

# Efficiency Enhancement In Estimation of Distribution Algorithms

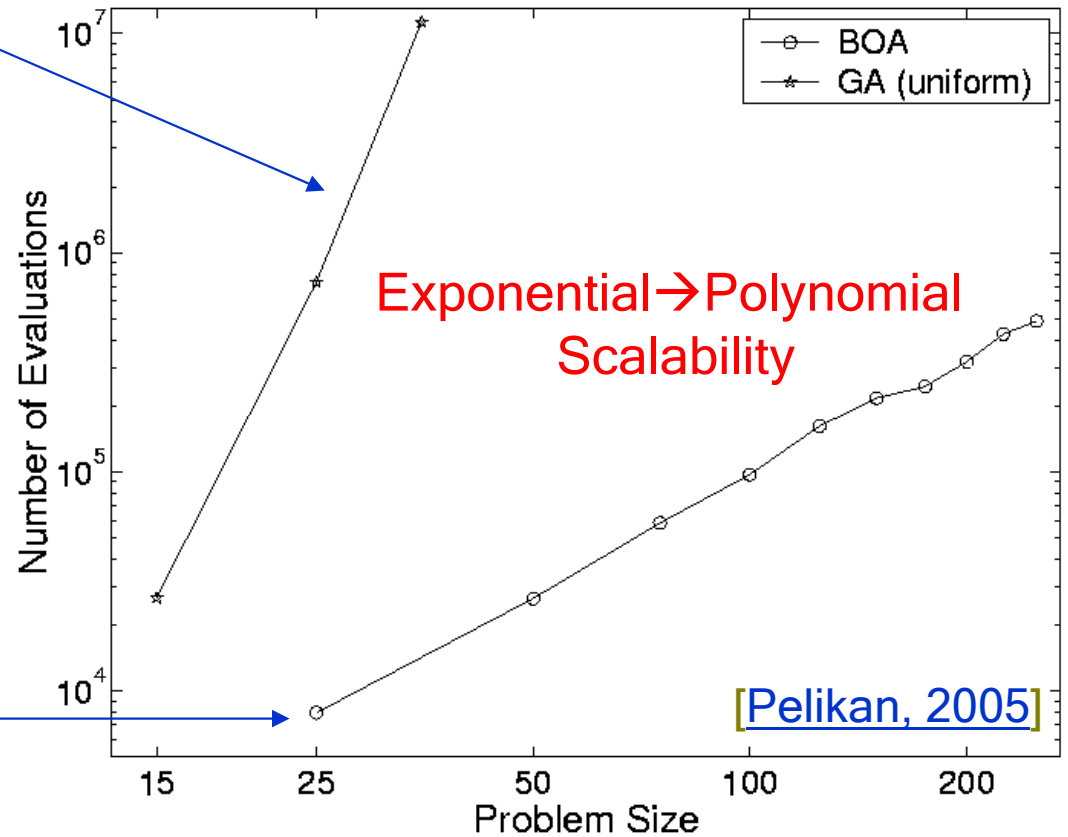
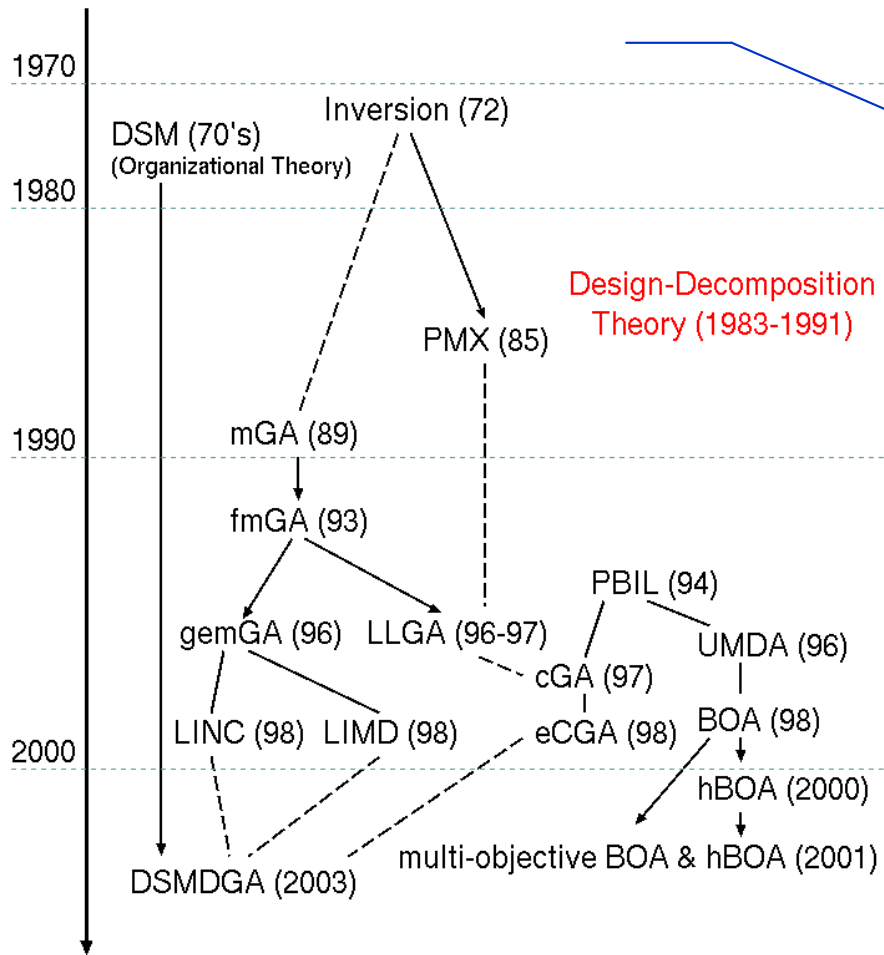
Kumara Sastry

