Lecture L6:
EAs for Complex Optimization Tasks

- Customized EAs for large scale application
- Multi-modal EAs
- Meta-model assisted EAs

Large-Scale Optimization:
Handling a large number of variables

- Large n, large pop-size, large computation
- Customized EAs
  - Representation
  - Operators
- EA's flexibility shows promise
- A case study involving millions of variables (Deb and Reddy, 2001)

Casting Scheduling

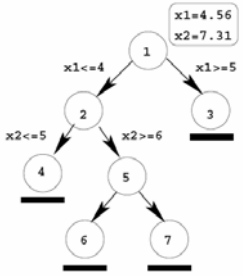


Casting Scheduling Problem (cont.)

- Maximum metal utilization:

$$\frac{1}{H}\sum_{i=1}^{H}\frac{100\sum_{k=1}^{K}w_k x_{ki}}{W_i}$$

- Constraints:
  - Demand satisfaction: $\sum_{i=1}^{H} x_{ki} = r_k$ for $k = 1, \ldots, K$
  - Capacity constraint: $\sum_{k=1}^{K} w_k x_{ki} \leq W_i$ for $i = 1, \ldots, H$



- An integer linear program (ILP)
- Branch-and-bound: exponential algorithm

Performance of LINGO

- Works up to n=500 on a Pentium IV (7 hrs.)

LINGO MILP Solver

Heat No.	Order Number										Utilization/ Cruc. Size	Efficiency (%)
	1	2	3	4	5	6	7	8	9	10		
1	0	1	1	0	0	0	2	1	0	0	623/650	95.85
2	2	0	0	0	1	0	0	0	2	0	615/650	94.62
3	1	0	0	1	3	1	0	0	0	0	611/650	94.00
4	2	0	0	0	1	0	0	1	0	0	645/650	99.23
5	0	0	0	1	0	2	0	0	1	6	612/650	94.15
6	1	1	0	0	2	1	0	0	0	0	591/650	90.92
7	0	0	2	2	1	0	0	0	2	0	585/650	90.00
8	0	3	0	0	0	1	0	0	1	0	611/650	94.00
9	0	2	3	0	1	0	0	0	0	0	650/650	100.00
10	1	0	0	5	0	0	0	0	1	0	635/650	97.69
	7	7	6	9	9	5	2	2	7	6	Average	95.05

- But does not work beyond

Off-The-Shelf EA Results

Number of Variables	Binary-coded GAs			Real-coded GAs		
	Population Size	Efficiency	Function Eval.	Population Size	Efficiency	Function Eval.
100	100	96.15	13,600	100	95.94	23,740
200	300	95.01	1,42,200	200	92.81	1,21,760
300	1,000	90.11	14,12,400	700	95.14	5,84,220

- Exponential function evaluations
- Random initialization, standard crossover and mutations are not enough
- Standard EA practice is too generic
- Need a customized EA

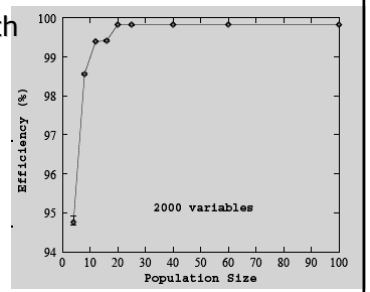
A Customized EA

- Custom initialization
  - Make random assignment (0,a)
  - Custom recombination
  - Normalize to satisfy equality constraint
- Custom recombination
  - Heat by heat construct offspring from the better parent
    - Feasible heat wins over infeasible
    - Lesser violation of utilization constraint wins
    - Better utilization wins, in case both are feasible
- Custom mutation: Fix the infeasibility of offspring

