Probabilistic Model-Building Genetic Algorithms

a.k.a. Estimation of Distribution Algorithms
a.k.a. Iterated Density Estimation Algorithms

Martin Pelikan

MEDAL

Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math. and Computer Science
University of Missouri at St. Louis
pelikan@cs.umsl.edu
http://medal.cs.umsl.edu/

[last update: April 2008]

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Foreword

- Motivation
  - Genetic and evolutionary computation (GEC) popular.
  - Toy problems great, but difficulties in practice.
  - Must design new representations, operators, tune, …
- This talk
  - Discuss a promising direction in GEC.
  - Combine machine learning and GEC.
  - Create practical and powerful optimizers.

Overview

- Introduction
  - Black-box optimization via probabilistic modeling.
- Probabilistic Model-Building GAs
  - Discrete representation
  - Continuous representation
  - Computer programs (PMBGP)
  - Permutations
- Conclusions

Problem Formulation

- Input
  - How do potential solutions look like?
  - How to evaluate quality of potential solutions?
- Output
  - Best solution (the optimum).
- Important
  - No additional knowledge about the problem.

Why View Problem as Black Box?

- Advantages
  - Separate problem definition from optimizer.
  - Easy to solve new problems.
  - Economy argument.
- Difficulties
  - Almost no prior problem knowledge.
  - Problem specifics must be learned automatically.
  - Noise, multiple objectives, interactive evaluation.

Representations Considered Here

- Start with
  - Solutions are n-bit binary strings.

- Later
  - Real-valued vectors.
  - Program trees.
  - Permutations

Typical Situation

- Previously visited solutions + their evaluation:

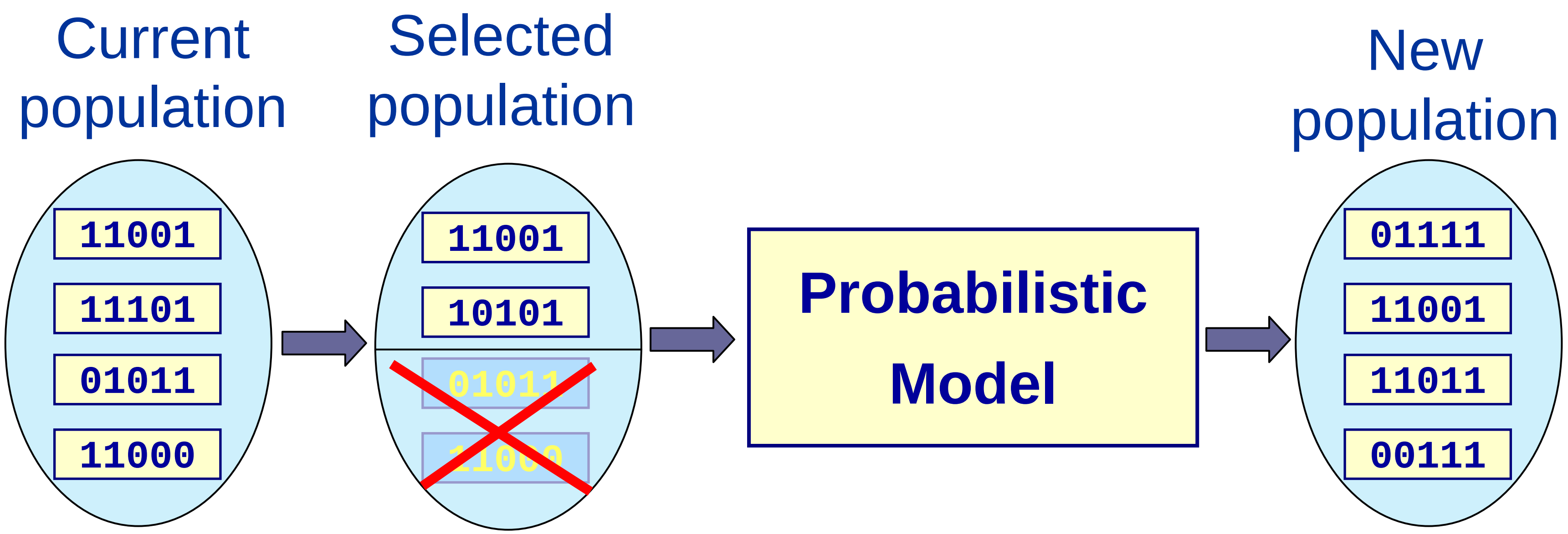
#	Solution	Evaluation
1	00100	1
2	11011	4
3	01101	0
4	10111	3

- Question: What solution to generate next?

Many Answers

- Hill climber
  - Start with a random solution.
  - Flip bit that improves the solution most.
  - Finish when no more improvement possible.
- Simulated annealing
  - Introduce Metropolis.
- Probabilistic model-building GAs
  - Inspiration from GAs and machine learning (ML).

Probabilistic Model-Building GAs



…replace crossover+mutation with learning and sampling probabilistic model

Other Names for PMBGAs

- Estimation of distribution algorithms (EDAs) (Mühlenbein & Paass, 1996)
- Iterated density estimation algorithms (IDEA) (Bosman & Thierens, 2000)

What Models to Use?

- Start with a simple example
  - Probability vector for binary strings.

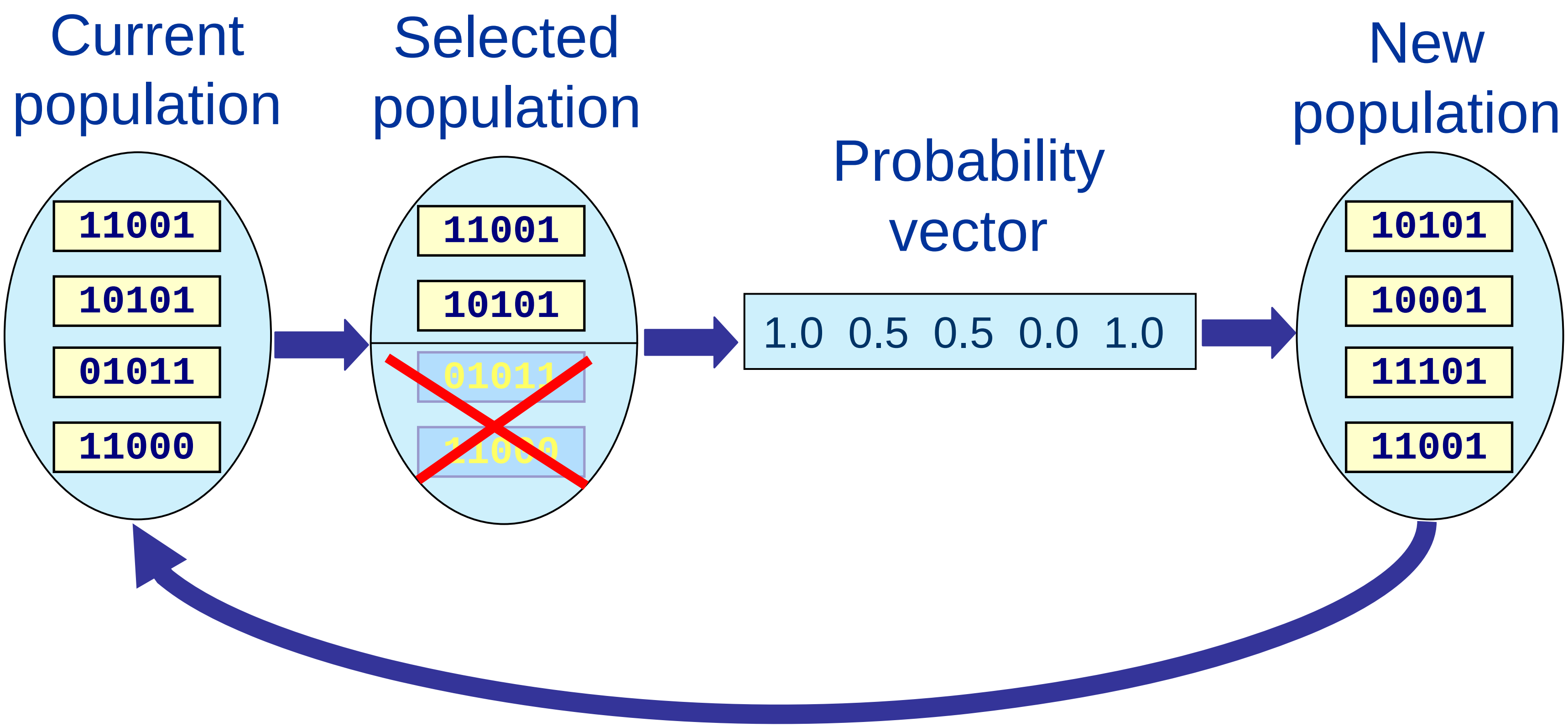
- Later
  - Dependency tree models (COMIT).
  - Bayesian networks (BOA).
  - Bayesian networks with local structures (hBOA).

Probability Vector

- Assume n-bit binary strings.
- Model: Probability vector $p=(p_1, \ldots, p_n)$
  - p_i = probability of 1 in position i
  - Learn p: Compute proportion of 1 in each position.
  - Sample p: Sample 1 in position i with prob. p_i

Example: Probability Vector

(Mühlenbein, Paass, 1996), (Baluja, 1994)



Probability Vector PMBGAs

- PBIL (Baluja, 1995)
  - Incremental updates to the prob. vector.
- Compact GA (Harik, Lobo, Goldberg, 1998)
  - Also incremental updates but better analogy with populations.
- UMDA (Mühlenbein, Paass, 1996)
  - What we showed here.
- DEUM (Shakya et al., 2004)
- All variants perform similarly.

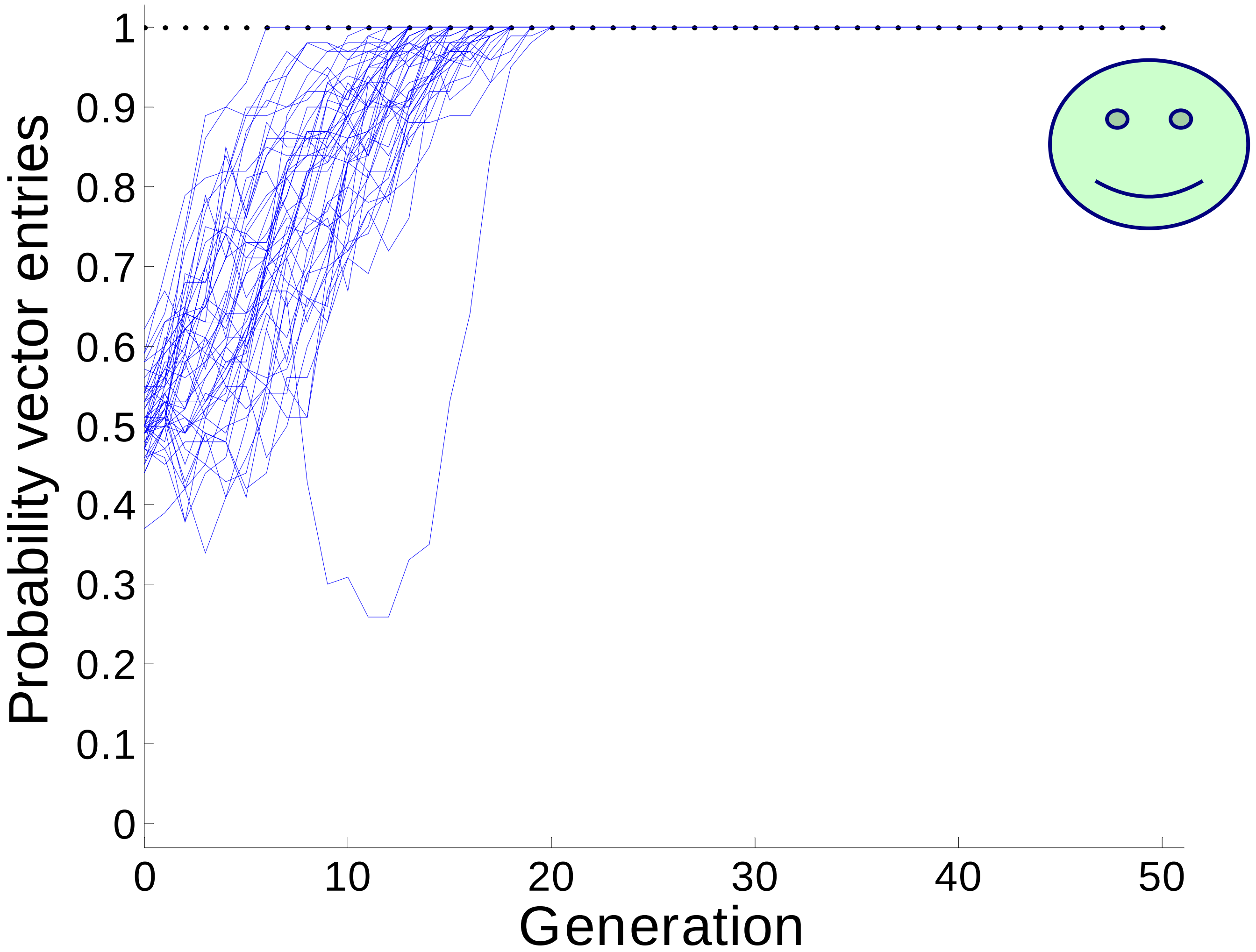
Probability Vector Dynamics

- Bits that perform better get more copies.
- And are combined in new ways.
- But context of each bit is ignored.

