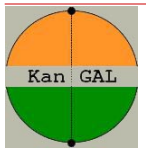
Lecture 5:
Multi-Modal Optimization and Function Approximations

Kalyanmoy Deb
Kanpur Genetic Algorithm Laboratory (KanGAL)
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, INDIA
deb@iitk.ac.in
http://www.iitk.ac.in/kangal/deb.htm

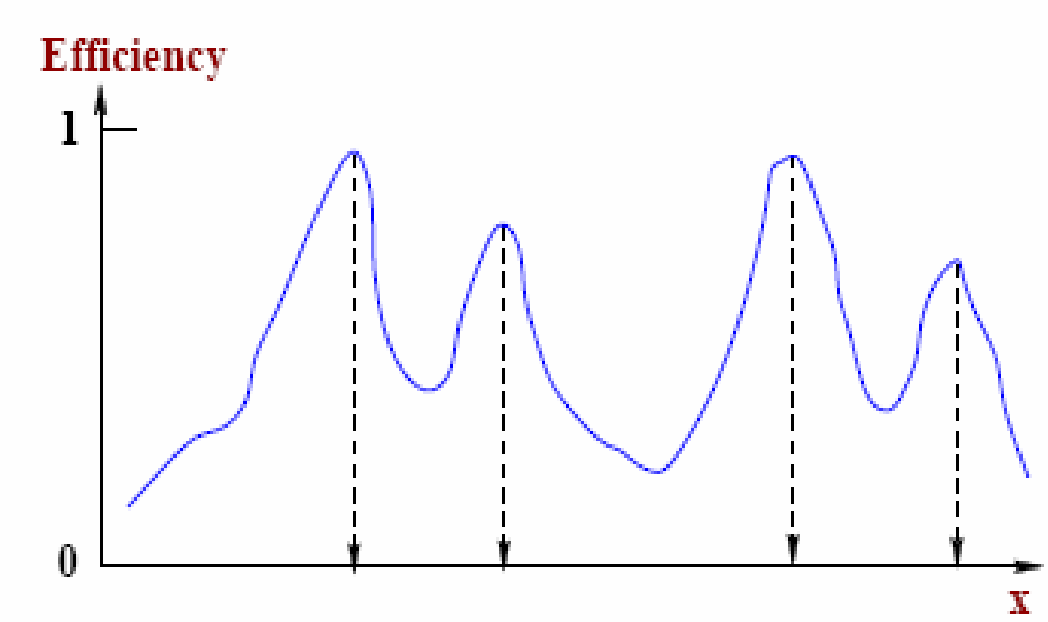


Overview of Lecture 5

- Finding multiple optimal solutions
- Niching methodologies
- Speciation methods
- Omni-Optimizer
- Function approximation techniques
  - Response surface methodology (RSM)
  - Kriging procedure
  - Artificial neural network