Effect of Population Size

- N=2000 variables with max. gen.=1000/N
- A critical population size is needed

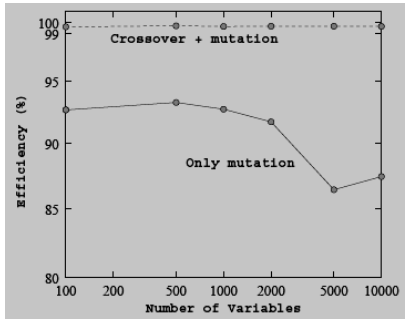
N:	4	8	12	16
# Heats:	211	204	201	201
N:	20	25	40	100
# Heats:	200	200	200	200



- Total metal required 113,280 kg
- 1 less heat: (90X650+109X500)=113,000 kg

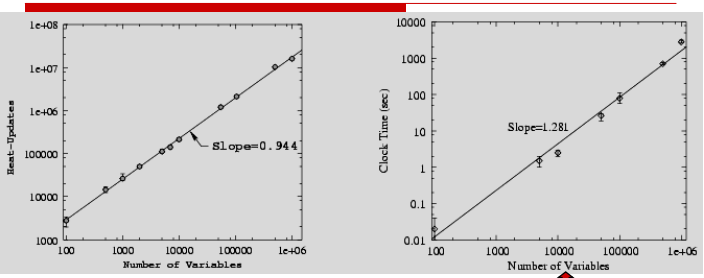
Effect of Recombination Operator

- Compared for a fixed number of generations
- Recombination operator is useful



Scale-Up Results

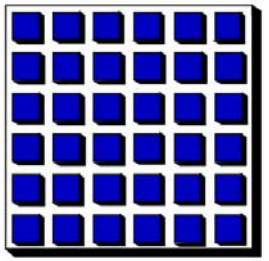
Solutions within a maximum of 200 gm error



- Customized GA has sub-quadratic complexity and up to one million variables (Deb et al., 2003)
- Never to our knowledge such a large problem was solved using EAs before 2003

Other Customizations: Innovative Representation

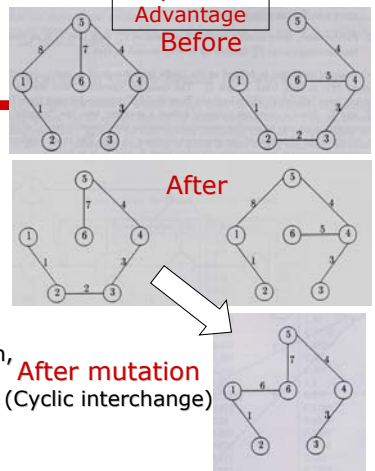
Operator Advantage



- An important matter in EA application
- Permutation as fixed coding
- Permutation: (a-b-c-d-e)
- Fixed coding:
- (0, 0-1, 0-2, 0-3, 0-4)
- Example: (0 1 0 2 2)
  _a_-> _c_a_b_ -> _c_a_d_b_-> (c-a-e-d-b)
- Allow any crossover to be used
- Valid permutations are created (Deb et al., 2003, IEEE)

EA Operators for Graphs

Operator Advantage

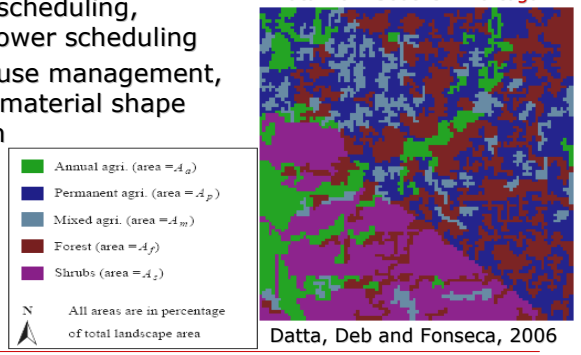


- Problem-specific operators
- Graphs for SCM, network, process flow, job-shop scheduling
- Sensor network design (Sen, Narshiman, Deb, 1998)
- Preserve graph properties