- Example problem 1: Onemax

$$f(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} X_i$$

Probability Vector on Onemax



Probability Vector: Ideal Scale-up

- O(n log n) evaluations until convergence
  - (Harik, Cantú-Paz, Goldberg, & Miller, 1997)
  - (Mühlenbein, Schlierkamp-Vosen, 1993)
- Other algorithms
  - Hill climber: O(n log n) (Mühlenbein, 1992)
  - GA with uniform: approx. O(n log n)
  - GA with one-point: slightly slower

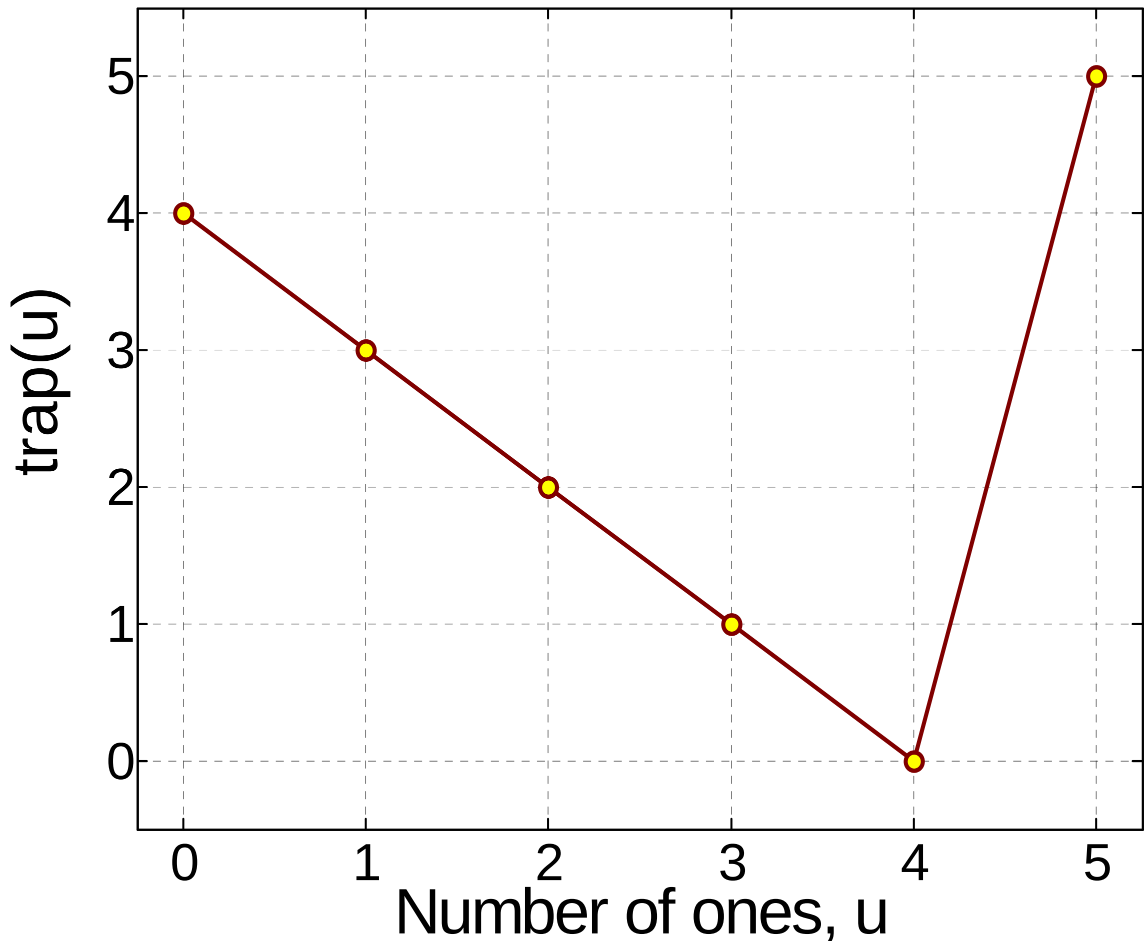
When Does Prob. Vector Fail?

- Example problem 2: Concatenated traps
  - Partition input string into disjoint groups of 5 bits.
  - Groups contribute via trap (ones=number of ones):

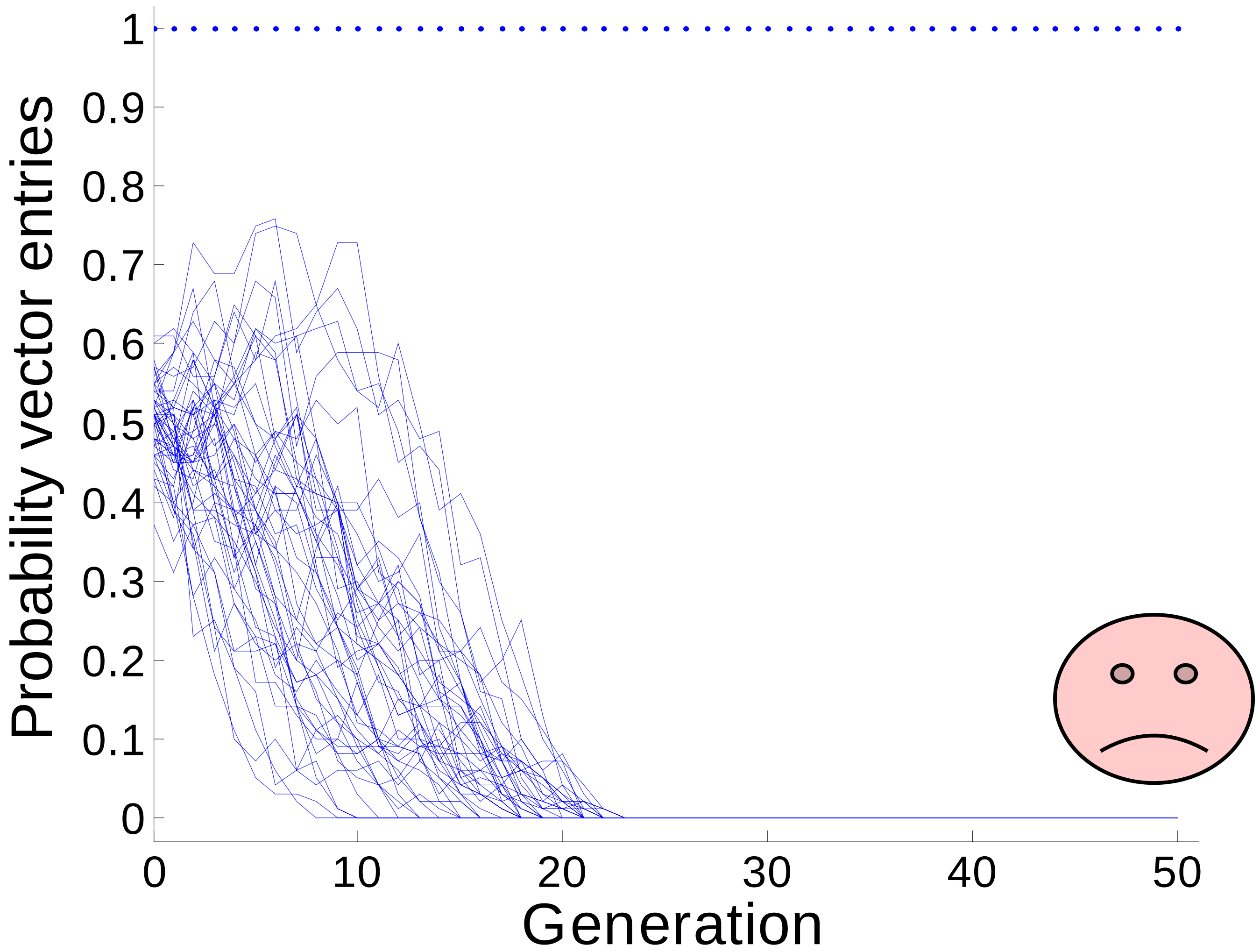
$$trap(ones) = \begin{cases} 5 & \text{if } ones = 5 \\ 4 - ones & \text{otherwise} \end{cases}$$

  - Concatenated trap = sum of single traps
  - Optimum: String 111...1

Trap-5



Probability Vector on Traps



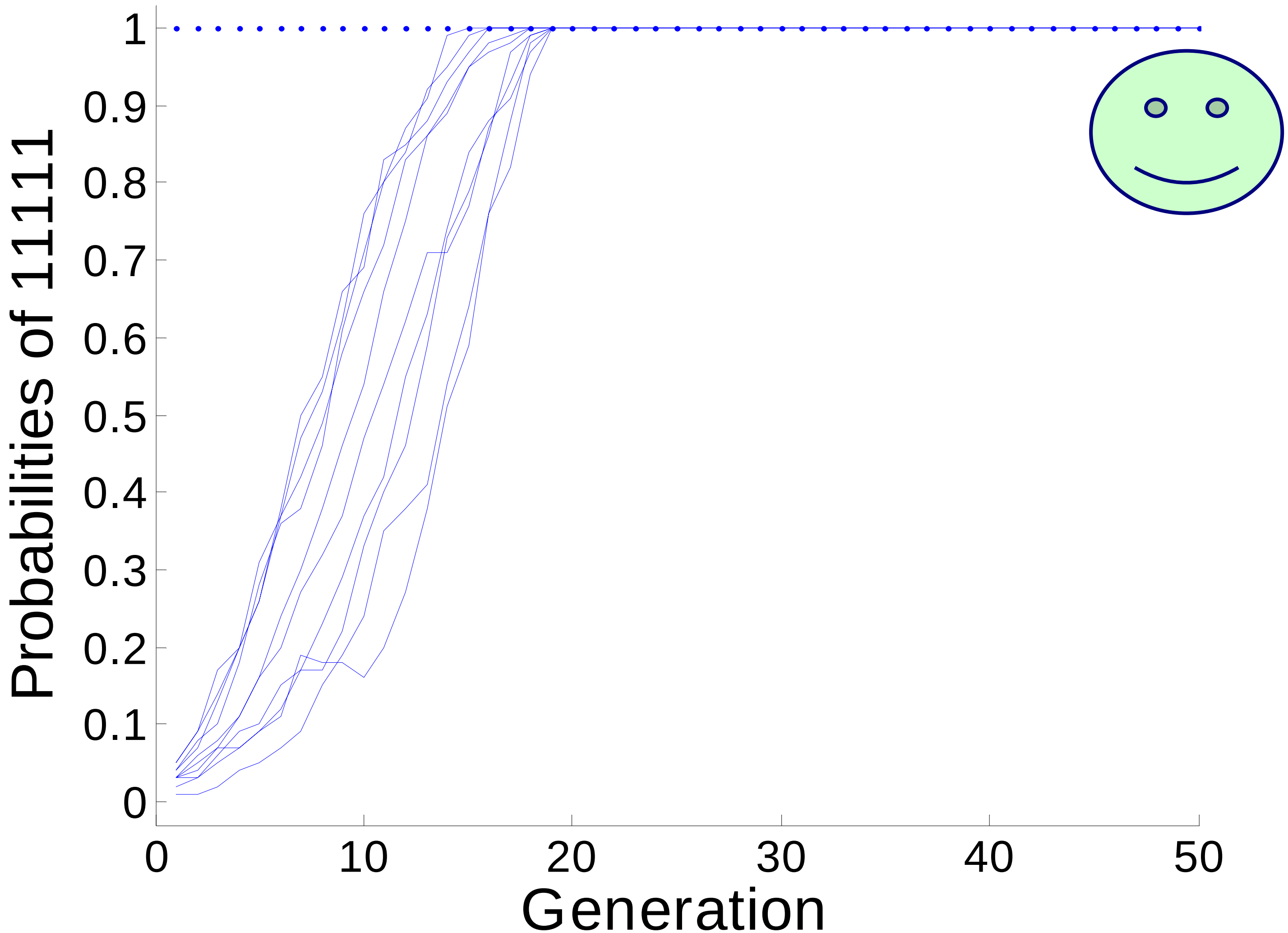
Why Failure?

- Onemax:
  - Optimum in 111...1
  - 1 outperforms 0 on average.
- Traps: optimum in 11111, but
  - f(0****) = 2
  - f(1****) = 1.375
- So single bits are misleading.

How to Fix It?

- Consider 5-bit statistics instead 1-bit ones.
- Then, 11111 would outperform 00000.
- Learn model
  - Compute p(00000), p(00001), ..., p(11111)
- Sample model
  - Sample 5 bits at a time
  - Generate 00000 with p(00000),
    00001 with p(00001), ...

Correct Model on Traps: Dynamics



Good News: Good Stats Work Great!

- Optimum in O(n log n) evaluations.
- Same performance as on onemax!
- Others
  - Hill climber: $O(n^5 \log n)$ = much worse.
  - GA with uniform: $O(2^n)$ = intractable.
  - GA with k-point xover: $O(2^n)$ (w/o tight linkage).

Challenge

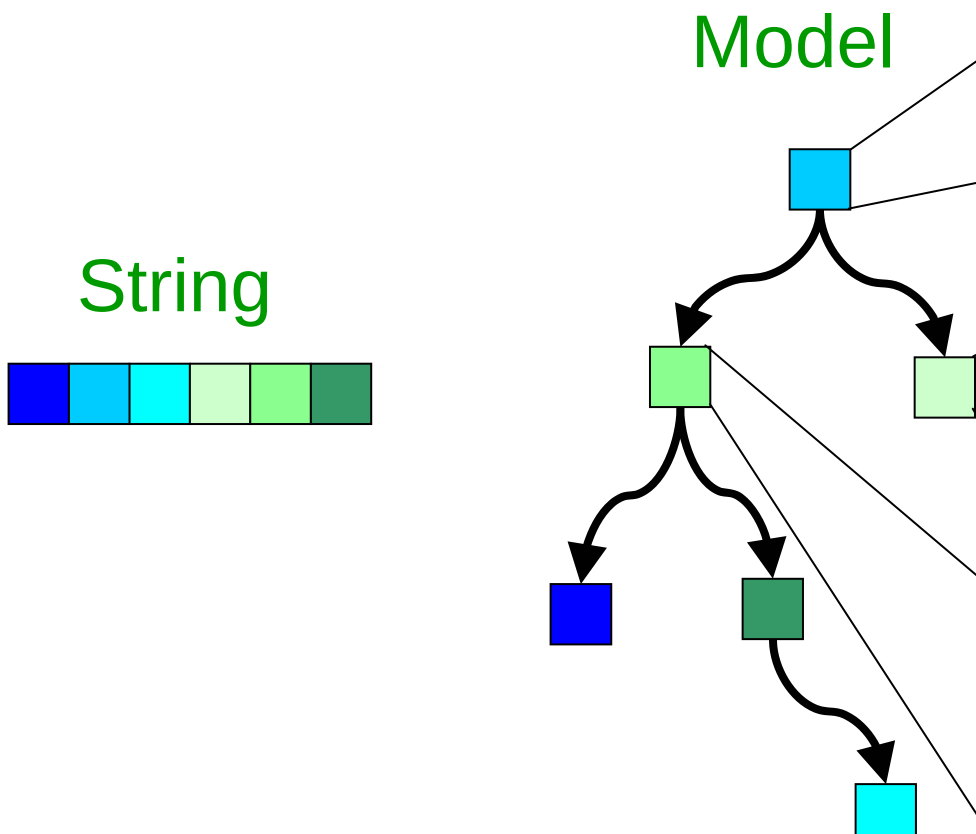
- If we could learn and use relevant context for each position
  - Find non-misleading statistics.
  - Use those statistics as in probability vector.
- Then we could solve problems decomposable into statistics of order at most *k* with at most $O(n^2)$ evaluations!
  - And there are many such problems (Simon, 1968).