Illinois Genetic Algorithms Laboratory  
Department of Industrial & Enterprise Systems Engineering  
University of Illinois at Urbana-Champaign

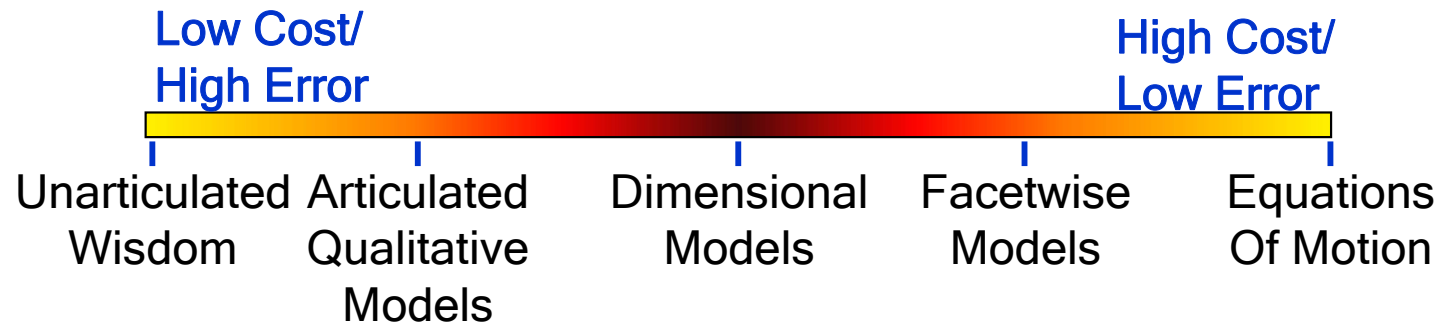
<http://www-illigal.ge.uiuc.edu>

[ksastry@uiuc.edu](mailto:ksastry@uiuc.edu)

# Competent Genetic Algorithms



[Goldberg (2002).  
Design of Innovation.]



# Estimation of Distribution Algorithms: Intractability To Tractability

## ❖ Competent genetic algorithms:

- ◆ Solve hard problems quickly, reliably, and accurately  
[Goldberg, 2002; Pelikan, 2005]
  - ★ Decomposable and Hierarchical problems
- ◆ **Intractability → tractability**
- ◆ Polynomial (usually sub-quadratic) scalability
- ◆ Even sub-quadratic scalability may be daunting
  - ★ Complex search problems
  - ★ 1000 variables, 10 seconds/evaluation → ~120 days!

# Efficiency Enhancement: Tractability To Practicality

- ❖ Even sub-quadratic evaluations can be daunting
  - ◆ Large number of variables
  - ◆ Expensive evaluation
  - ◆ Limited computational resource
  - ◆ Large order sub-problems in decomposition
- ❖ Need Efficiency-Enhancement Techniques
  - ◆ Tractability → Practicality
  - ◆ Premium on principled design methodology that yields greatest speed-up

# Outline

- ❖ Decomposition of efficiency-enhancement techniques
- ❖ Efficiency enhancement in EAs
  - ◆ Parallelization
  - ◆ Evaluation relaxation
  - ◆ Time Continuation
  - ◆ Hybridization
- ❖ Efficiency enhancement in EDAs
- ❖ Other efficiencies for practical use of EDAs

[ [http://medal.cs.umsl.edu/files/EE\\_for\\_EDAs.pdf](http://medal.cs.umsl.edu/files/EE_for_EDAs.pdf),  
<http://gal31.ge.uiuc.edu/kumara/2005/11/24/principled-efficiency-enhancement-techniques/>]

# Rationale For Decomposition: Trade-Offs

## ❖ Quality-Duration trade-off

- ◆ Longer run duration, with high quality solution Vs. shorter run duration with lower quality solution

## ❖ Accuracy-Cost trade-off

- ◆ Accurate, but costly evaluation Vs. less accurate, but cheaper evaluation

## ❖ Time Budgeting tradeoffs

- ◆ Single large population run Vs. multiple small population runs

# Decomposition of Efficiency Enhancement Techniques

## ❖ Four part harmony of efficiency-enhancement techniques (EETs)

- ◆ **Parallelization** [[Cantú-Paz, 2000](#); Ocenasek et al, 2003; Munetomo et al 2005]
- ◆ **Hybridization** [Moscata, 1989; Davis, 1991; Hart, 1994; [Goldberg & Voessner, 1999](#); Krasnogor, 2002; [Sinha, 2003](#); Pelikan, Sastry & Goldberg, 2006]
- ◆ **Time utilization** [[Goldberg, 1999](#); [Srivastava, 2002](#); [Sastry & Goldberg, 2004](#); Lima, et al 2005; Lima et al 2006]
- ◆ **Evaluation relaxation** [[Miller, 1997](#); [Sastry, 2002](#); Albert, 2002; Jin, 2003; [Sastry, Pelikan & Goldberg, 2004](#); Pelikan & Sastry, 2004; Sastry, Lima, Goldberg, 2006]

# Parallelization

- ❖ Distribute one or more EA components among multiple processors
- ❖ EAs are easy to parallelize
  - ◆ Population-based methods
  - ◆ Fitness does not depend on other individuals
- ❖ Four types of parallel EAs [[Cantú-Paz, 2000](#)]
  - ◆ Single-population master-slaves
  - ◆ Multiple populations
  - ◆ Fine grained
  - ◆ Hierarchical combinations

[Adamidis, 1994; Cantú-Paz, 1998; Lin *et al*, 1994; Alba & Troya, 1999]

# Hybridization

- ❖ Couple domain-specific and local-search techniques with GAs
  - ◆ Faster convergence, Repair, Initialization, Solution refinement, Robustness, etc.,
- ❖ A necessary component of industrial strength GAs
- ❖ Theoretical understanding is limited:
  - ◆ Division of labor: Spatial & temporal
  - ◆ Effect of local search on sampling
  - ◆ Optimal duration of local search