Contiguity Preserving Operator: Resource Allocation Problems

Operator Advantage

Data from Southern Portugal



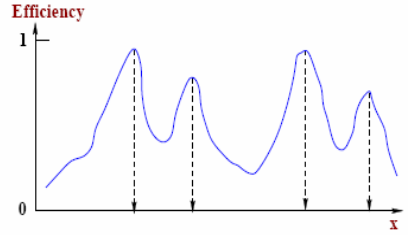
- Crew scheduling, manpower scheduling
- Land-use management, multi-material shape design

Datta, Deb and Fonseca, 2006

Multi-Modal Optimization:

Handling multiple optimal solutions

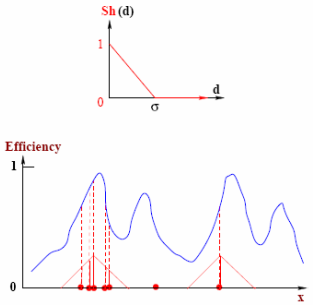
- To solve problems with multiple local/global optimum
- Classical methods can find only one optimum at a time
- EAs can, in principle, find multiple optima simultaneously, due to their population approach



Niching Concept

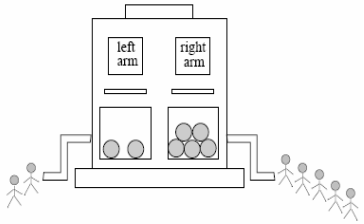
Multiple niches (human and animal) coexist in nature

- By sharing resources (land, food, etc.)
- How to mimmick the concept in EAs?
- Reduce selection pressure for crowded solutions
- Use a sharing function based on two-armed bandit problem



Two-Armed Bandit Problem

- If we know which arm pays more, all population members crowd there
  - Same as converging to one optima
- Introduce sharing payoff strategy
- All player share payoff with others interested in playing the same arm
- Steady-state distribution: Players in both arms

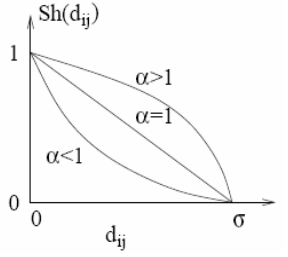


$f_L/m_L = f_R/m_R$

Sharing Function

- Goldberg and Richardson (1997)
- d is a distance measure between two solns.
  - Phenotypic distance: $d(x_i,x_j)$, x: variable
  - Genotypic distance: $d(s_i,s_j)$, s: string
- Calculate niche count, $nc_i=\Sigma_j Sh(d_{ij})$
- Shared fitness: $f_i'=f_i/nc_i$
- Use proportionate selection opeartor

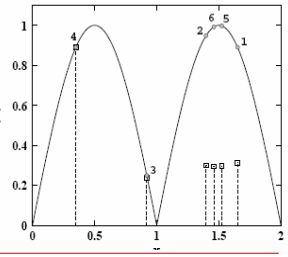
$$Sh(d_{ij}) = \begin{cases} 1-\left(\frac{d_{ij}}{\sigma}\right)^{\alpha} & \text{if } d_{ij} \le \alpha, \\ 0 & \text{otherwise.} \end{cases}$$



An Example: Phenotypic Sharing

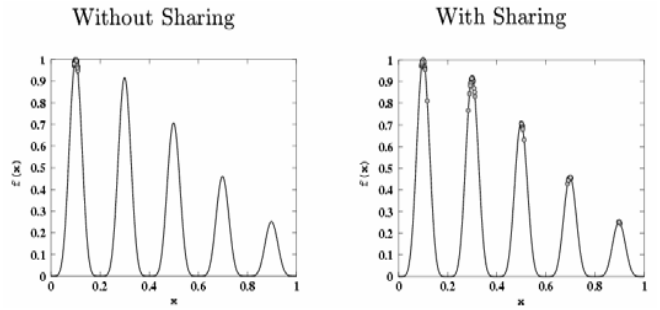
- Maximize $f(x) = |\sin(\pi x)|$, $x \in [0, 2]$
- $nc_i = \sum_{j=1}^{N} Sh(d_{ij})$