What's Next?

- COMIT
  - Use tree models
- Extended compact GA
  - Cluster bits into groups.
- Bayesian optimization algorithm (BOA)
  - Use Bayesian networks (more general).

Beyond single bits: COMIT

(Baluja, Davies, 1997)



P(X=1)
75 %

X	P(Y=1|X)
0	30 %
1	25 %

X	P(Z=1|X)
0	86 %
1	75 %

How to Learn a Tree Model?

- Mutual information:

$$I(X_i, X_j) = \sum_{a,b} P(X_i = a, X_j = b) \log \frac{P(X_i = a, X_j = b)}{P(X_i = a)P(X_j = b)}$$

- Goal
  - Find tree that maximizes mutual information between connected nodes.
  - Will minimize Kullback-Leibler divergence.
- Algorithm
  - Prim's algorithm for maximum spanning trees.

Prim's Algorithm

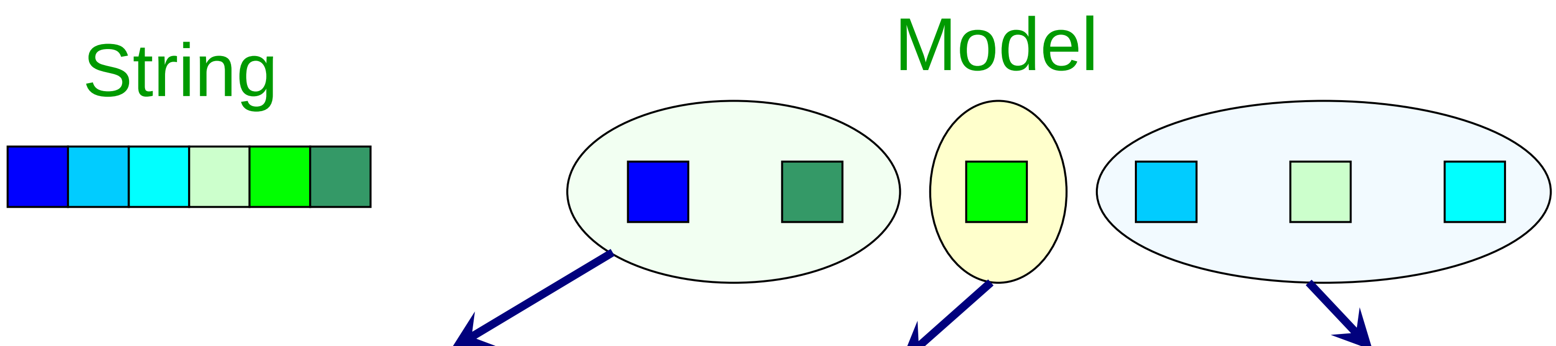
- Start with a graph with no edges.
- Add arbitrary node to the tree.
- Iterate
  - Hang a new node to the current tree.
  - Prefer addition of edges with large mutual information (greedy approach).
- Complexity: $O(n^2)$

Variants of PMBGAs with Tree Models

- COMIT (Baluja, Davies, 1997)
  - Tree models.
- MIMIC (DeBonet, 1996)
  - Chain distributions.
- BMDA (Pelikan, Mühlenbein, 1998)
  - Forest distribution (independent trees or tree)

Beyond Pairwise Dependencies: ECGA

- Extended Compact GA (ECGA) (Harik, 1999).
- Consider groups of string positions.



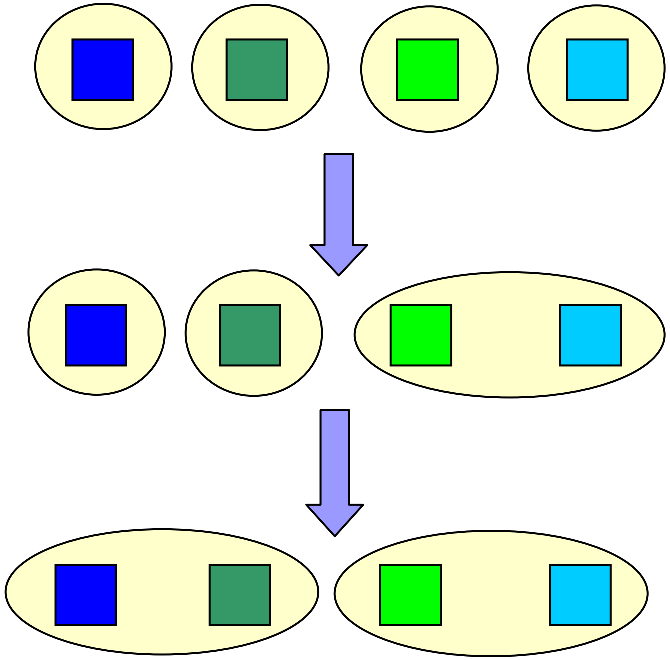
00	16 %
01	45 %
10	35 %
11	4 %

0	86 %
1	14 %

000	17 %
001	2 %
. . .	
111	24 %

Learning the Model in ECGA

- Start with each bit in a separate group.
- Each iteration merges two groups for best improvement.



How to Compute Model Quality?

- ECGA uses minimum description length.
- Minimize number of bits to store model+data:

$$MDL(M,D) = D_{Model} + D_{Data}$$

- Each frequency needs (0.5 log *N*) bits:

$$D_{Model} = \sum_{g \in G} 2^{|g|-1} \log N$$

- Each solution *X* needs -log p(*X*) bits:

$$D_{Data} = -N \sum_{X} p(X) \log p(X)$$

Sampling Model in ECGA

- Sample groups of bits at a time.
- Based on observed probabilities/proportions.
- But can also apply population-based crossover similar to uniform but w.r.t. model.

Building-Block-Wise Mutation in ECGA

- Sastry, Goldberg (2004); Lima et al. (2005)
- Basic idea
  - Use ECGA model builder to identify decomposition
  - Use the best solution for BB-wise mutation
  - For each k-bit partition (building block)
    - Evaluate the remaining 2^{k-1} instantiations of this BB
    - Use the best instantiation of this BB
- Result (for order-k separable problems)
  - BB-wise mutation is $O\left(\sqrt{k}\log n\right)$ times faster than ECGA!
  - But only for separable problems (and similar ones).

What's Next?

- We saw
  - Probability vector (no edges).
  - Tree models (some edges).
  - Marginal product models (groups of variables).

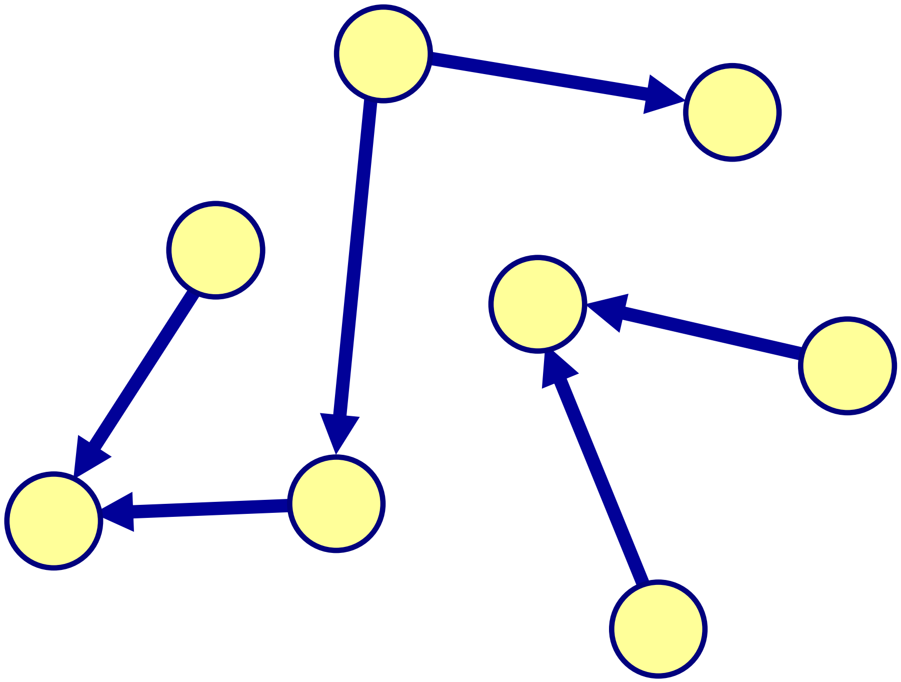
- Next: Bayesian networks
  - Can represent all above and more.

Bayesian Optimization Algorithm (BOA)

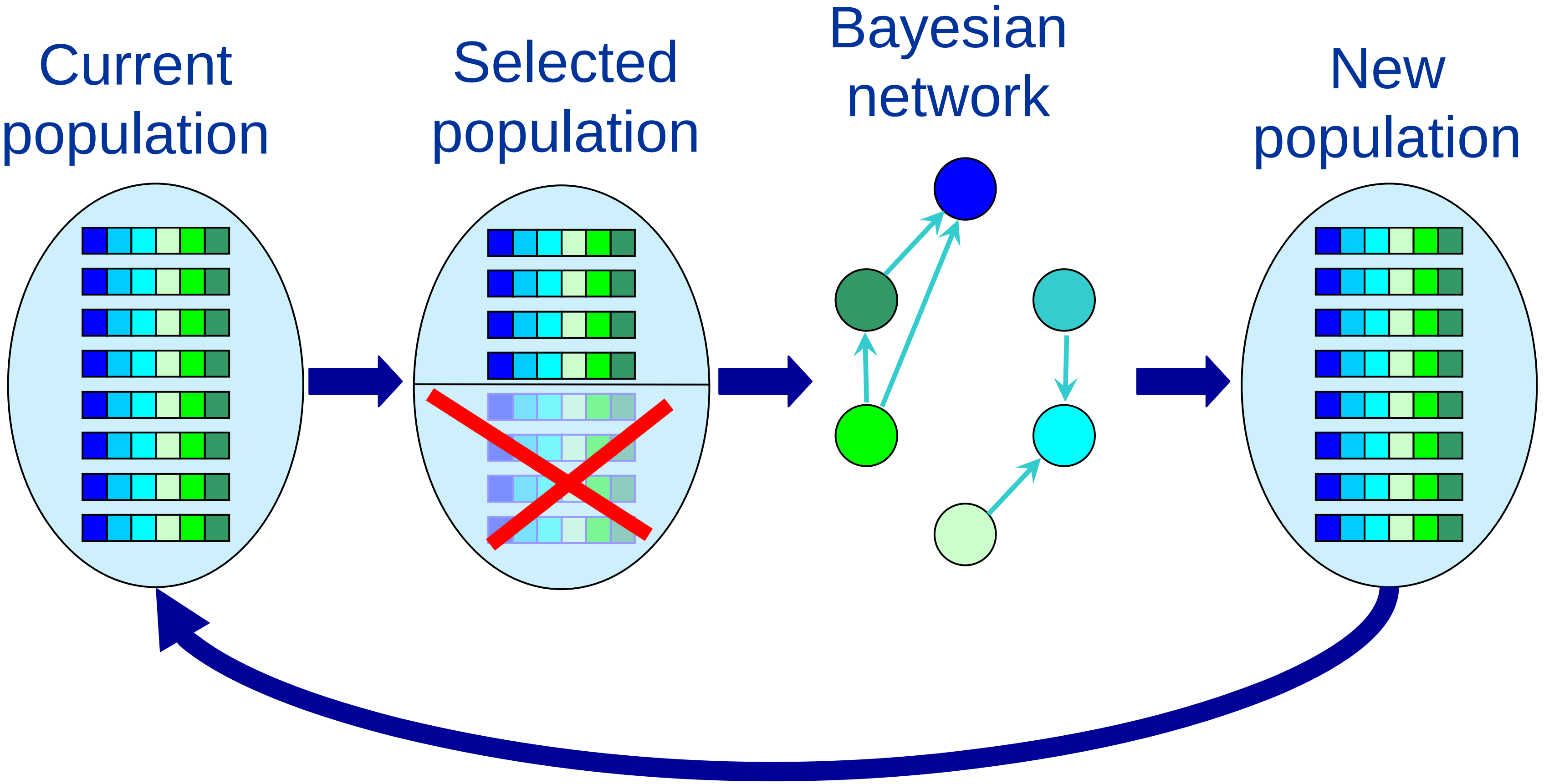
- Pelikan, Goldberg, & Cantú-Paz (1998)
- Use a Bayesian network (BN) as a model.
- Bayesian network
  - Acyclic directed graph.
  - Nodes are variables (string positions).
  - Conditional dependencies (edges).
  - Conditional independencies (implicit).

Example: Bayesian Network (BN)

- Conditional dependencies.
- Conditional independencies.



BOA


Current population
Selected population
Bayesian network
New population

Learning BNs

- Two things again:
  - Scoring metric (as MDL in ECGA).
  - Search procedure (in ECGA done by merging).