Multi-Modal Evolutionary Algorithms

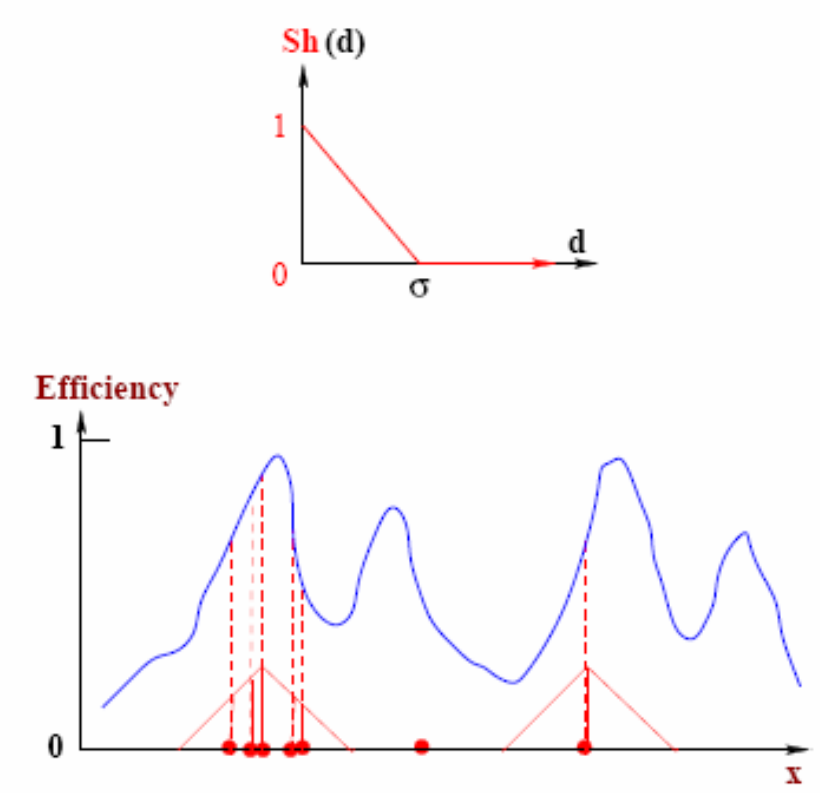
- To solve problems with multiple local/global optimum
- Classical methods can find only one optimum at a time
- EAs can, in principle, find multiple optima simultaneously



Niching Concept

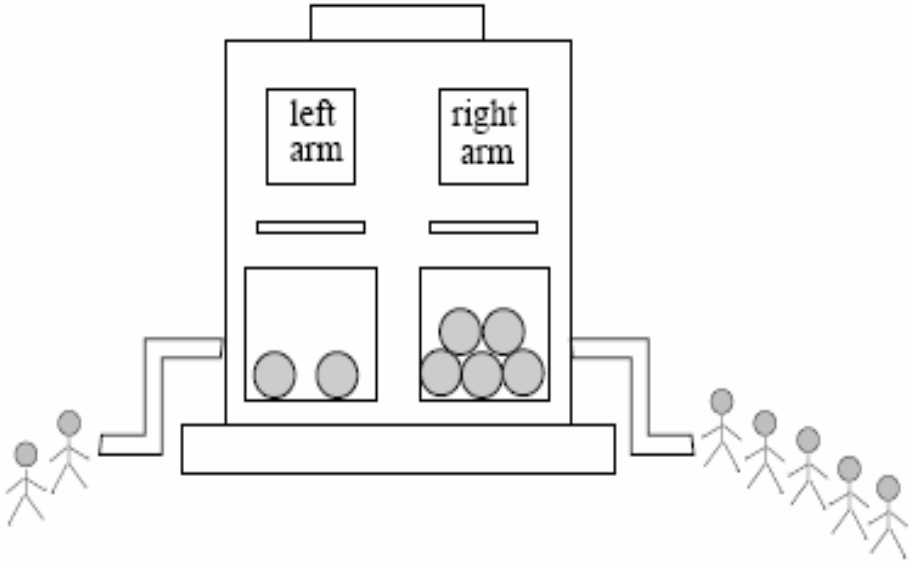
Multiple niches (human and animal) coexist in nature