[Bosworth et al, 1972, Moscata, 1989; Davis, 1991; Orvash & Davis, 1993; Ibaraki, 1997; Fleurent & Ferland, 1994; Ramsey & Grefenstette, 1993; Hartman & Rieger, 2001; Lima *et al*, 2005; Hart, 1994; [Goldberg & Voessner, 1999](#); Krasnogor, 2002; [Sinha, 2003](#)]

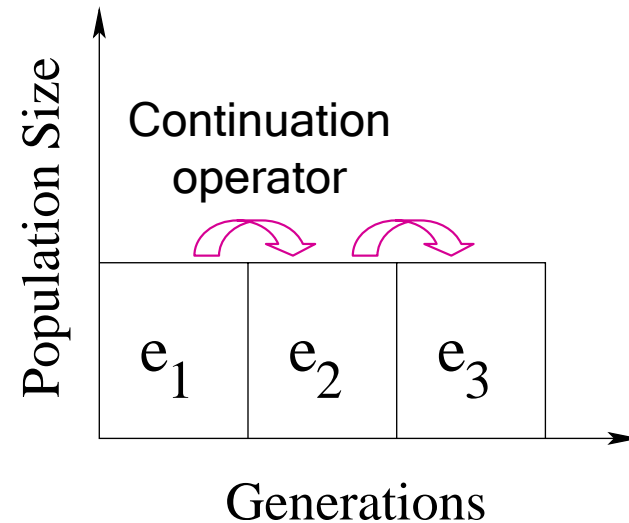
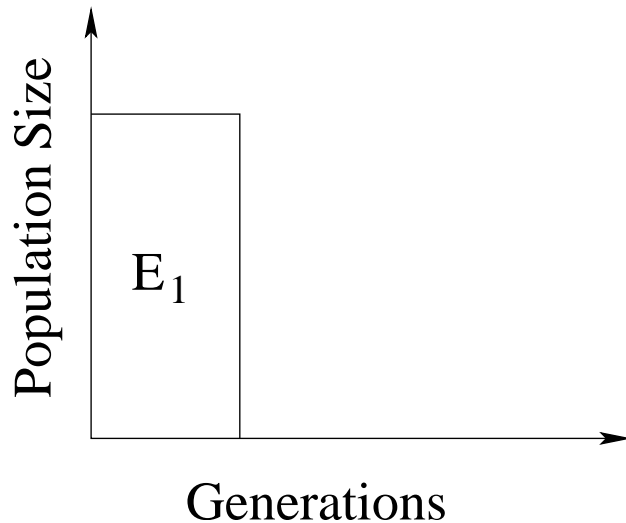
# Evaluation Relaxation

- ❖ Replace an accurate, but costly function evaluation with approximate, but cheap function evaluations
- ❖ Approximation introduces error in evaluation
- ❖ Error comes in two flavors: **Variance** and **Bias**
  - ◆ **Variance** can be handled **spatially**
  - ◆ **Bias** has to be handled **temporally**
- ❖ Source of approximation:
  - ◆ Exogenous
  - ◆ Endogenous
  - ◆ Exogenous and Endogenous

[[Miller, 1997](#); [Sastry, 2002](#); Albert, 2002; Jin, 2003; [Sastry, Pelikan & Goldberg, 2004](#); Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006]

# Time Continuation

- ❖ Large population, single-convergence epoch Vs small population, multiple epochs.
  - ◆ Given fixed computational resources
    - ★ Pop size x run duration x # epochs



- ★ Continuation operator: **Selectively perturb incorrectly set subsolutions**

[[Goldberg, 1999](#); [Srivastava, 2002](#); Vrajitoru, 2000; Luke, 2001; [Sastry & Goldberg, 2004](#); Lima et al, 2005; Lima et al 2006]

# Tractability to Practicality: Combining Different Facets of EETs

- ❖ If each efficiency enhancement is independent
  - ◆ Speed-up of each EET would hold
  - ◆ **Total speed-up is multiplicative not additive!**
  - ★ Moderate speed-ups are also substantial

Master-slave parallel GA on 100 cpus:	100.00
Speed-up from hybridization:	1.30
Speed-up from continuation:	1.20
Speed-up from evaluation relaxation:	1.25
Total Speed-up: $100 * 1.3 * 1.2 * 1.5$	<u>195.00</u>

- ★ **Computing power of additional 95 processors!**

# Efficiency Enhancement in EDAs

- ❖ EETs in EDAs as opposed to simple EAs
  - ◆ **Effective use of information in the probabilistic models**
  - ◆ Calls for tight integration of models and surrogate design
- ❖ Efficiency enhancement of probabilistic model building
  - ◆ Model building can be expensive as well
  - ◆ Speedup model building process
- ❖ Efficient EDA implementations
  - ◆ Program and memory efficiencies

# Tight Integration of Probabilistic Models and EETs

- ❖ Illustrate with two examples
- ❖ Evaluation relaxation in EDAs
  - ◆ Inducing surrogate form using probabilistic models
- ❖ Adaptive time continuation in EDAs
  - ◆ Inducing neighborhood operators for local search using probabilistic models

# Inducing Surrogate Form Using Probabilistic Models

## ❖ Sub-structure identification

- ◆ Use linkage-learning techniques including EDAs
  - ★ E.g., eCGA

## ❖ Surrogate form induced from the probabilistic model

- ◆ Example model: [1 3] [2] [4]
- ◆ Surrogate:

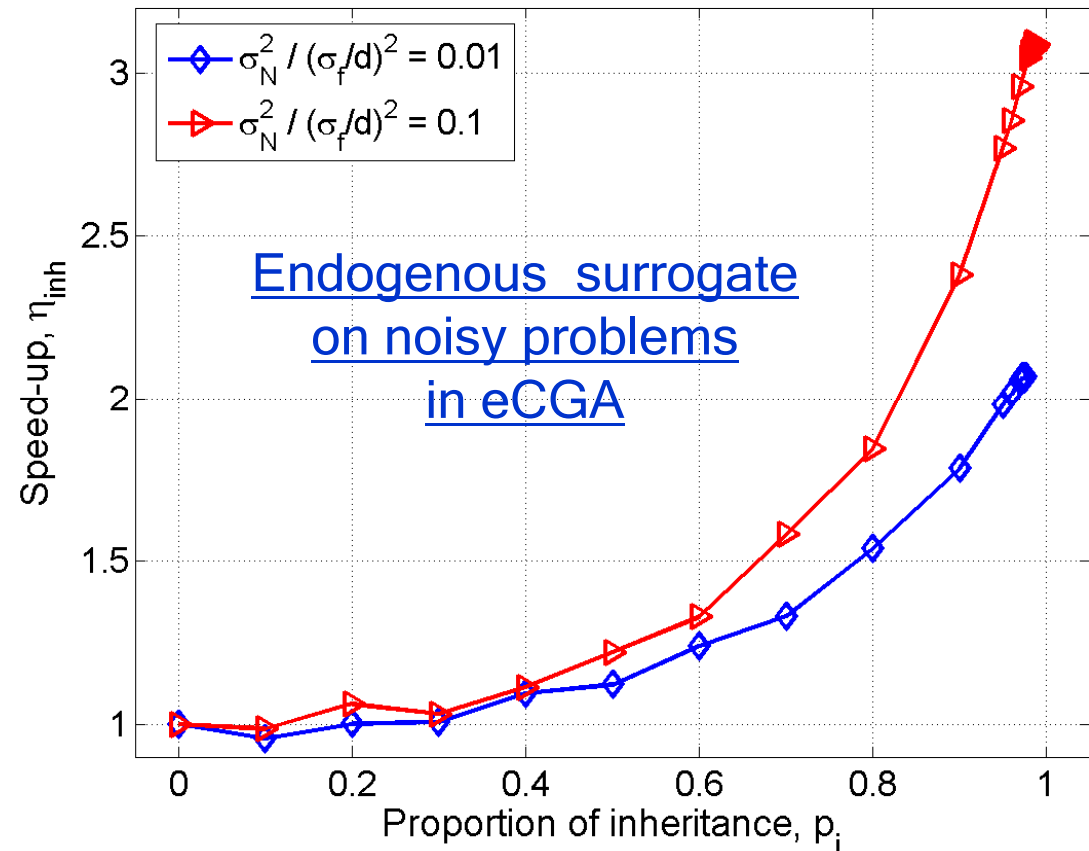
$$f_s = c_0 + c_1x_1 + c_3x_3 + c_{1,3}x_1x_3 + c_2x_2 + c_4x_4$$

## ❖ Coefficients of Surrogates: Subsolution qualities

- ◆ Can use linear regression methods

# Efficiency Enhancement via Substructural Surrogates

- ❖ Identify key substructures using EDAs
- ❖ Infer the surrogate structure from the substructural model
- ❖ Use least squares method to estimate the coefficients of the surrogate
- ❖ Only 1-3% individuals need expensive evaluation
- ❖ Speedup: 2-3
  - ◆ Simple surrogates don't give any speedup

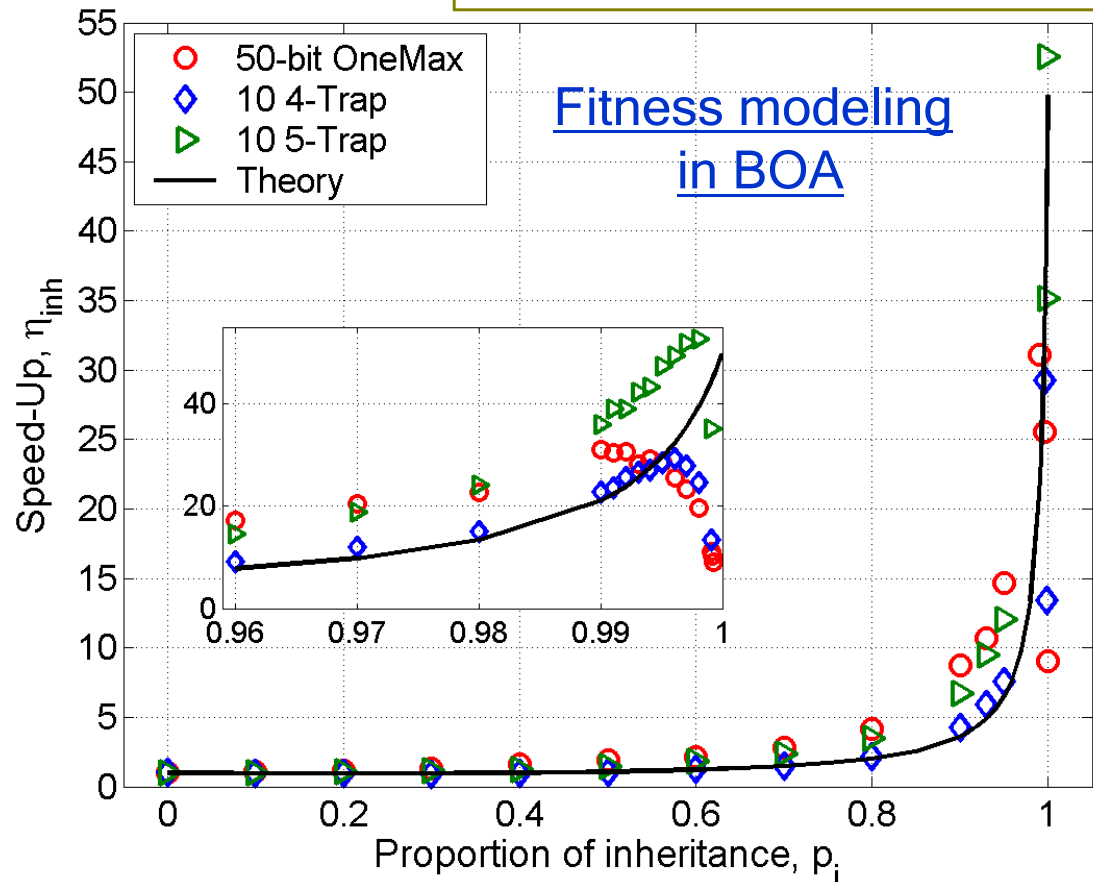
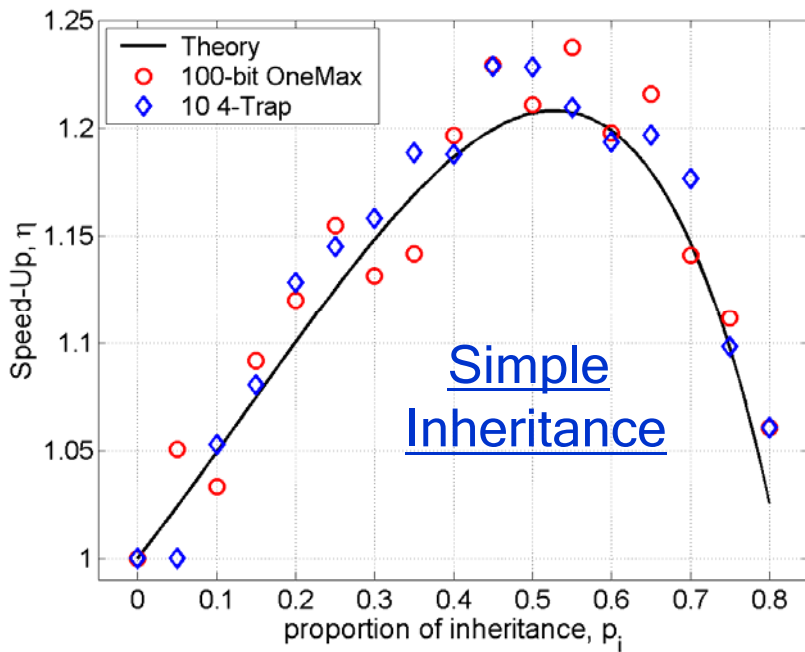