Learning BNs: Scoring Metrics

- Bayesian metrics
  - Bayesian-Dirichlet with likelihood equivallence

$$BD(B) = p(B)\prod_{i=1}^{n}\prod_{\pi_i}\frac{\Gamma(m'(\pi_i))}{\Gamma(m'(\pi_i)+m(\pi_i))}\prod_{x_i}\frac{\Gamma(m'(x_i,\pi_i)+m(x_i,\pi_i))}{\Gamma(m'(x_i,\pi_i))}$$

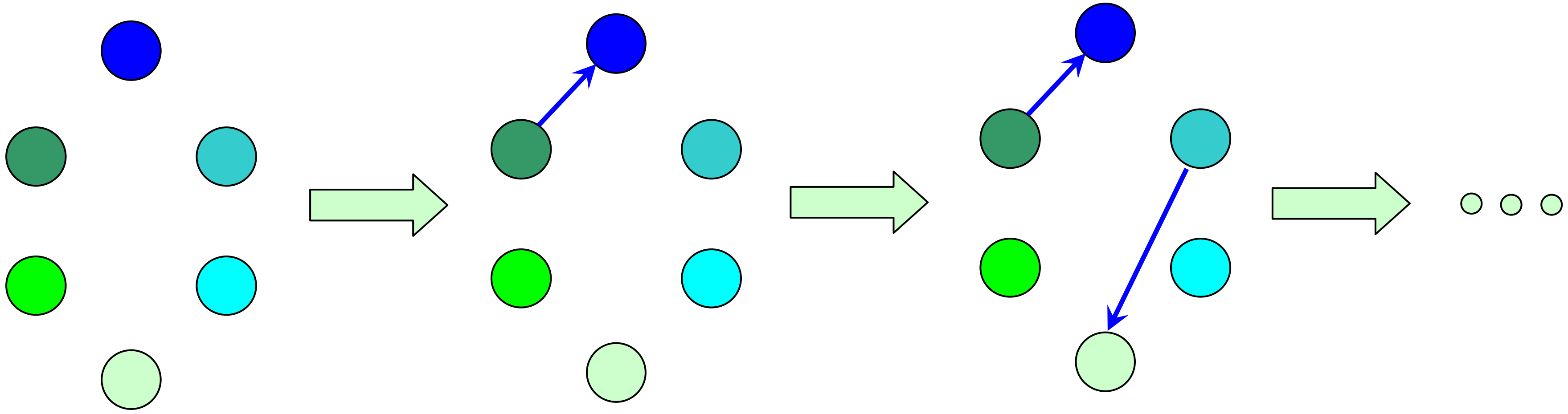
- Minimum description length metrics
  - Bayesian information criterion (BIC)

$$BIC(B) = \sum_{i=1}^{n}\left(-H(X_i\,|\,\Pi_i)N - 2^{|\Pi_i|}\frac{\log_2 N}{2}\right)$$

Learning BNs: Search Procedure

- Start with empty network (like ECGA).
- Execute primitive operator that improves the metric the most (greedy).
- Until no more improvement possible.
- Primitive operators
  - Edge addition (most important).
  - Edge removal.
  - Edge reversal.

Learning BNs: Example



BOA and Problem Decomposition

- Conditions for factoring problem decomposition into a product of prior and conditional probabilities of small order in Mühlenbein, Mahnig, & Rodriguez (1999).
- In practice, approximate factorization sufficient that can be learned automatically.
- Learning makes complete theory intractable.

BOA Theory: Population Sizing

- Initial supply (Goldberg et al., 2001)
  - Have enough stuff to combine. ⇨ $O\left(2^k\right)$
- Decision making (Harik et al, 1997)
  - Decide well between competing partial sols. ⇨ $O\left(\sqrt{n}\log n\right)$
- Drift (Thierens, Goldberg, Pereira, 1998)
  - Don’t lose less salient stuff prematurely. ⇨ $O(n)$
- Model building (Pelikan et al., 2000, 2002)
  - Find a good model. ⇨ $O\left(n^{1.05}\right)$

BOA Theory: Num. of Generations

- Two extreme cases, everything in the middle.
- Uniform scaling
  - Onemax model (Muehlenbein & Schlierkamp-Voosen, 1993)

$$\boxed{O\left(\sqrt{n}\right)}$$

- Exponential scaling
  - Domino convergence (Thierens, Goldberg, Pereira, 1998)

$$\boxed{O(n)}$$

Good News: Challenge Met!

- Theory
  - Population sizing (Pelikan et al., 2000, 2002)
    - Initial supply.
    - Decision making.
    - Drift.
    - Model building.

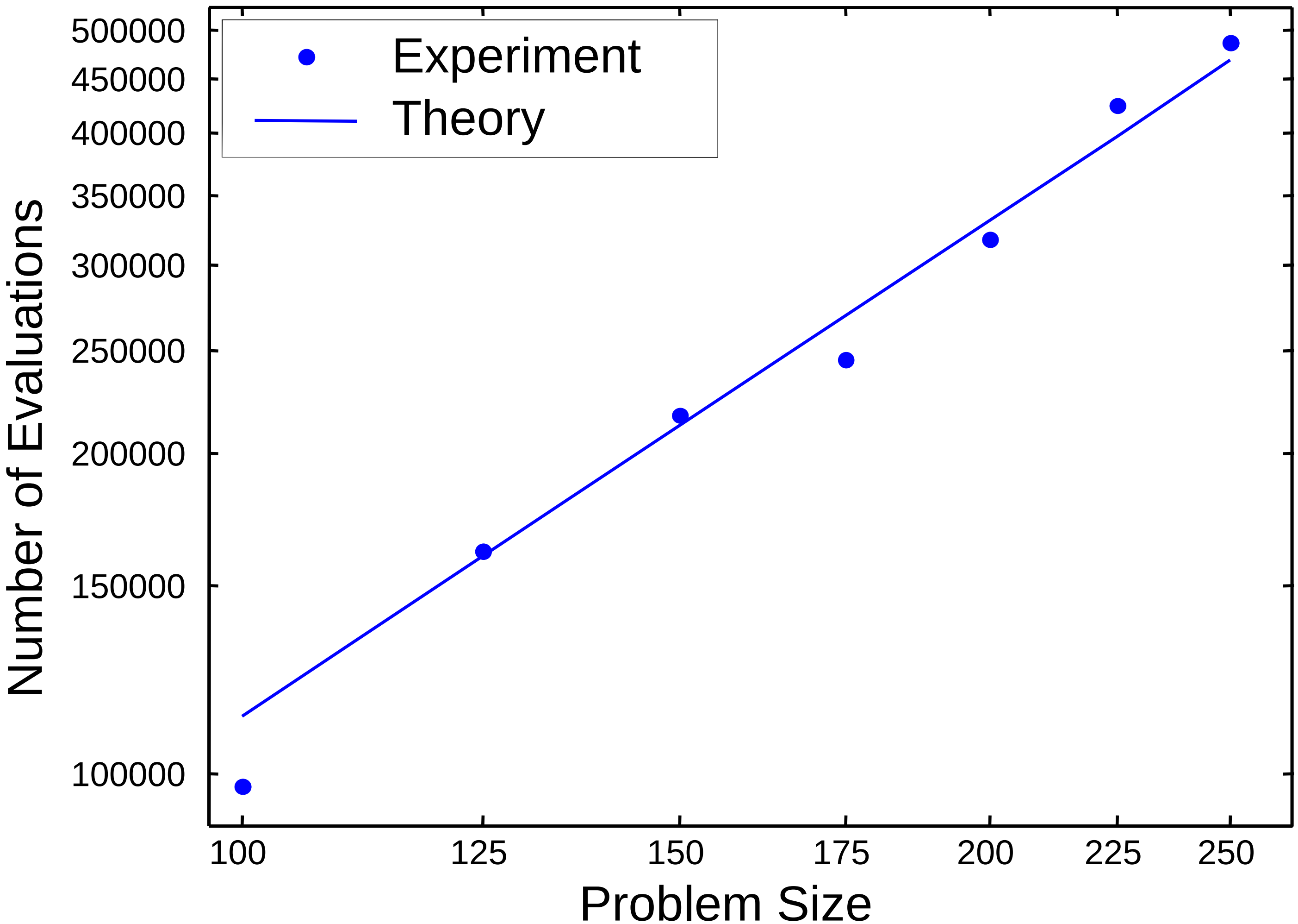
  → $O(n)$ to $O(n^{1.05})$

  - Number of iterations (Pelikan et al., 2000, 2002)
    - Uniform scaling.
    - Exponential scaling.

  → $O(n^{0.5})$ to $O(n)$

- BOA solves order-k decomposable problems in $O(n^{1.55})$ to $O(n^2)$ evaluations!

Theory vs. Experiment (5-bit Traps)

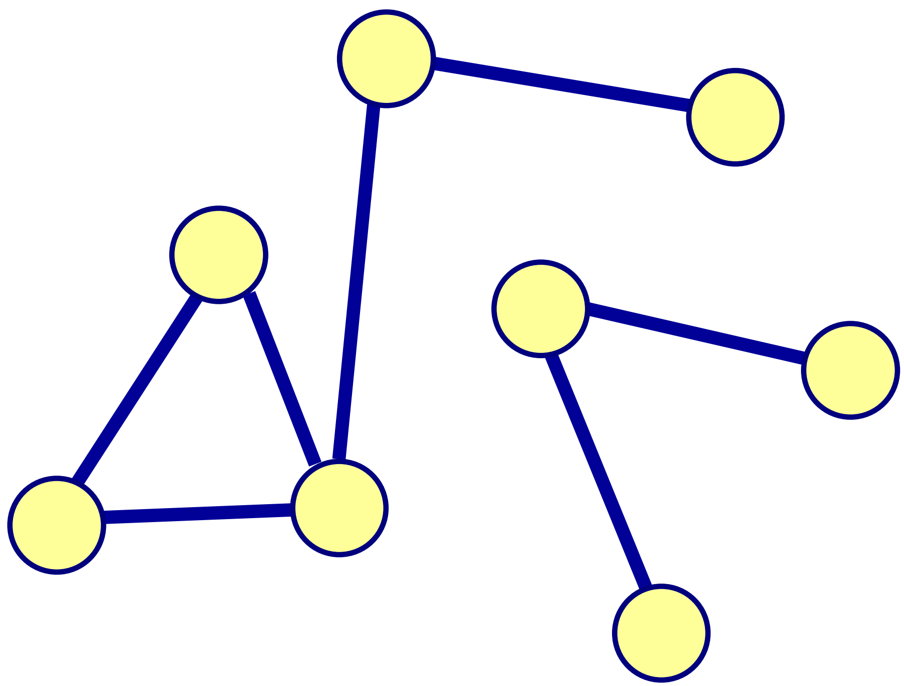


BOA Siblings

- Estimation of Bayesian Networks Algorithm (EBNA) (Etxeberria, Larrañaga, 1999).
- Learning Factorized Distribution Algorithm (LFDA) (Mühlenbein, Mahnig, Rodriguez, 1999).

Another Option: Markov Networks

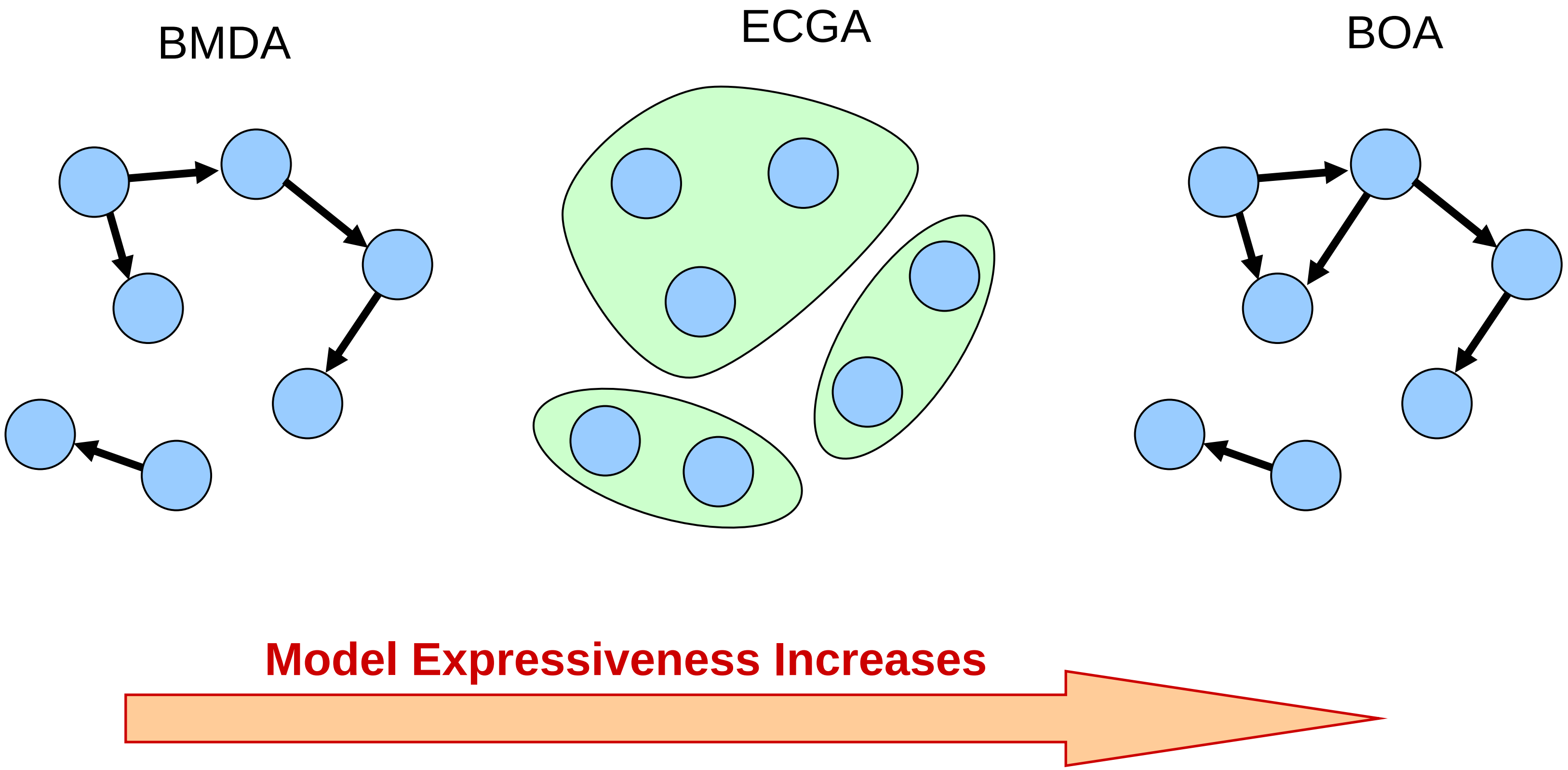
- MN-FDA, MN-EDA (Santana; 2003, 2005)
- Similar to Bayes nets but with undirected edges.