Sol. i	String	$x^{(i)}$	f_i	nc_i	f_i'
1	110100	1.651	0.890	2.856	0.312
2	101100	1.397	0.948	3.160	0.300
3	011101	0.921	0.246	1.048	0.235
4	001011	0.349	0.890	1.000	0.890
5	110000	1.524	0.997	3.364	0.296
6	101110	1.460	0.992	3.364	0.295



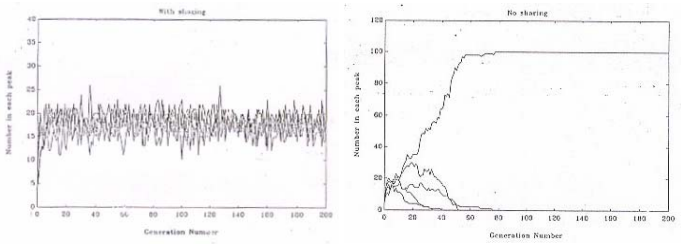
Simulation Results

- Inclusion of niche-formation strategy



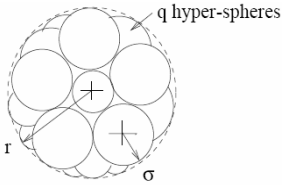
Distribution of Population Among Optima

- With and without sharing



Phenotypic σ_{share}

- Assume number of optima: q
- 1-D space:
  $\sigma_{share} = (x_{max}-x_{min})/2q$
- p-D space:
  - r: Euclidean length of radius of entire p-D hyper-sphere
  - Volume: $V = cr^p$
  - Divide volume into q parts
  - σ_{share} is the radius of each part
  - $V = cr^p = q\, c\sigma_{share}^p$ yields



$$\sigma_{share} = r/q^{1/p}$$

Other Niching Approaches

- Clearing approach
- Clustering approach
- Crowding approach
- Pre-selection approach
- Restrictive tournament selection approach
- All require at least one tunable parameter



Clearing Strategy

- Clear bad near good solutions
  - Sort population according to fitness
  - Keep the best and assign zero fitness to all others within *d* from best
  - Keep the next best and assign zero fitness to others within *d* from next best
  - Continue
  - Perform a selection with assigned fitness
- Petrowski (1996)



Clearing Strategy (cont.)

- Most members are wasted
- A better approach
  - All zero-fitness solutions are re-initialized outside *d* of best
  - Performs better



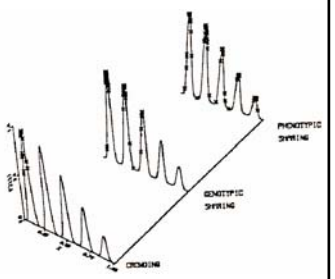
Clustering Approach

- Based on phenotypic space, a clustering is performed
  - With a predefined *d*
  - With a predefined number of clusters
  - Niching within each cluster
  - Mating restriction can also be performed
- Parameter sensitive
- Yin and Germay (1993)



Crowding Approach

- De Jong (1975)
  - Child solution replaces the most similar solution from a subpopulation
  - Subpopulation size
- Mahfoud (1993)
- Maintains diversity
- Not as good as sharing (Deb and Goldberg, 1989)



Preselection Procedure

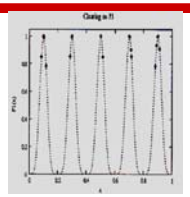
- Similar to crowding
- Cavicchio (1971)
  - Offspring (better than a parent) replaces the most similar parent
- Diversity is maintained
- Both crowding and preselection are weak for multi-modal optimization

Continuously Updated Sharing Approach

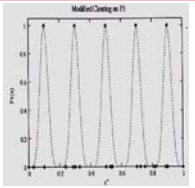
- Sharing approach require proportionate selection
- Update tournament selection
  - On a subpopulation compute shared fitness for each candidate
  - Choose the one with better shared fitness
  - A strategy with niche count values are also suggested for minimization
- Oei et al. (1991)
- Used in other EAs (multi-objective NPGA)