[Sastry et al 2006]

# Substructural Surrogates in BOA

- ❖ **Speed-Up**: Ratio of # function evaluations without efficiency enhancement to that with it.
- ❖ **Only 1-15% individuals need evaluation**
- ❖ **Speed-Up: 30-53**

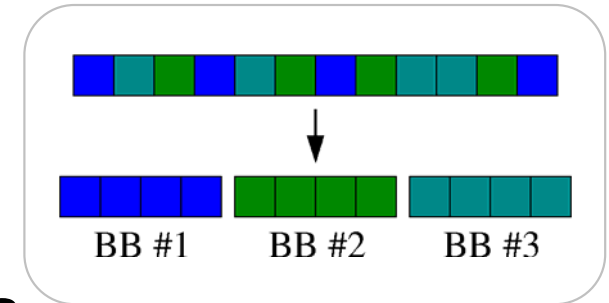
$$\eta \approx (1 - p_i)^{-1} (1 + p_i)^{-1.5}$$



# Inducing Neighborhood Operators Using Probabilistic Models

## ❖ Inducing good neighborhoods

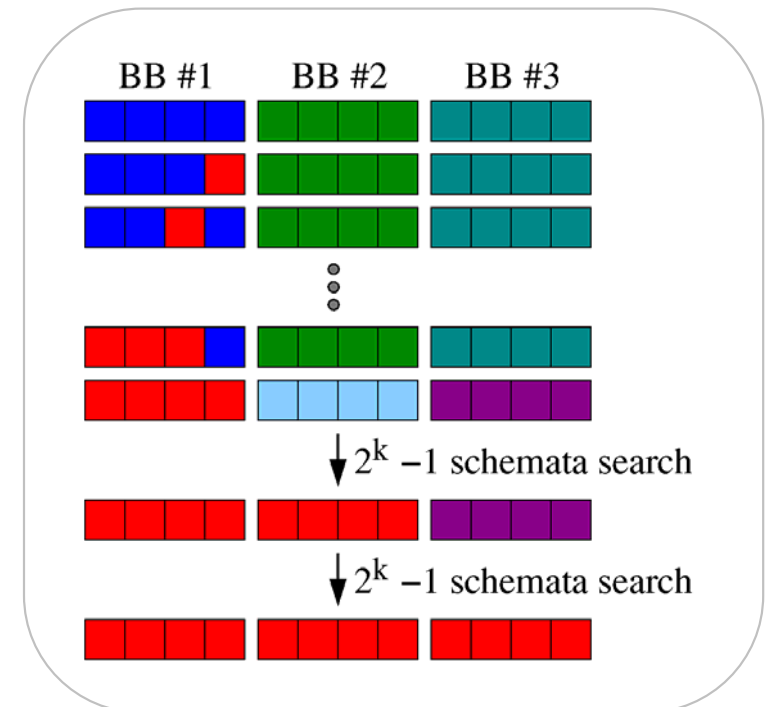
- ◆ Sample of candidate solutions
- ◆ Use linkage-learning procedures developed for selectorecombinative GAs



## ❖ Neighborhood: Space vs. Sub-space

## ❖ Search: Hillclimbing vs. Random

- ◆ Consider linkage partitions
  - ★ Arbitrary left-to-right order
- ◆ Choose the best schemata
  - ★ Among the  $2^k$  possible ones