Yet Another Option: Dependency Networks

- Estimation of dependency networks algorithm (EDNA)
  - Gamez, Mateo, Puerta (2007).
  - Use dependency network as a model.
  - Dependency network learned from pairwise interactions.
  - Use Gibbs sampling to generate new solutions.
- Dependency network
  - Parents of a variable= all variables influencing this variable.
  - Dependency network can contain cycles.

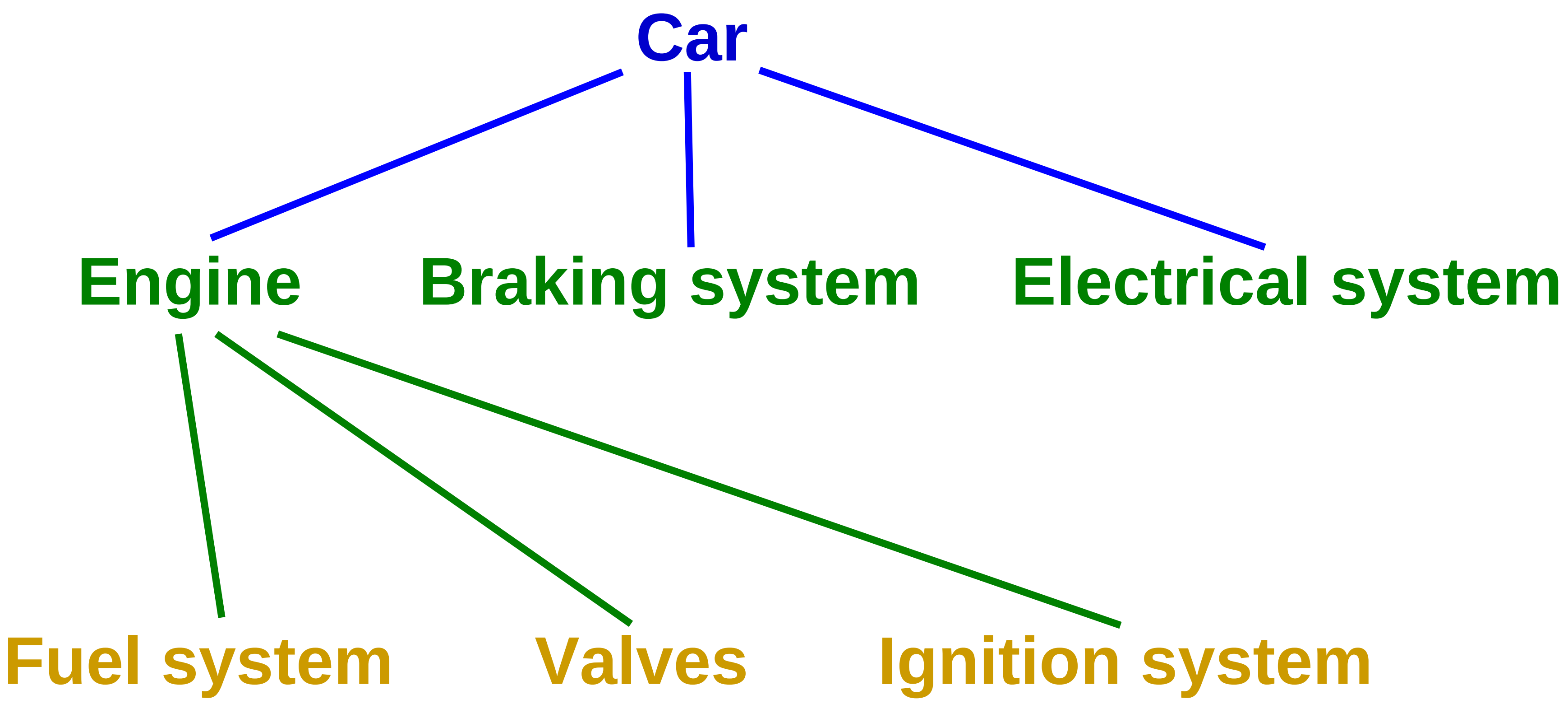
Model Comparison



From single level to hierarchy

- Single-level decomposition powerful.
- But what if single-level decomposition is not enough?
- Learn from humans and nature
  - Decompose problem over multiple levels.
  - Use solutions from lower level as basic building blocks.
  - Solve problem hierarchically.

Hierarchical Decomposition



Three Keys to Hierarchy Success

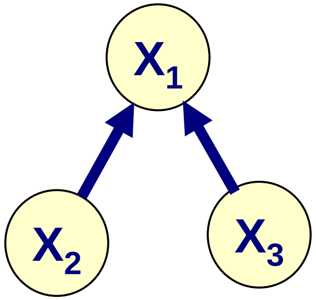
- Proper decomposition
  - Must decompose problem on each level properly.
- Chunking
  - Must represent & manipulate large order solutions.
- Preservation of alternative solutions
  - Must preserve alternative partial solutions (chunks).

Hierarchical BOA (hBOA)

- Pelikan & Goldberg (2000, 2001)
- Proper decomposition
  - Use Bayesian networks like BOA.
- Chunking
  - Use local structures in Bayesian networks.
- Preservation of alternative solutions.
  - Use restricted tournament replacement (RTR).
  - Can use other niching methods.

Local Structures in BNs

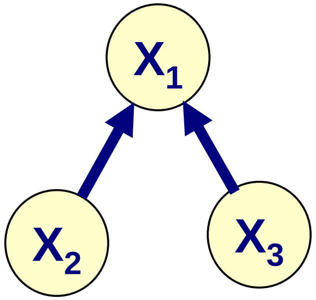
- Look at one conditional dependency.
  - 2^k probabilities for k parents.
- Why not use more powerful representations for conditional probabilities?

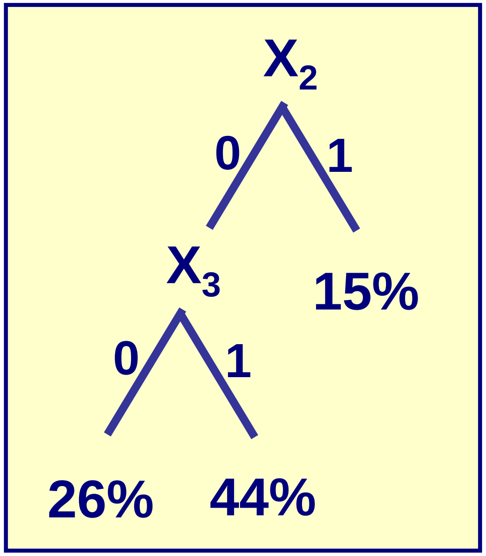


X_2X_3	$P(X_1=0\|X_2X_3)$
00	26 %
01	44 %
10	15 %
11	15 %

Local Structures in BNs

- Look at one conditional dependency.
  - 2^k probabilities for k parents.
- Why not use more powerful representations for conditional probabilities?



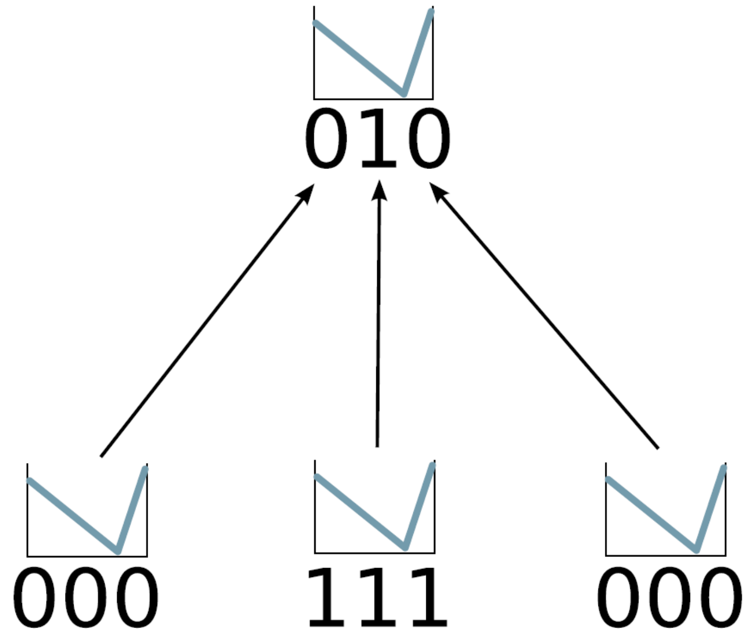


Restricted Tournament Replacement

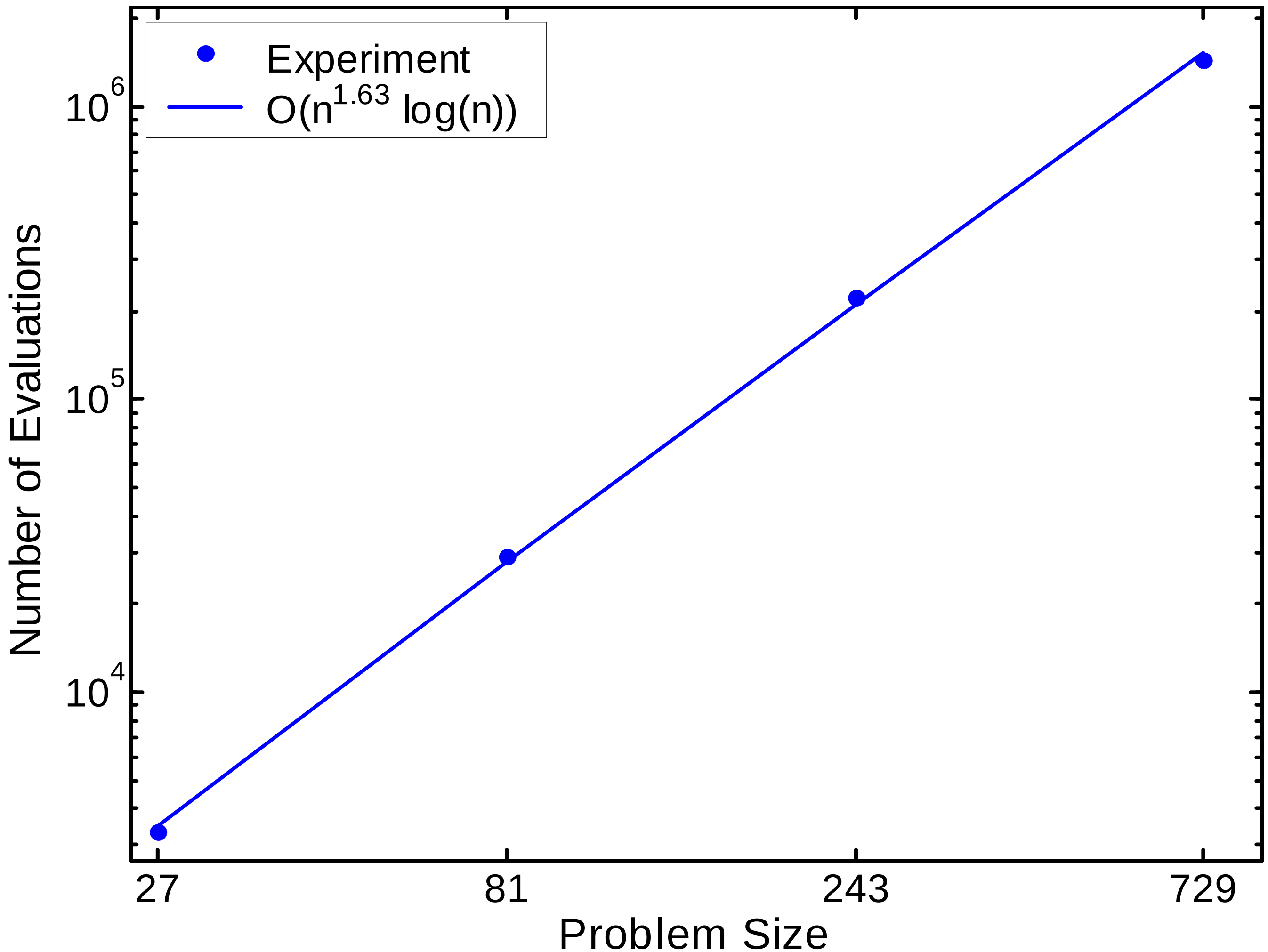
- Used in hBOA for niching.
- Insert each new candidate solution x like this:
  - Pick random subset of original population.
  - Find solution y most similar to x in the subset.
  - Replace y by x if x is better than y.

Hierarchical Traps: The Ultimate Test

- Combine traps on more levels.
- Each level contributes to fitness.
- Groups of bits map to next level.



hBOA on Hierarchical Traps



PMBGAs Are Not Just Optimizers

- PMBGAs provide us with two things
  - Optimum or its approximation.
  - Sequence of probabilistic models.
- Probabilistic models
  - Encode populations of increasing quality.
  - Tell us a lot about the problem at hand.
  - Can we use this information?