- 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

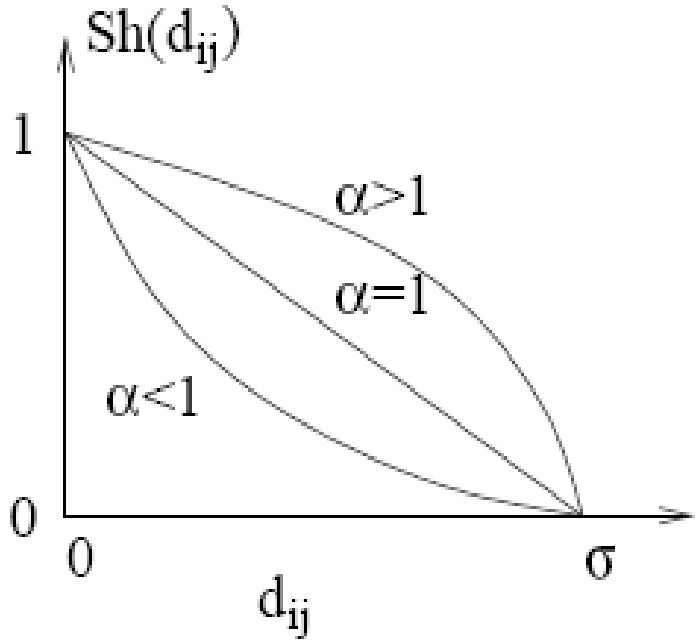


$$f_L/m_L = f_R/m_R$$

Sharing Function

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

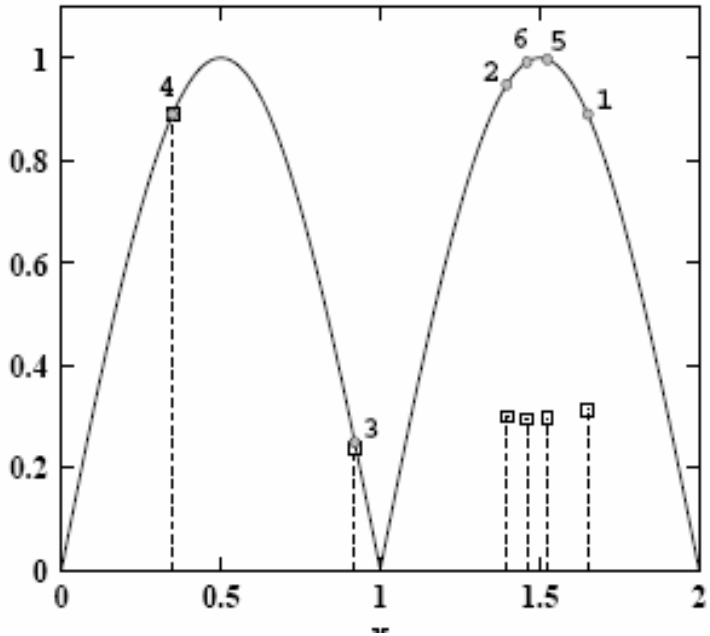
$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma}\right)^{\alpha} & \text{if } d_{ij} \le \alpha, \\ 0 & \text{otherwise.} \end{cases}$$