# Deterministic Problems: Mutation

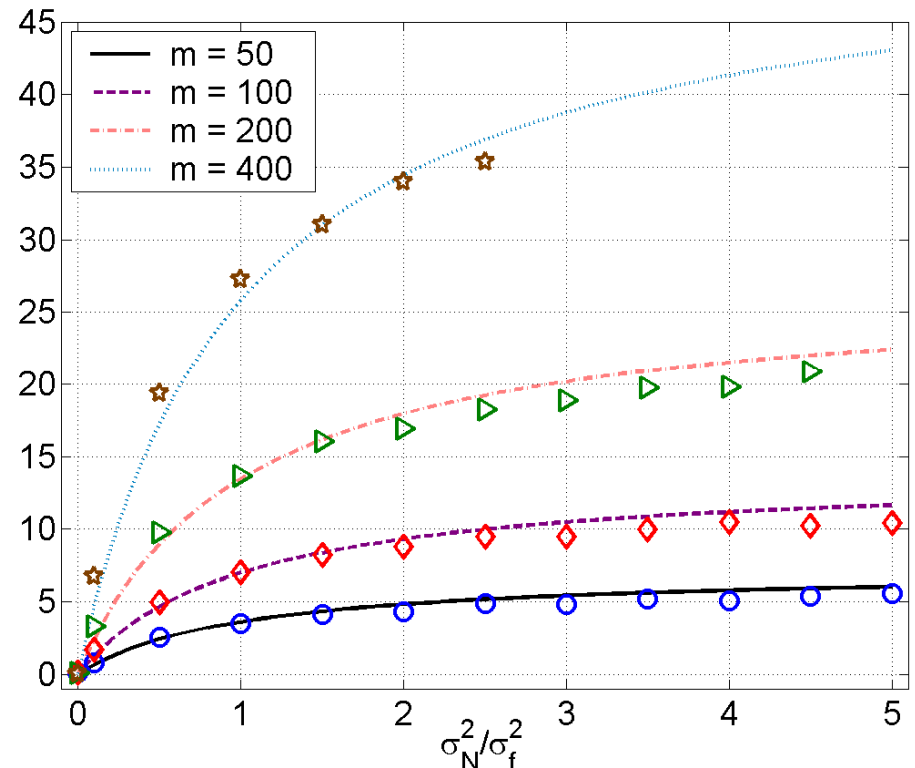
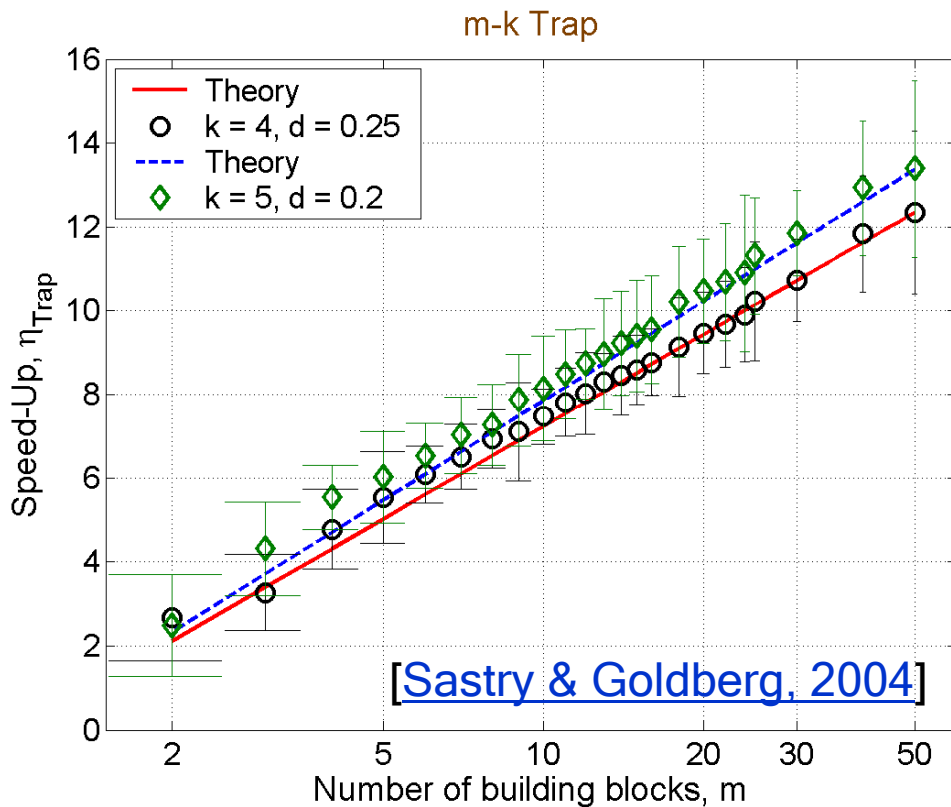
# Noisy Problems: Crossover