Efficiency Enhancement for PMBGAs

- Sometimes $O(n^2)$ is not enough
  - High-dimensional problems (1000s of variables)
  - Expensive evaluation (fitness) function
- Solution
  - Efficiency enhancement techniques

Efficiency Enhancement Types

- 7 efficiency enhancement types for PMBGAs
  - Parallelization
  - Hybridization
  - Time continuation
  - Fitness evaluation relaxation
  - Prior knowledge utilization
  - Incremental and sporadic model building
  - Learning from experience

Multi-objective PMBGAs

- Methods for multi-objective GAs adopted
  - Multi-objective hBOA (from NSGA-II and hBOA)
    (Khan, Goldberg, & Pelikan, 2002)
    (Pelikan, Sastry, & Goldberg, 2005)
  - Another multi-objective BOA (from SPEA2)
    (Laumanns, & Ocenasek, 2002)
  - Multi-objective mixture-based IDEAs
    (Thierens, & Bosman, 2001)
  - Regularity Model Based Multiobjective EDA (RM-MEDA)
    (Zhang, Zhou, Jin, 2008)

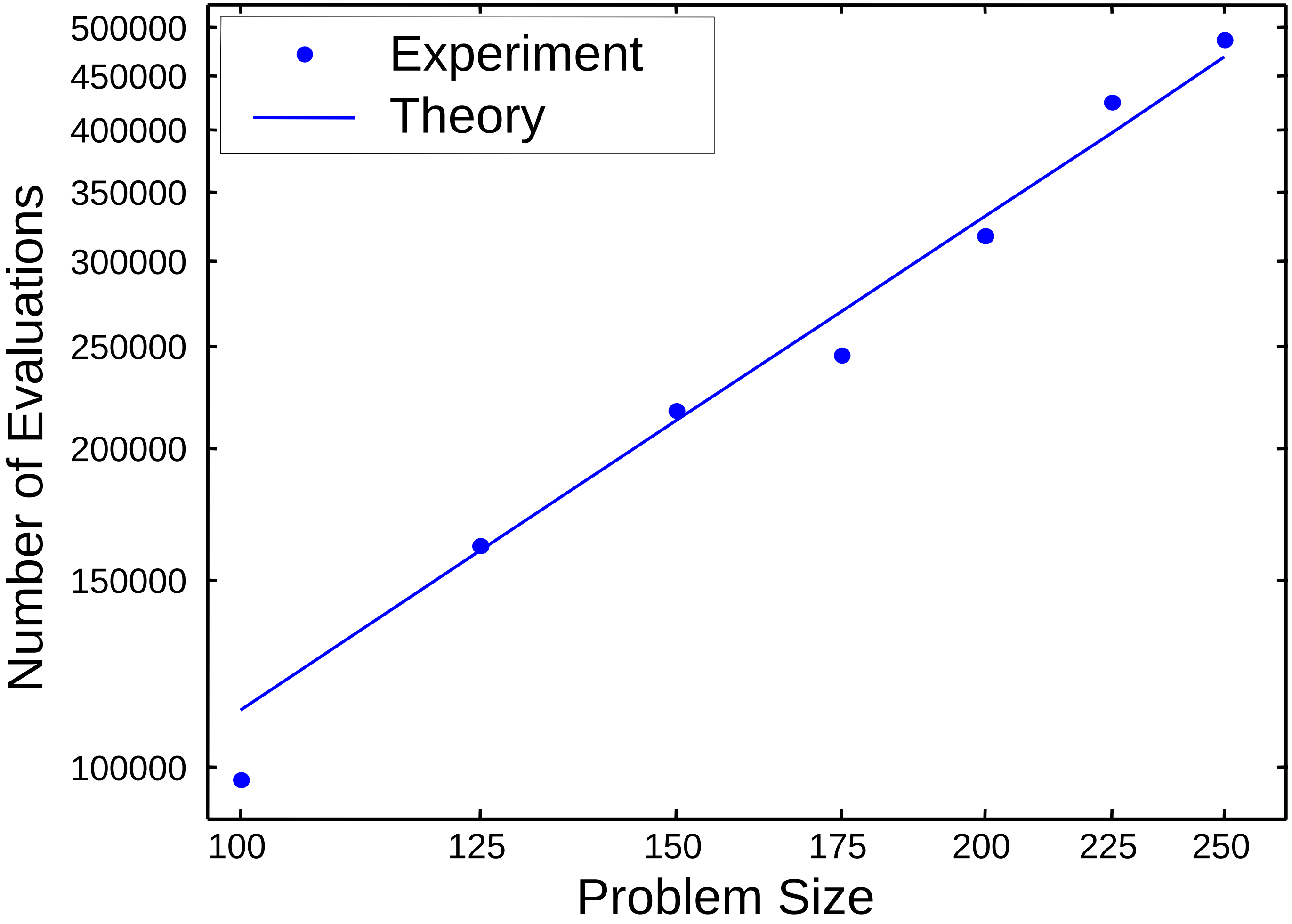
Promising Results with Discrete PMBGAs

- Artificial classes of problems
- Physics
- Bioinformatics
- Computational complexity and AI
- Others

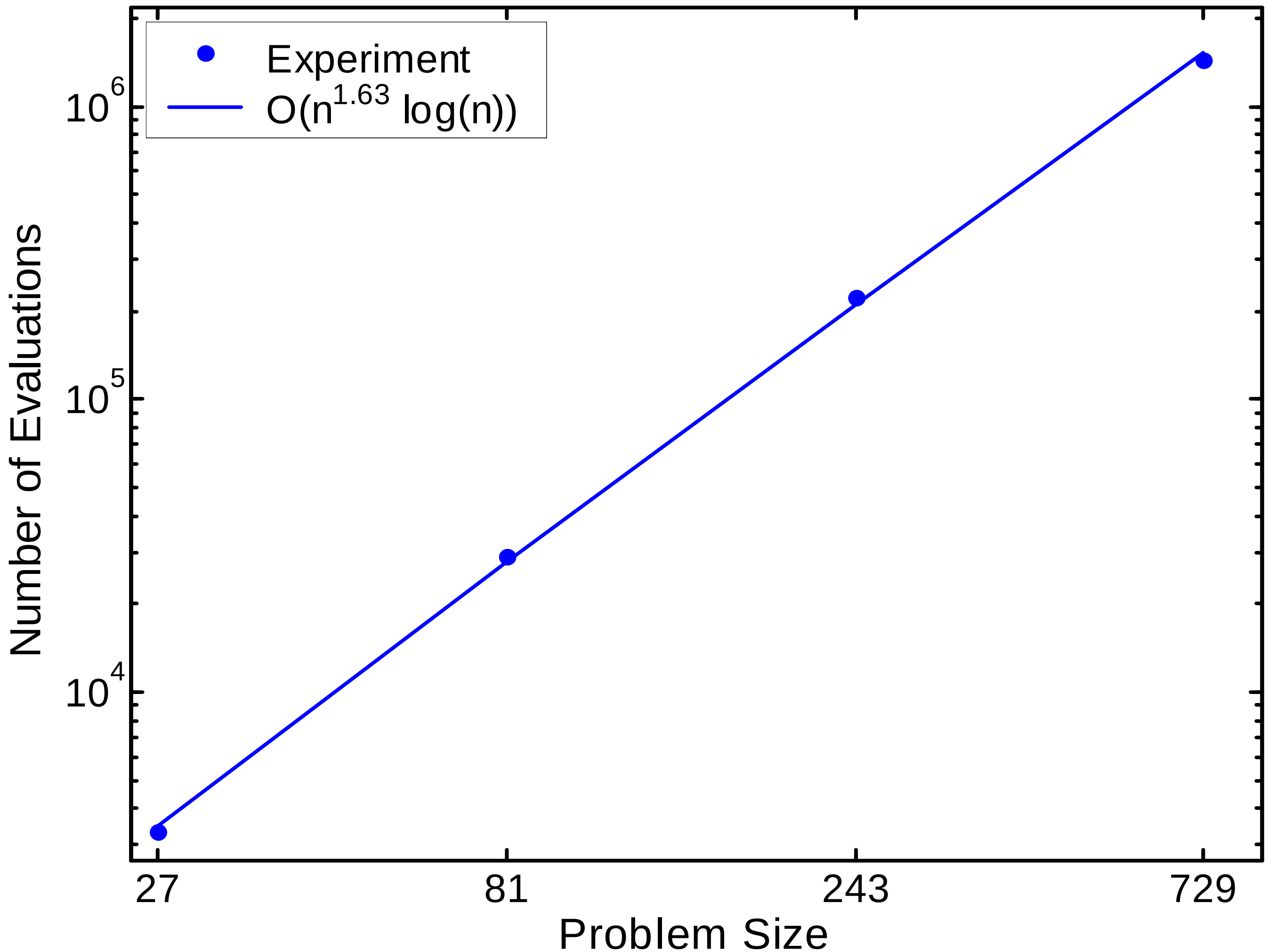
Results: Artificial Problems

- Decomposition
  - Concatenated traps (Pelikan et al., 1998).
- Hierarchical decomposition
  - Hierarchical traps (Pelikan, Goldberg, 2001).
- Other sources of difficulty
  - Exponential scaling, noise (Pelikan, 2002).

BOA on Concatenated 5-bit Traps



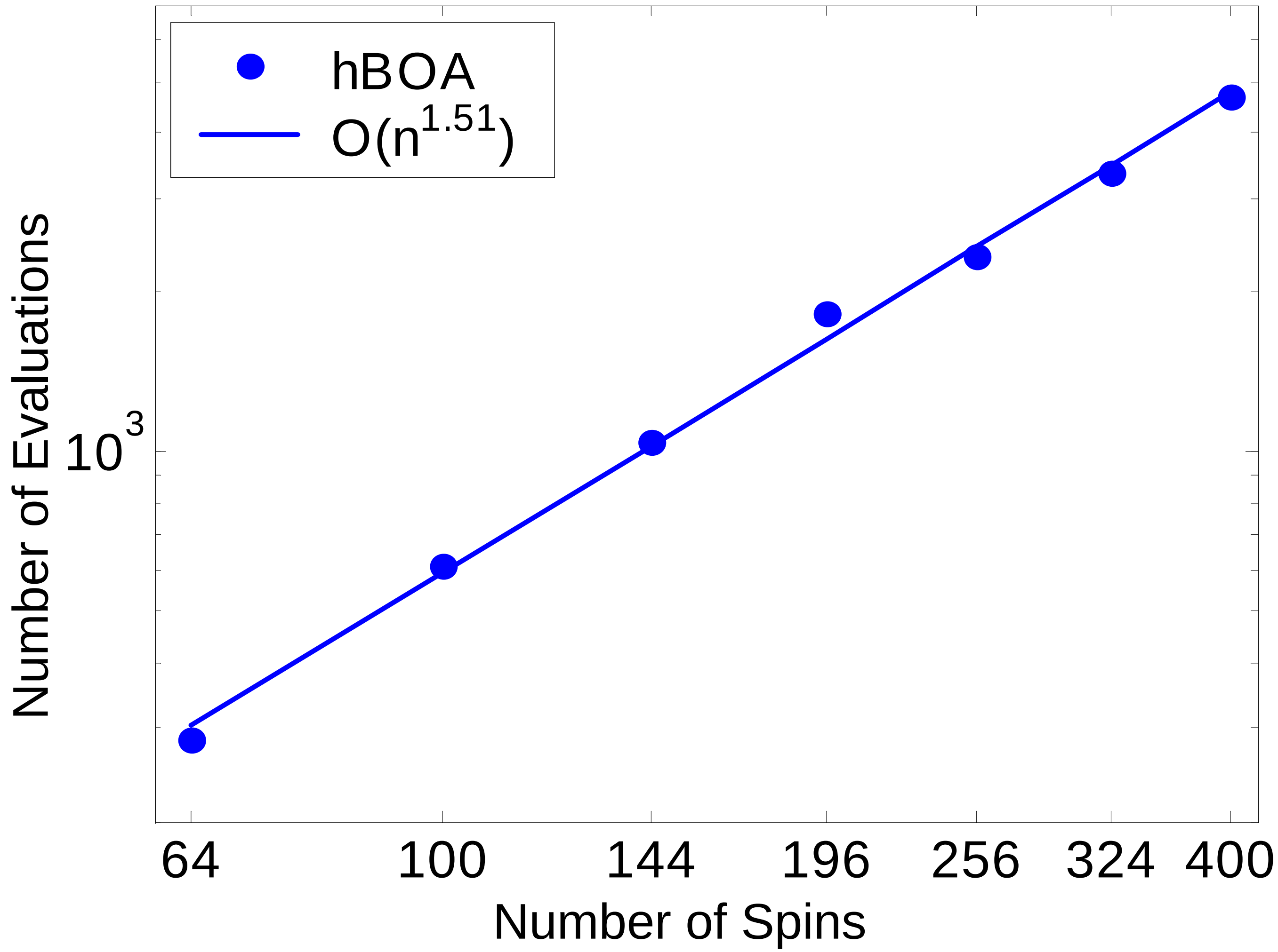
hBOA on Hierarchical Traps



Results: Physics

- Spin glasses (Pelikan et al., 2002, 2006, 2008) (Hoens, 2005) (Santana, 2005) (Shakya et al., 2006)
  - ±J and Gaussian couplings
  - 2D and 3D spin glass
  - Sherrington-Kirkpatrick (SK) spin glass
- Silicon clusters (Sastry, 2001)
  - Gong potential (3-body)

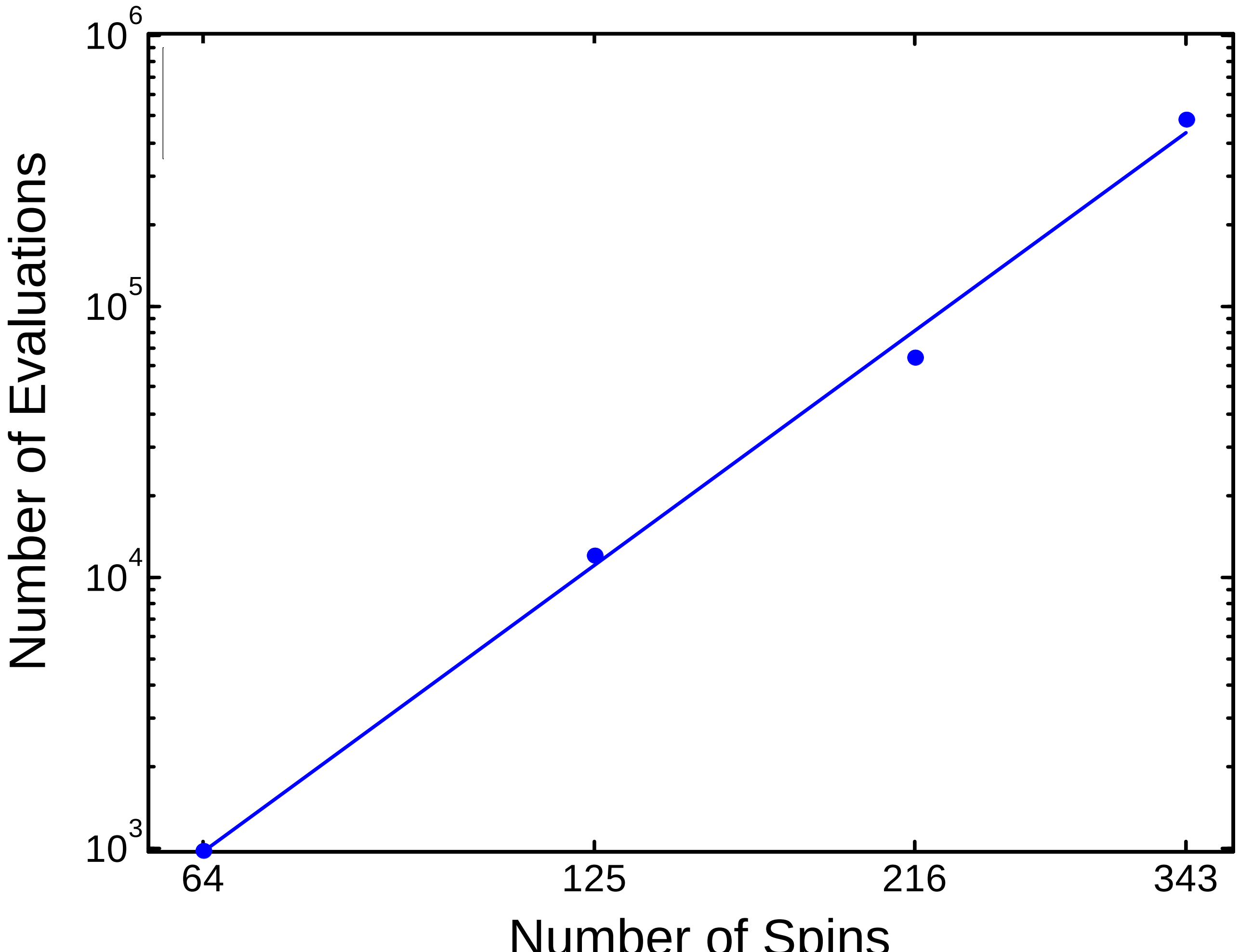
hBOA on Ising Spin Glasses (2D)