An Example: Phenotypic Sharing

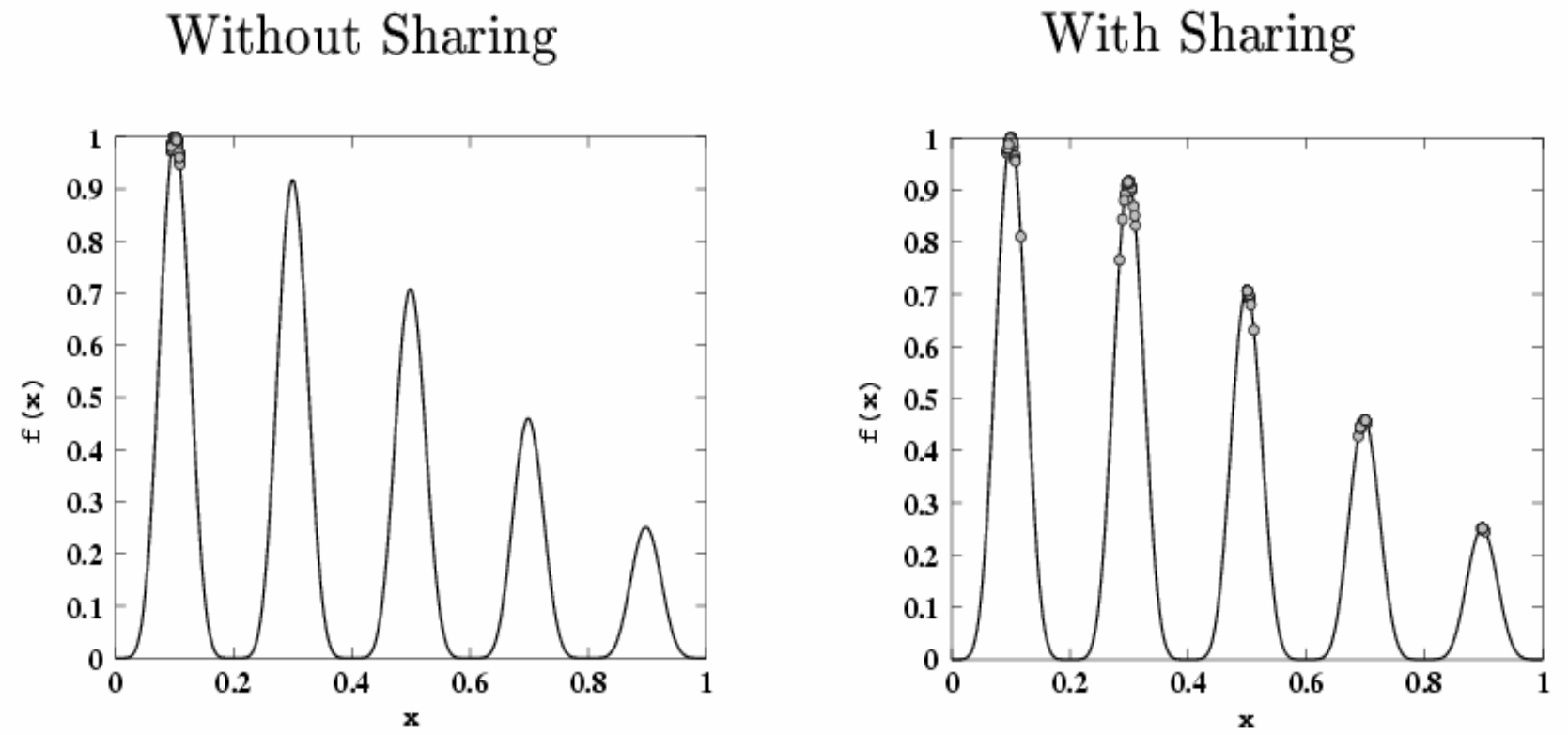
- Maximize $f(x) = |\sin(\pi x)|$, $x \in [0, 2]$
- $nc_i = \sum_{j=1}^{N} Sh(d_{ij})$