- ❖ Deterministic fitness: Mutation is more efficient

$$\eta = \frac{n_{fe,M}}{n_{fe,XO}} = \mathcal{O}(\sqrt{k} \log m)$$

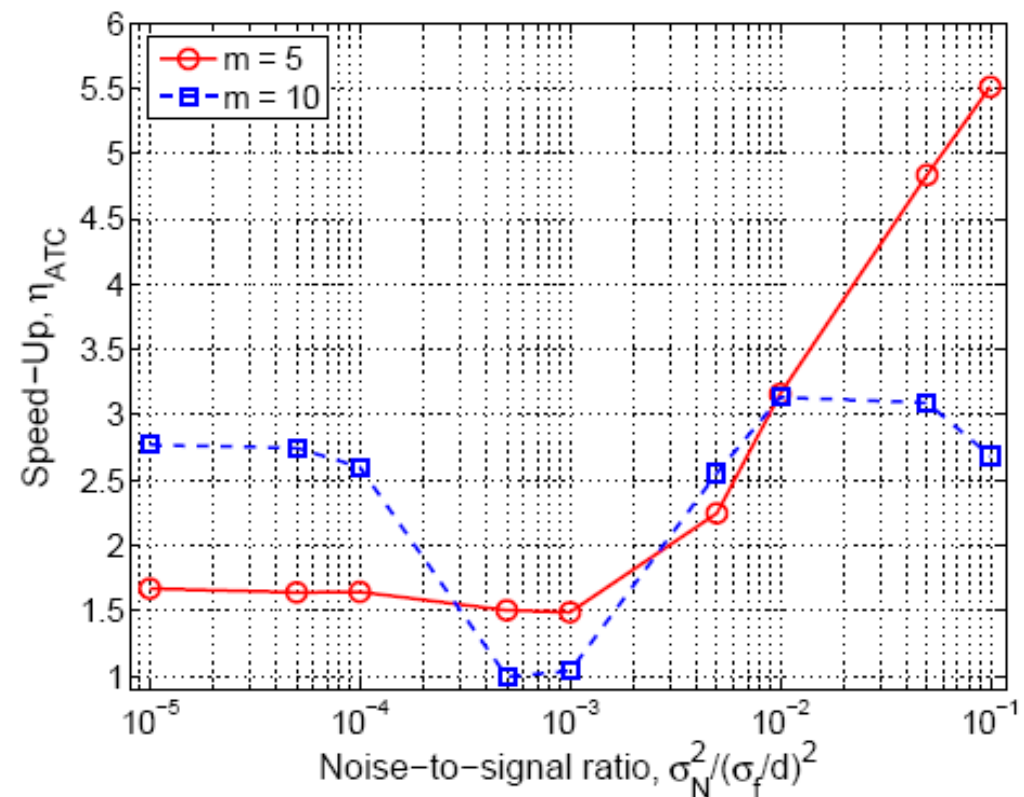
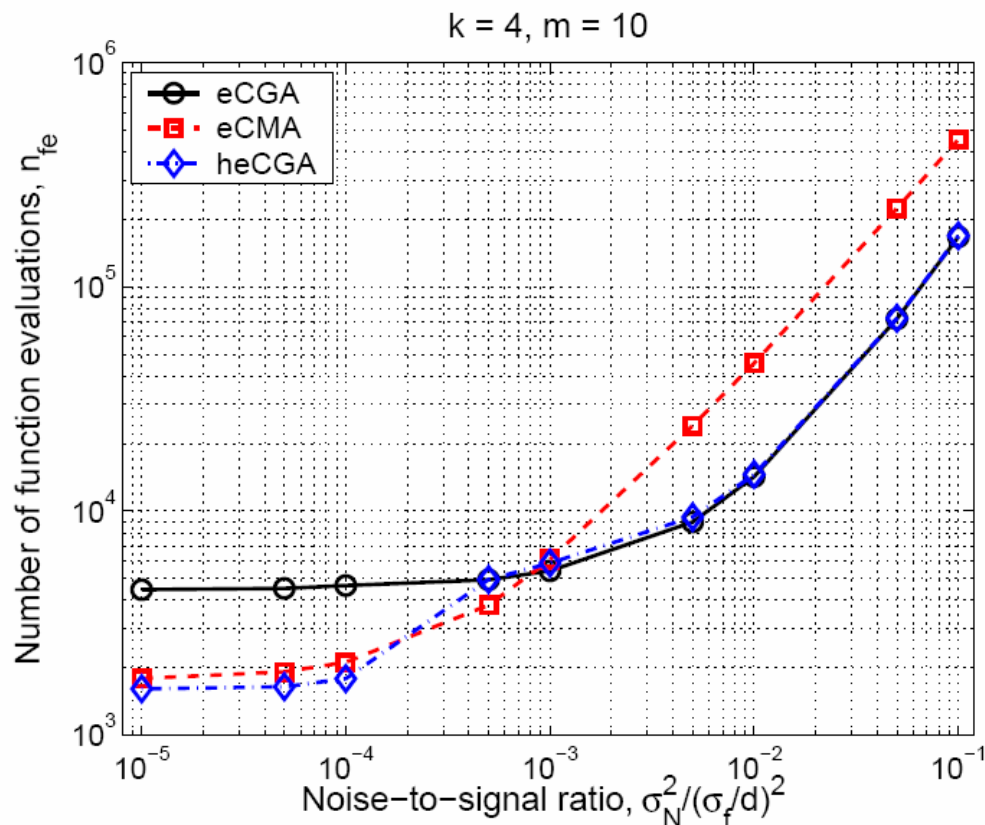
- ❖ Noisy fitness: Recombination is more efficient

$$\eta_{\text{noise}} = \frac{n_{fe,XO}}{n_{fe,M}} = \mathcal{O}\left[\sqrt{k} \frac{m}{\log m} \left(\frac{\sigma_N^2/\sigma_f^2}{1 + \sigma_N^2/\sigma_f^2}\right)\right]$$



# Adaptive Time Continuation in EDAs: Noisy Fitness

- ❖ As noise increases, switch from mutation-dominated search to crossover-dominated search mode