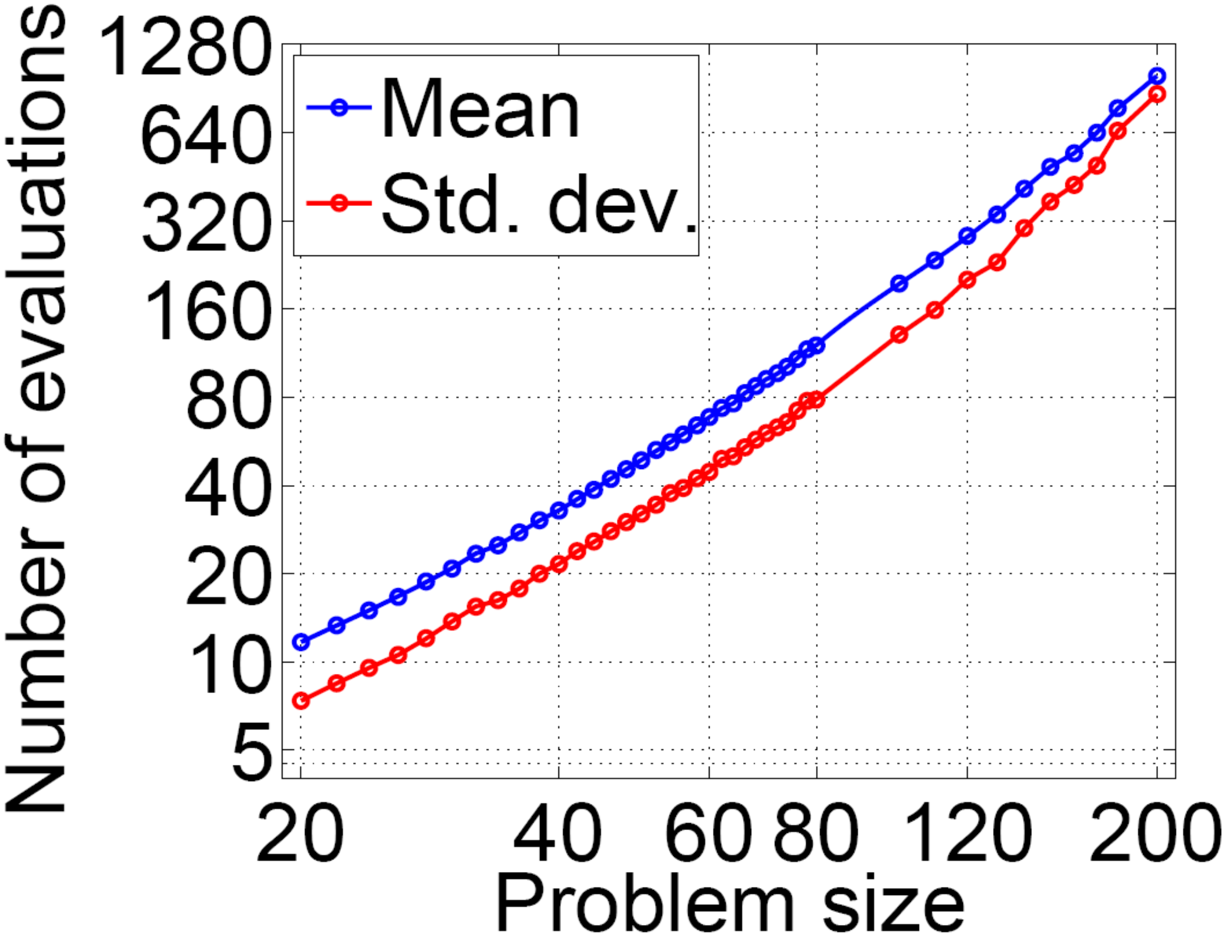
Results on 2D Spin Glasses

- Number of evaluations is O($n^{1.51}$).
- Overall time is O($n^{3.51}$).
- Compare O($n^{3.51}$) to O($n^{3.5}$) for best method (Galluccio & Loebl, 1999)
- Great also on Gaussians.

hBOA on Ising Spin Glasses (3D)



hBOA on SK Spin Glass



Results: Computational Complexity, AI

- MAXSAT, SAT (Pelikan, 2002)
  - Random 3CNF from phase transition.
  - Morphed graph coloring.
  - Conversion from spin glass.

- Feature subset selection (Inza et al., 2001) (Cantu-Paz, 2004)

Results: Some Others

- Military antenna design (Santarelli et al., 2004)
- Groundwater remediation design (Arst et al., 2004)
- Forest management (Ducheyne et al., 2003)
- Nurse scheduling (Li, Aickelin, 2004)
- Telecommunication network design (Rothlauf, 2002)
- Graph partitioning (Ocenasek, Schwarz, 1999; Muehlenbein, Mahnig, 2002; Baluja, 2004)
- Portfolio management (Lipinski, 2005, 2007)
- Quantum excitation chemistry (Sastry et al., 2005)
- Maximum clique (Zhang et al., 2005)
- Cancer chemotherapy optimization (Petrovski et al., 2006)
- Minimum vertex cover (Pelikan et al., 2007)
- Protein folding (Santana et al., 2007)
- Side chain placement (Santana et al., 2007)

Discrete PMBGAs: Summary

- No interactions
  - Univariate models; PBIL, UMDA, cGA.
- Some pairwise interactions
  - Tree models; COMIT, MIMIC, BMDA.
- Multivariate interactions
  - Multivariate models: BOA, EBNA, LFDA.
- Hierarchical decomposition
  - hBOA

Discrete PMBGAs: Recommendations

- Easy problems
  - Use univariate models; PBIL, UMDA, cGA.
- Somewhat difficult problems
  - Use bivariate models; MIMIC, COMIT, BMDA.
- Difficult problems
  - Use multivariate models; BOA, EBNA, LFDA.
- Most difficult problems
  - Use hierarchical decomposition; hBOA.

Real-Valued PMBGAs

- New challenge
  - Infinite domain for each variable.
  - How to model?

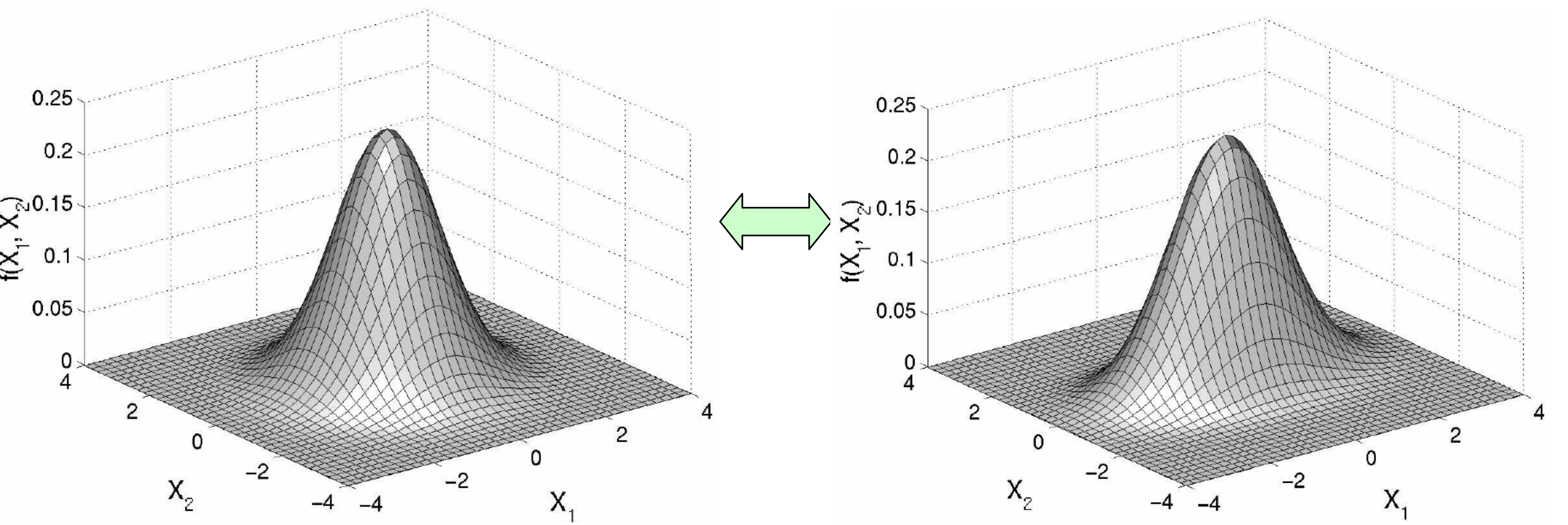
- 2 approaches
  - Discretize and apply discrete model/PMBGA
  - Create model for real-valued variables
    - Estimate pdf.

PBIL Extensions: First Step

- SHCwL: Stochastic hill climbing with learning (Rudlof, Köppen, 1996).
- Model
  - Single-peak Gaussian for each variable.
  - Means evolve based on parents (promising solutions).
  - Deviations equal, decreasing over time.
- Problems
  - No interactions.
  - Single Gaussians=can model only one attractor.
  - Same deviations for each variable.

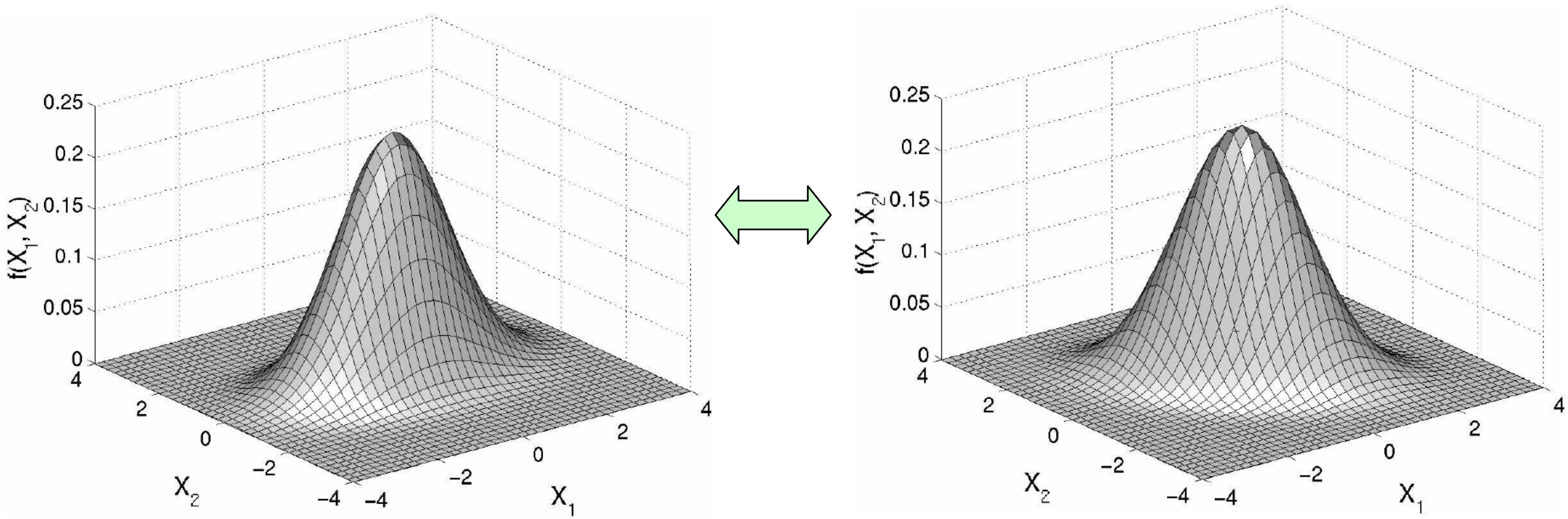
Use Different Deviations

- Sebag, Ducoulombier (1998)
- Some variables have higher variance.
- Use special standard deviation for each