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



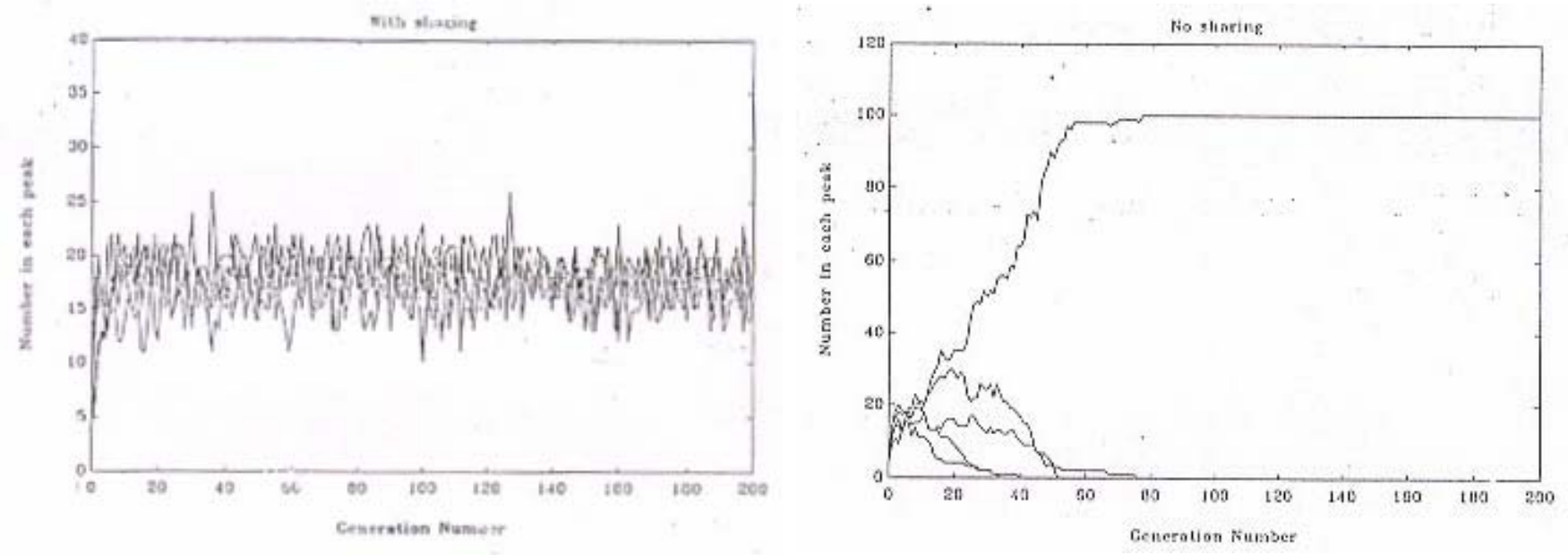
Simulation Results

- Inclusion of niche-formation strategy

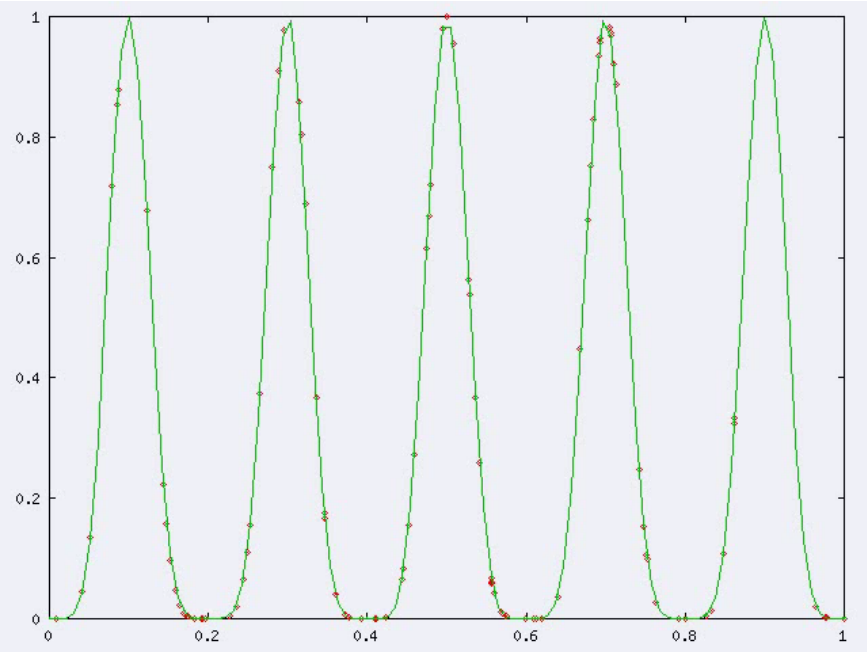


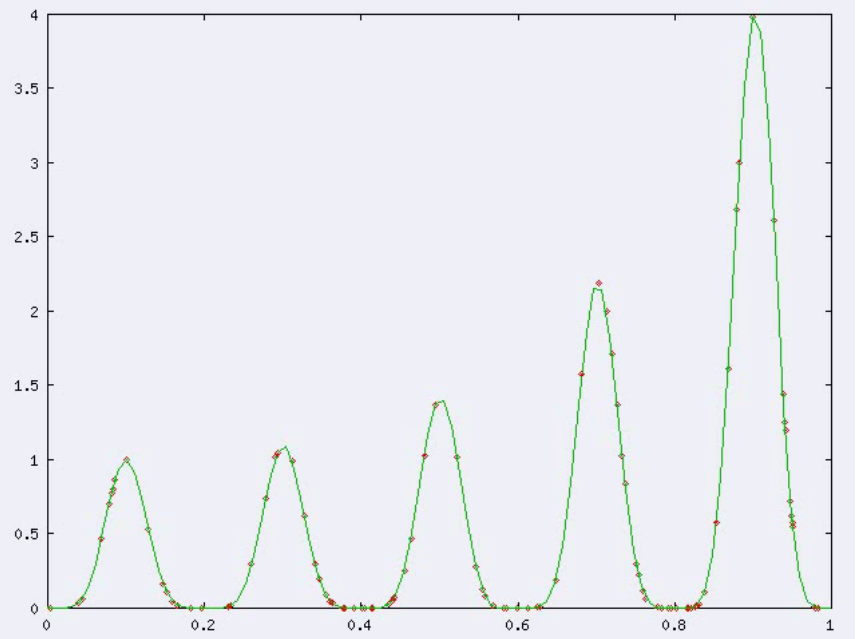
Distribution of Population Among Optima

- With and without sharing



Simulation on Two Problems

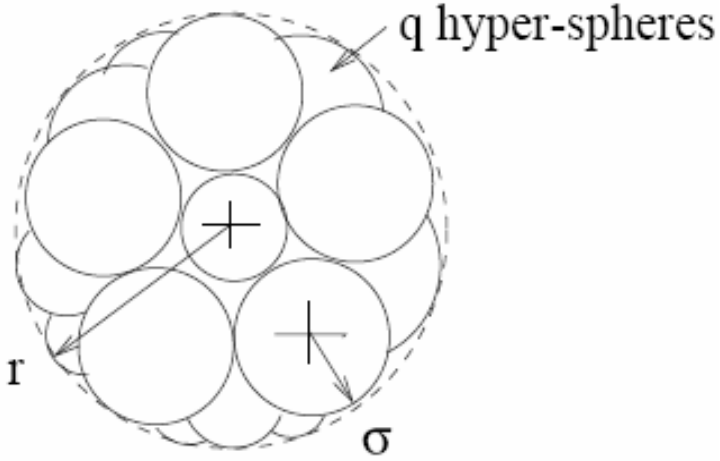