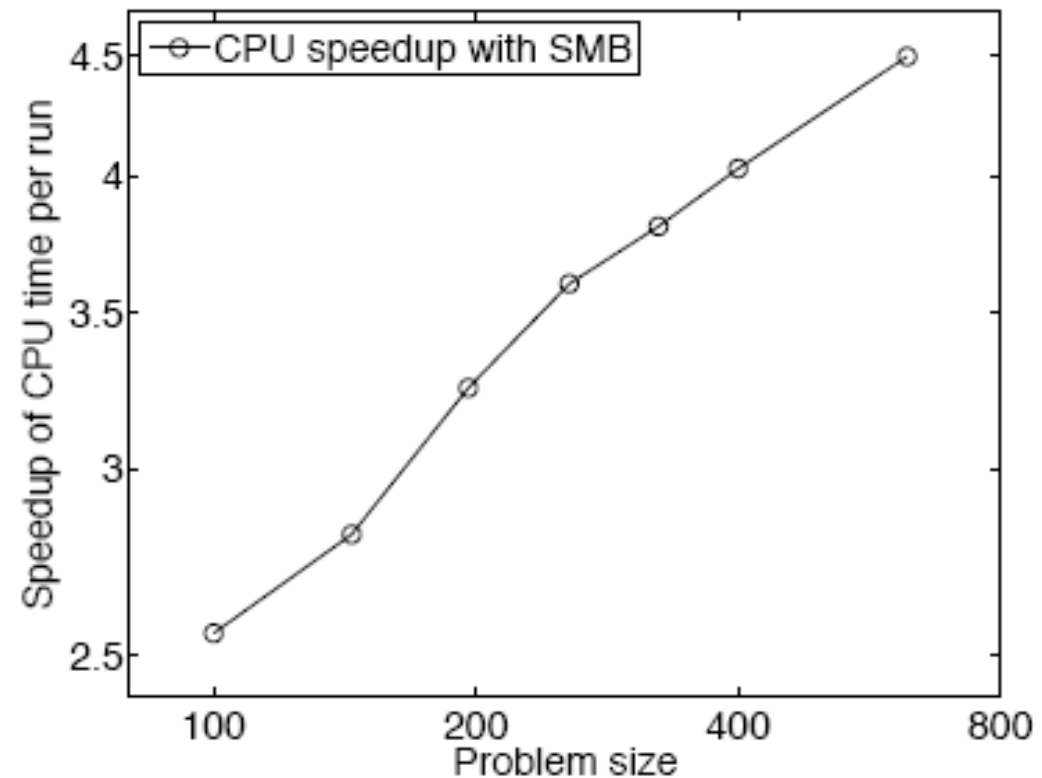
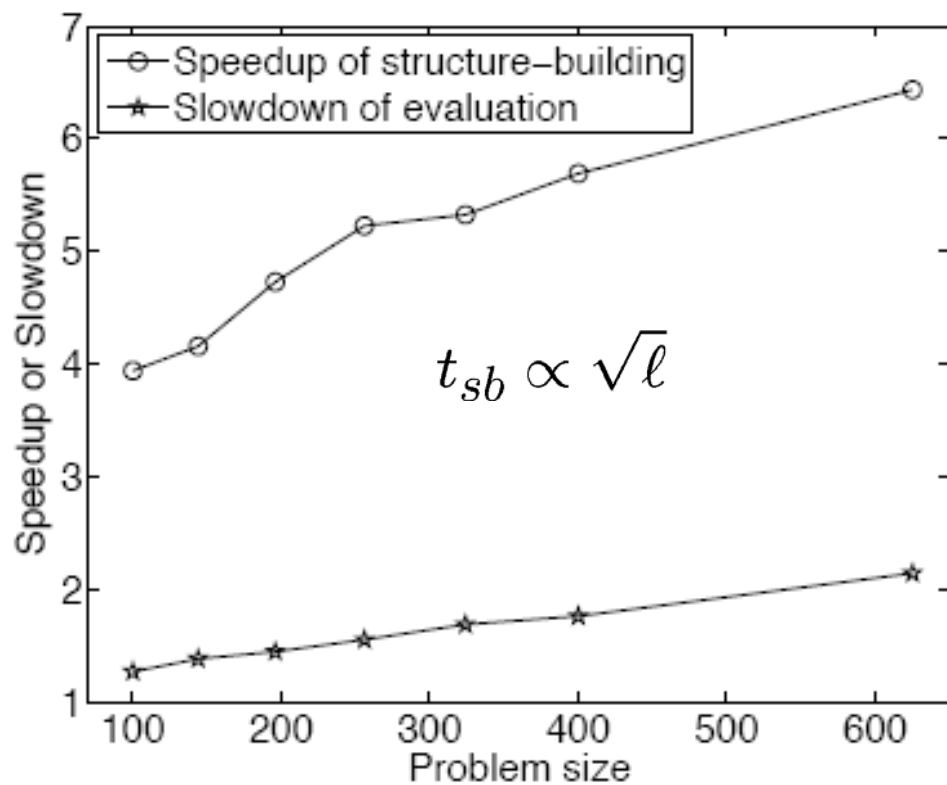
[Lima et al 2005; Lima et al 2006]

# Efficiency Enhancement of Model Building

- ❖ **Parallelization of model building** [Ocenasek et al, 2003; Munetomo et al 2005]
- ❖ **Prior knowledge utilization** [Baluja, 2002]
- ❖ **Model relaxation techniques**
  - ◆ **Example: Sporadic model building** [Pelikan et al 2006]
    - ★ **Learn structure of probabilistic models intermittently**
    - ★ Remaining generations use structure from previous iteration
    - ★ Parameters are always updated
    - ★ **Save time in most expensive part of model building**

# Speedup via Sporadic Model Building

- ❖ Model-building speed-up increases with problem size
- ❖ Evaluation slowdown increases with problem size
- ❖ Speedup in CPU time increases with problem size



[Pelikan, Sastry & Goldberg, 2006]

# Road to Supermultiplicative Speedups: Combining Competent GAs with Different Facets of EETs

## ❖ Multiplicative speedup is substantial

Master-slave parallel GA on 100 cpus:	100.00
Speed-up from hybridization:	1.30
Speed-up from continuation:	1.20
Speed-up from evaluation relaxation:	1.25
Total Speed-up: $100 * 1.3 * 1.2 * 1.5$	<u>195.00</u>

- ❖ Tight integration of probabilistic models of EDAs with design of EETs yield significantly higher speedups
  - ◆ 1.25 to 53 for an evaluation-relaxation scheme
  - ◆ Evidence of supermultiplicative speedups!
  - ◆ Can explain with facetwise models