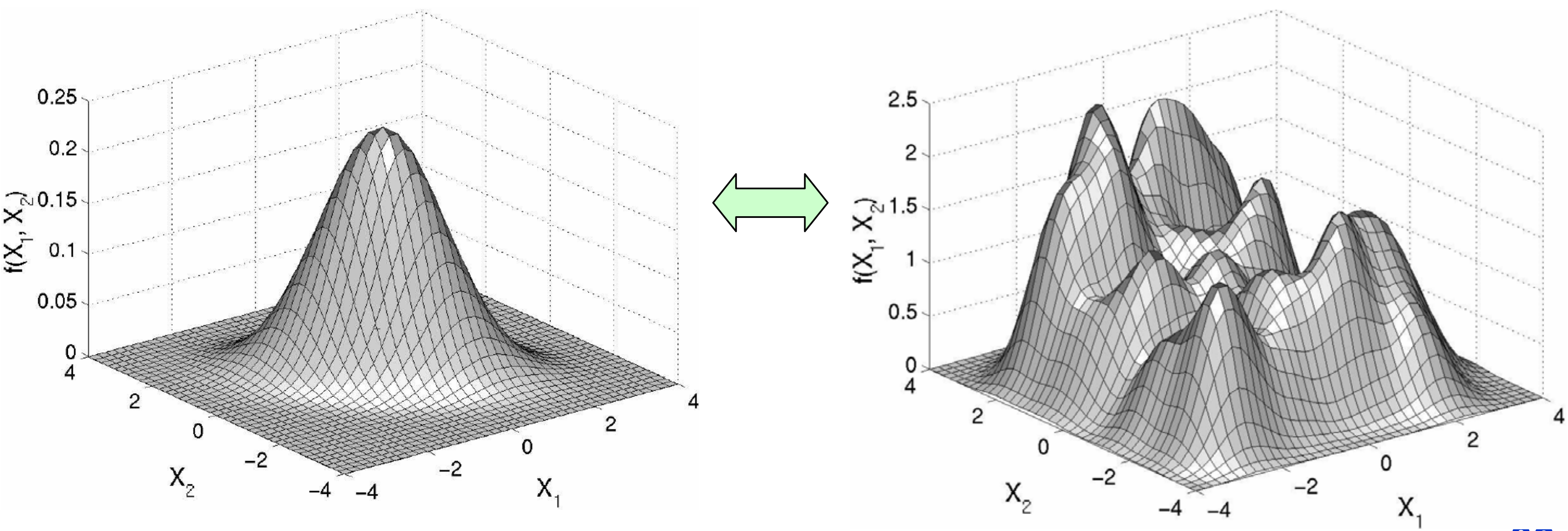
Use Covariance

- Covariance allows rotation of 1-peak Gaussians.
- EGNA (Larrañaga et al., 2000)
- IDEA (Bosman, Thierens, 2000)



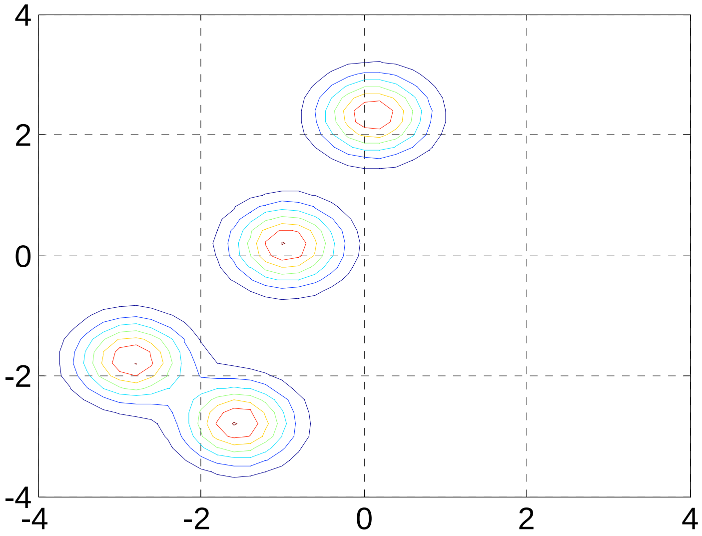
How Many Peaks?

- One Gaussian vs. kernel around each point.
- Kernel distribution similar to ES.
- IDEA (Bosman, Thierens, 2000)



Mixtures: Between One and Many

- Mixture distributions provide transition between one Gaussian and Gaussian kernels.
- Mixture types
  - Over one variable.
    - Gallagher, Frean, & Downs (1999).
  - Over all variables.
    - Pelikan & Goldberg (2000).
    - Bosman & Thierens (2000).
  - Over partitions of variables.
    - Bosman & Thierens (2000).
    - Ahn, Ramakrishna, and Goldberg (2004).



Mixed BOA (mBOA)

- Mixed BOA (Ocenasek, Schwarz, 2002)
- Local distributions
  - A decision tree (DT) for every variable.
  - Internal DT nodes encode tests on other variables
    - Discrete: Equal to a constant
    - Continuous: Less than a constant
  - Discrete variables:
    DT leaves represent probabilities.
  - Continuous variables:
    DT leaves contain a normal kernel distribution.

Real-Coded BOA (rBOA)

- Ahn, Ramakrishna, Goldberg (2003)
- Probabilistic Model
  - Underlying structure: Bayesian network
  - Local distributions: Mixtures of Gaussians
- Also extended to multiobjective problems (Ahn, 2005)

Aggregation Pheromone System (APS)

- Tsutsui (2004)
- Inspired by aggregation pheromones
- Basic idea
  - Good solutions emit aggregation pheromones
  - New candidate solutions based on the density of aggregation pheromones
  - Aggregation pheromone density encodes a mixture distribution

Adaptive Variance Scaling

- Adaptive variance in mBOA
  - Ocenasek et al. (2004)
- Normal IDEAs
  - Bosman et al. (2006, 2007)
  - Correlation-triggered adaptive variance scaling
  - Standard-deviation ratio (SDR) triggered variance scaling

Real-Valued PMBGAs: Discretization

- Idea: Transform into discrete domain.
- Fixed models
  - 2^k equal-width bins with k-bit binary string.
  - Goldberg (1989).
  - Bosman & Thierens (2000); Pelikan et al. (2003).
- Adaptive models
  - Equal-height histograms of 2k bins.
  - k-means clustering on each variable.
  - Pelikan, Goldberg, & Tsutsui (2003); Cantu-Paz (2001).

Real-Valued PMBGAs: Summary

- Discretization
  - Fixed
  - Adaptive
- Real-valued models
  - Single or multiple peaks?
  - Same variance or different variance?
  - Covariance or no covariance?
  - Mixtures?
  - Treat entire vectors, subsets of variables, or single variables?

Real-Valued PMBGAs: Recommendations

- Multimodality?
  - Use multiple peaks.
- Decomposability?
  - All variables, subsets, or single variables.
- Strong linear dependencies?
  - Covariance.
- Partial differentiability?
  - Combine with gradient search.

PMBGP (Genetic Programming)

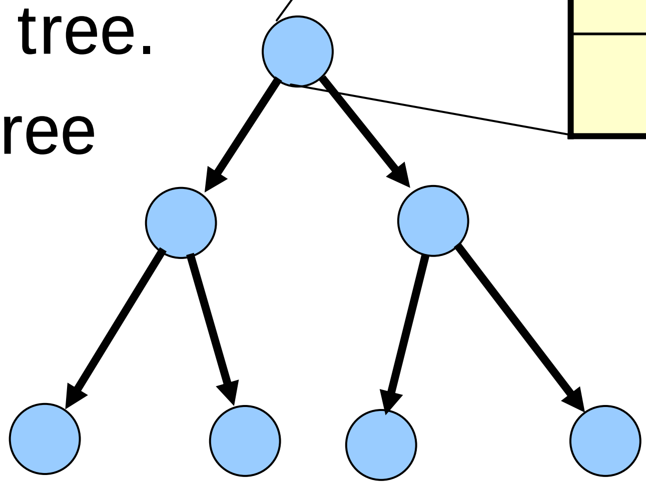
- New challenge
  - Structured, variable length representation.
  - Possibly infinitely many values.
  - Position independence (or not).
  - Low correlation between solution quality and solution structure (Looks, 2006).

- Approaches
  - Use explicit probabilistic models for trees.
  - Use models based on grammars.

PIPE

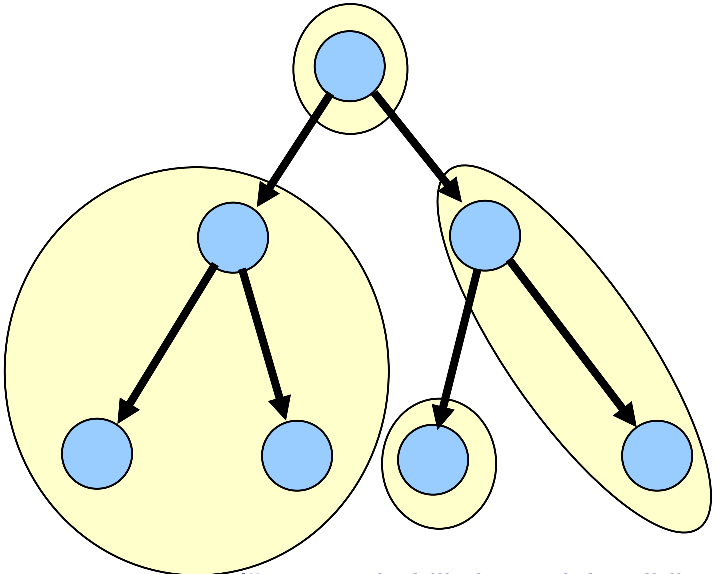
- Probabilistic incremental program evolution (Salustowicz & Schmidhuber, 1997)
- Store frequencies of operators/terminals in nodes of a *maximum* tree.
- Sampling generates tree from top to bottom

X	P(X)
sin	0.15
+	0.35
-	0.35
X	0.15



eCGP

- Sastry & Goldberg (2003)
- ECGA adapted to program trees.
- Maximum tree as in PIPE.
- But nodes partitioned into groups.



BOA for GP

- Looks, Goertzel, & Pennachin (2004)
- Combinatory logic + BOA
  - Trees translated into uniform structures.
  - Labels only in leaves.
  - BOA builds model over symbols in different nodes.
- Complexity build-up
  - Modeling limited to max. sized structure seen.
  - Complexity builds up by special operator.

MOSES

- Looks (2006).
- Evolve demes of programs.
- Each deme represents similar structures.
- Apply PMBGA to each deme (e.g. hBOA).
- Introduce new demes/delete old ones.
- Use normal forms to reduce complexity.

PMBGP with Grammars

- Use grammars/stochastic grammars as models.
- Grammars restrict the class of programs.

- Some representatives
  - Program evolution with explicit learning (Shan et al., 2003)
  - Grammar-based EDA for GP (Bosman, de Jong, 2004)
  - Stochastic grammar GP (Tanev, 2004)
  - Adaptive constrained GP (Janikow, 2004)

PMBGP: Summary

- Interesting starting points available.
- But still lot of work to be done.
- Much to learn from discrete domain, but some completely new challenges.
- Research in progress

PMBGAs for Permutations

- New challenges
  - Relative order
  - Absolute order
  - Permutation constraints
- Two basic approaches
  - Random-key and real-valued PMBGAs
  - Explicit probabilistic models for permutations

Random Keys and PMBGAs

- Bengoetxea et al. (2000); Bosman et al. (2001)
- Random keys (Bean, 1997)
  - Candidate solution = vector of real values
  - Ascending ordering gives a permutation
- Can use any real-valued PMBGA (or GEA)
  - IDEAs (Bosman, Thierens, 2002)
  - EGNA (Larranaga et al., 2001)
- Strengths and weaknesses
  - Good: Can use any real-valued PMBGA.
  - Bad: Redundancy of the encoding.

Direct Modeling of Permutations

- Edge-histogram based sampling algorithm (EHBSA) (Tsutsui, Pelikan, Goldberg, 2003)
  - Permutations of *n* elements
  - Model is a matrix $A=(a_{i,j})_{i,j=1, 2, ..., n}$
  - $a_{i,j}$ represents the probability of edge (i, j)
  - Uses template to reduce exploration
  - Applicable also to scheduling

ICE: Modify Crossover from Model

- ICE
  - Bosman, Thierens (2001).
  - Represent permutations with random keys.
  - Learn multivariate model to factorize the problem.
  - Use the learned model to modify crossover.
- Performance
  - Typically outperforms IDEAs and other PMBGAs that learn and sample random keys.

Multivariate Permutation Models

- Basic approach
  - Use any standard multivariate discrete model.
  - Restrict sampling to permutations in some way.
  - Bengoetxea et al. (2000), Pelikan et al. (2007).
- Strengths and weaknesses
  - Use explicit multivariate models to find regularities.
  - High-order alphabet requires big samples for good models.
  - Sampling can introduce unwanted bias.
  - Inefficient encoding for only relative ordering constraints, which can be encoded simpler.

Conclusions

- Competent PMBGAs exist
  - Scalable solution to broad classes of problems.
  - Solution to previously intractable problems.
  - Algorithms ready for new applications.
- PMBGAs do more than just solve the problem
  - They provide us with sequences of probabilistic models.
  - The probabilistic models tell us a lot about the problem.
- Consequences for practitioners
  - Robust methods with few or no parameters.
  - Capable of learning how to solve problem.
  - But can incorporate prior knowledge as well.
  - Can solve previously intractable problems.

Starting Points

- World wide web
- Books and surveys
  - Larrañaga & Lozano (eds.) (2001). Estimation of distribution algorithms: A new tool for evolutionary computation. Kluwer.
  - Pelikan et al. (2002). A survey to optimization by building and using probabilistic models. Computational optimization and applications, 21(1), pp. 5-20.
  - Pelikan (2005). Hierarchical BOA: Towards a New Generation of Evolutionary Algorithms. Springer.
  - Lozano, Larrañaga, Inza, Bengoetxea (2007). Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms, Springer.
  - Pelikan, Sastry, Cantu-Paz (eds.) (2007). Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications, Springer.

Online Code (1/2)

- BOA, BOA with decision graphs, dependency-tree EDA
  http://medal.cs.umsl.edu/
- ECGA, xi-ary ECGA, BOA, and BOA with decision trees/graphs
  http://www-illigal.ge.uiuc.edu/
- mBOA
  http://jiri.ocenasek.com/
- PIPE
  http://www.idsia.ch/~rafal/
- Real-coded BOA
  http://www.evolution.re.kr/

Online Code (2/2)

- Demos of APS and EHBSA
  http://www.hannan-u.ac.jp/~tsutsui/research-e.html

- RM-MEDA: A Regularity Model Based Multiobjective EDA
  Differential Evolution + EDA hybrid
  http://cswww.essex.ac.uk/staff/qzhang/mypublication.htm

- Naive Multi-objective Mixture-based IDEA (MIDEA)
  Normal IDEA-Induced Chromosome Elements Exchanger (ICE)
  Normal Iterated Density-Estimation Evolutionary Algorithm (IDEA)
  http://homepages.cwi.nl/~bosman/code.html