Phenotypic σ_{share}

- Assume number of optima: *q*
- 1-D space:

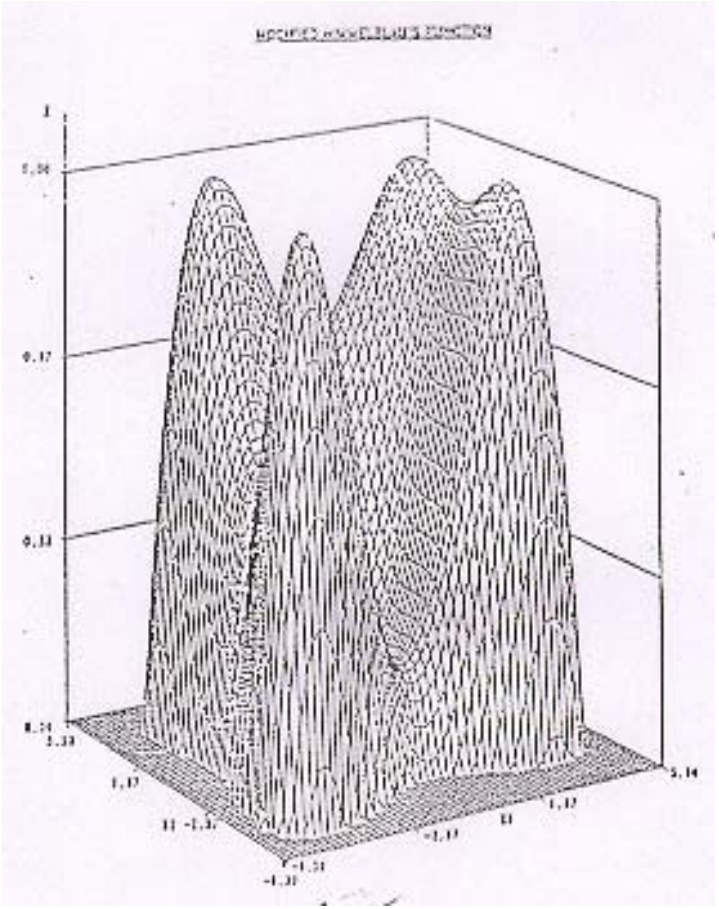
  $\sigma_{share} = (x_{max}-x_{min})/2q$
- *p*-D space:
  - r: Euclidean length of radius of entire *p*-D hyper-sphere
  - Volume: $V = cr^p$
  - Divide volume into *q* parts
  - σ_{share} is the radius of each part
  - $V = cr^p = q\, c\sigma_{share}^p$ yields