Comparison of Niching Methods

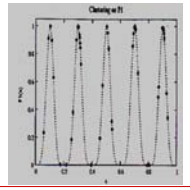
Clearing



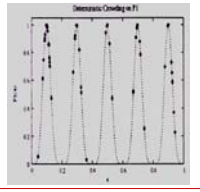
Modified Clearing



Clustering

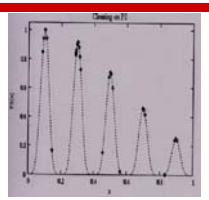


Crowding

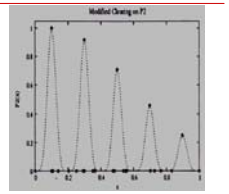


More Comparisons

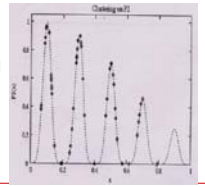
Clearing



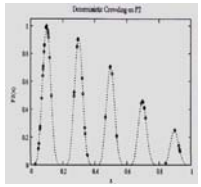
Modified Clearing



Clustering



Crowding



Comparison on a Scalable Problem

- Optima are placed randomly in the search space
- Gaussian shape of optimal basin
- A8 (Mod. Clearing) performs the best
  - A1: Clearing
  - A2: Clustering
  - A4: Crowding

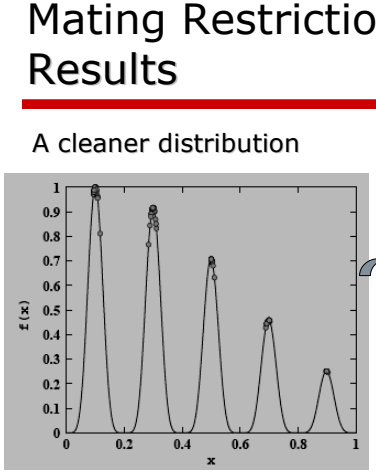
No. of Peaks		A1	A2	A4	A8
20	SR	15.78	19.0	19.28	**20.0**
	SD	4.18	1.44	1.20	**0.0**
	TT	112.6	23.39	120.4	2156.7
	SD	54.69	4.52	69.15	1635.4
30	SR	25. 0	27.38	28.58	**30.0**
	SD	3.02	2.82	1.56	**0.0**
	TT	155.0	36.72	144.8	2526.8
	SD	219.6	3.83	245.6	1854.5
40	SR	35.14	36.24	38.12	**40.0**
	SD	4.50	2.80	2.23	**0.0**
	TT	121.6	120.7	125.0	2967.9
	SD	15.14	68.94	9.23	1797.6
50	SR	44.86	42.2	47.6	**50.0**
	SD	2.53	4.40	2.43	**0.0**
	TT	321.8	222.8	335.6	3326.0
	SD	64.14	60.92	138.9	2142.7

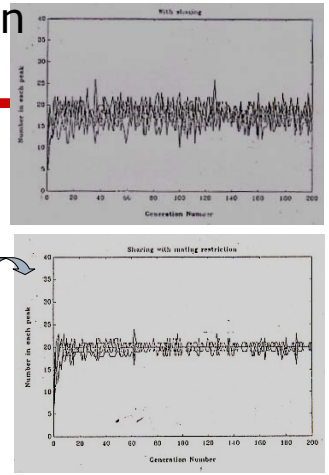
Mating Restriction: Speciation

- Allow mating with intra-species individuals
- Solutions only within σ_{mating} mates to each other (restricted crossover)
  - Does not allow too many lethal solutions
- Causes stable subpopulations in many optima
- Other approaches:
  - Tag-template, taxon-exemplar etc.

