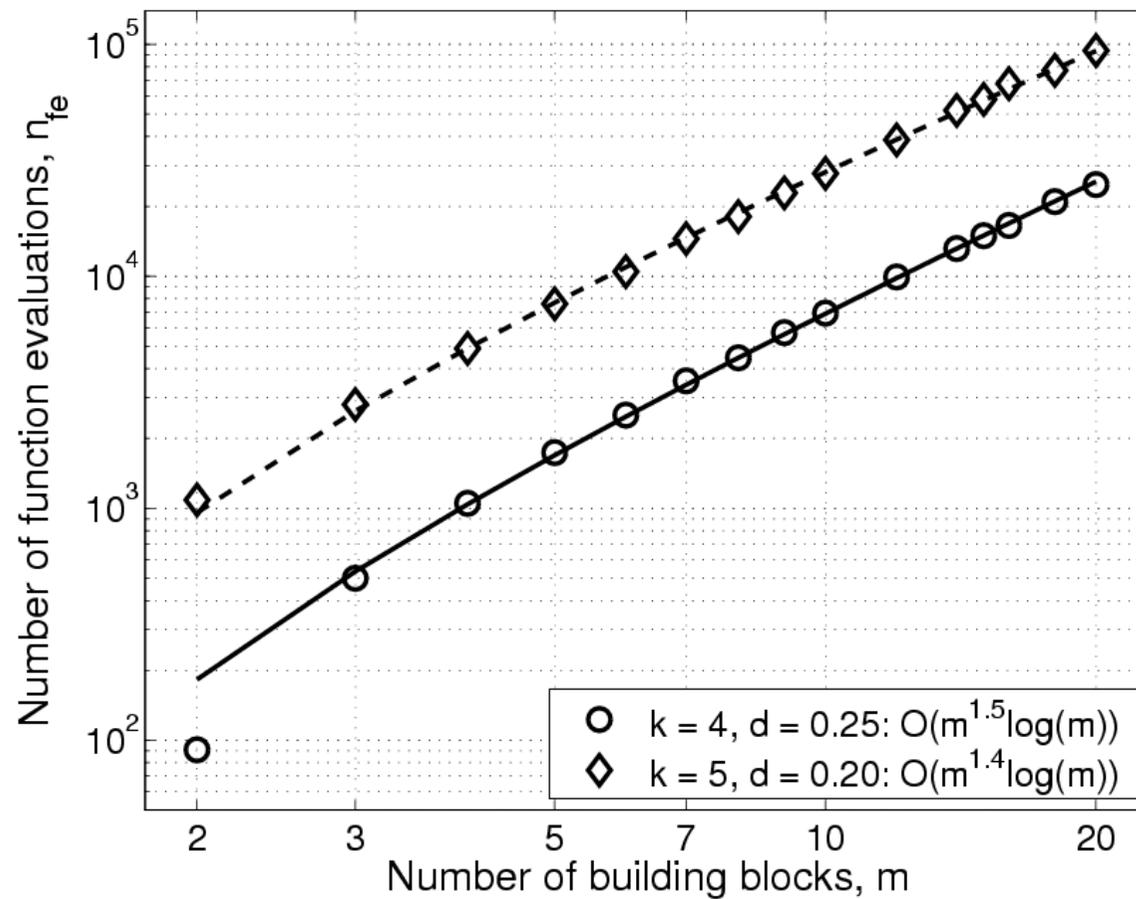
Scalability

- Solves problems of bounded difficulty within a low-order polynomial number of fitness function evaluations.
- Without apriori linkage information.

Scalability (popsize)



Scalability (func. evals.)



Other EDAs

- There are more sophisticated EDAs using more powerful probabilistic models such as Bayesian Networks.
 - Bayesian Optimization Algorithm (Pelikan, Goldberg & Cantú-Paz, 1999).

Other model-based EAs

- There are also non-probabilistic model-based EAs that are very effective at learning linkage.
 - Linkage Tree Genetic Algorithm (Thierens, 2010).
 - Parameter-less Population Pyramid (Goldman & Punch, 2014)