Mating Restriction Results

A cleaner distribution





Tag-Template Scheme

- Tag-template scheme
  Tag Templ. Functional String
  0010 : **10: 0100011010110
- Tag of one should match with template of other parent for mating
- They all participate in crossover and mutation
- Artificial evolution of tag-template for individual optimum
- Deb and Goldberg (1989) motivated by Booker's (1982) taxon-exempler idea



Optimization with Meta-Models:

Handling computationally expensive problems

- Evaluation of most real-world problems is computationally expensive
- Optimization algorithm run into days
- Following approaches with EAs:
  - Parallel computing of solutions, easily possible with EA's population approach
    - Almost linear speed-up
  - Use approximate models of objective function and constraints
  - Combination of above



Meta-Model Assisted EAs

- Serial approach
  - A few high-fidelity solutions can be used to make a model and then find optimum of model
- Successive optimization-cum-modeling approach
  - Model, optimization, model, optimization, …
  - Need to switch judiciously, self-adaptive method is desired



Metamodeling Techniques

- Three tasks
  - Generate data using an experimental design method or a computer
  - Choose a model to represent data
  - Fit the model to the data
- Some common techniques
  - Response surface methodology (RSM)
  - Neural networks
  - Kriging
  - Inductive learning



Response Surface Method (RSM)

- Box and Wilson (1951)
- Model: Error is independent of x
  - $y(\mathbf{x}) = g(\mathbf{x}) + \epsilon,\ \epsilon = N(0, \sigma),$
  - $\hat{y}(\mathbf{x}) = g(\mathbf{x})$
- Assume g(x), usually parametric linear or quadratic

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j$$

- β_i determined by least-square regression from observed data
  - Optimize to minimize error: Find mean β_i
  - Variance of β_i determine predictive capability

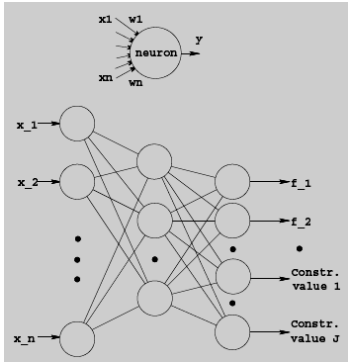
RSM Tips

- Popular and most widely used
- Best suited in applications with random error
- However, limited handling on non-linearity
- Usually applied for k<10
- Sequencial RSM with move limits and trust region approach

Artificial Neural Networks (ANN)

- Mimicking natural neural processing
- Input: Variables (n)
- Output: Objective and constraint functions
- Sigmoidal activation function for hidden and output neurons

$$y = \frac{1}{1 + \exp\left(-(\sum_i w_i x_i + \beta)/T\right)}$$



ANN (cont.)

- Obtain relationship among input-output
- Optimal connection and its weights for minimum error

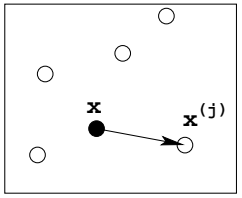
$$E = \|\mathbf{y}_p - \hat{\mathbf{y}}_p\|$$

- A non-linear regression method
- Suitable for large (104) parameters
- Best suited for deterministic problems (no random error)
- Computationally expensive procedure

Kriging Procedure

- D. G. Krige (a geologist): Statistical analysis of mining data
- Predict a value at a point from a given set of observations

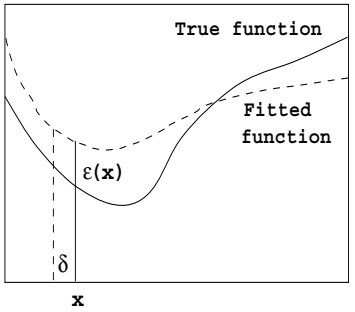
$$f(\mathbf{x}) = \sum_{i=1}^{M} \lambda_i y(\mathbf{x}^{(j)})$$



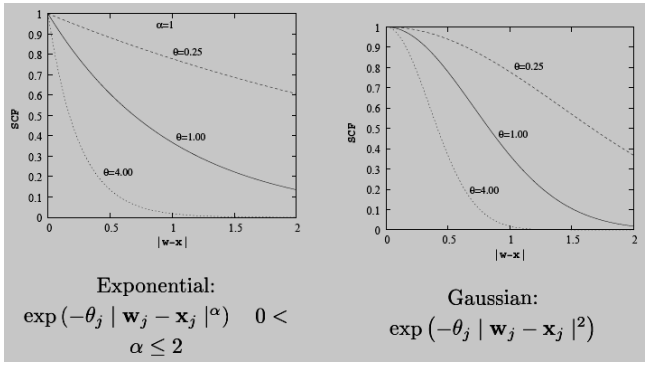
- λ_i depends on distance of X from observed points
- Flexible, but complex
- Suited for k<50, deterministic problems

Fundamentals of Kriging

- F(x) is true function
- f(x) is fitted function
- ε(x) = F(x)-f(x)
- If error at x is large, it is reasonable to believe that
  - Error at x+δ is also large

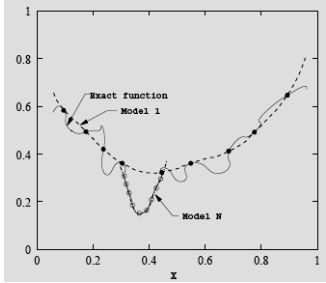


Commonly Used SCFs



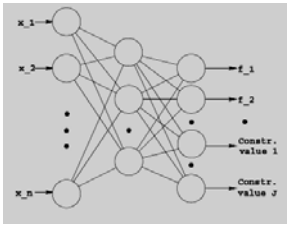
Successive Iterative Procedure

- Nain and Deb (2003)
- Successive approximations to the problem
- Initial coarse approximate model defined over the whole range of decision variables with small database
- Gradual finer approximate models localized in the search space

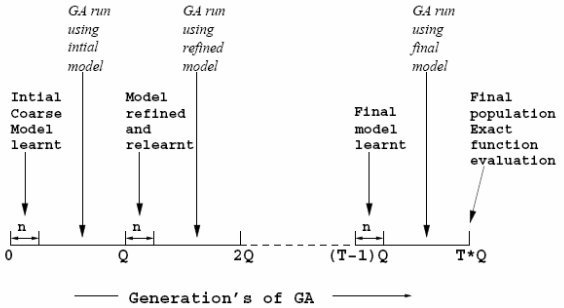


ANN Model

- A feed-forward neural network
- Input: Decision variables (size n)
- Output: Objective functions and constraint violations (size M+J)
- One hidden layer with H neurons
- Sigmoidal activation for hidden and output neurons



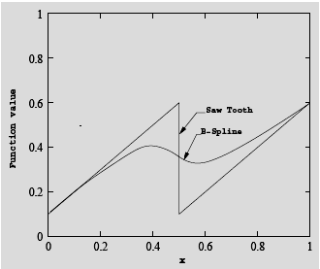
Generation-wise Sketch of Proposed Approach



n/Q fraction of exact evaluations, although the ratio can be made smaller later

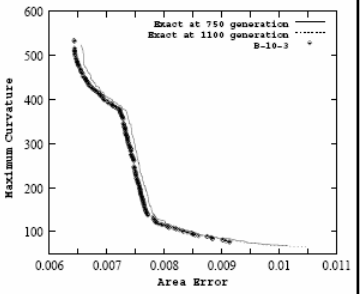
A Case Study from Curve-Fitting

- Minimize error from the curve
- Minimize the maximize curvature
- 41 control points forming a B-spline curve (39 varied)



Curve Fitting Problem (cont.)

- Smoothness in fitting and fitting error
- B-10-3 finds a front in (750x200) evaluations similar to NSGA-II in (1100x200) evaluations
- A saving of 32% evaluations



Another Case Study: Canteliver Plate Design
(Weight and Stiffness)

- B-10-3 is found to be most efficient