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

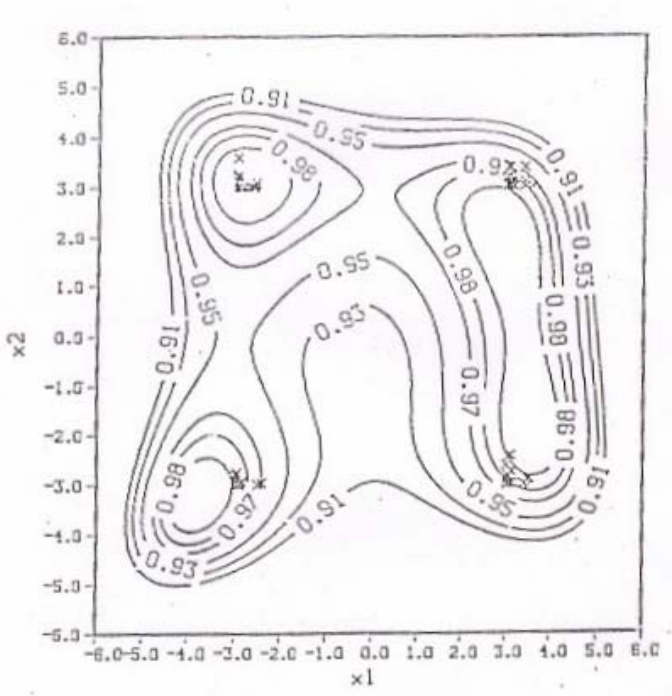
More Simulation Results

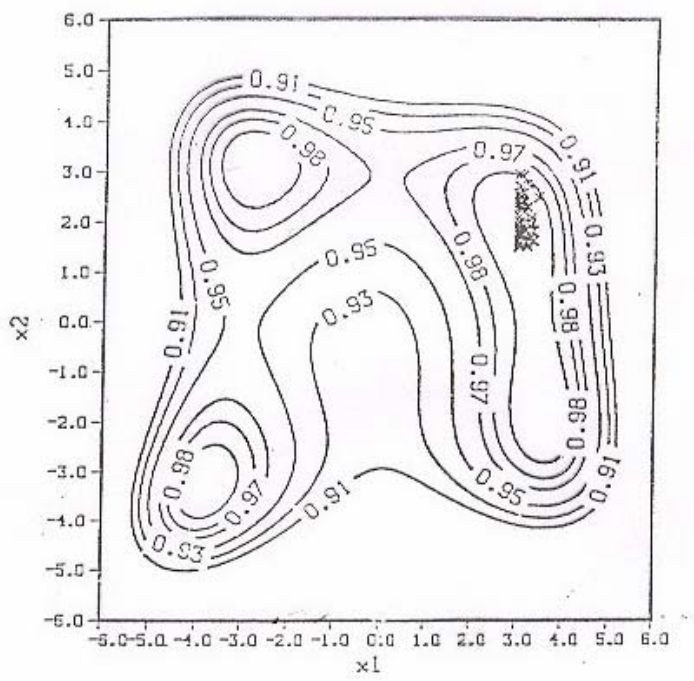
- A two-dimensional function
- Four optima
- Interested in finding all four optima



More Simulation Results (cont.)

- With and without sharing (phenotypic)



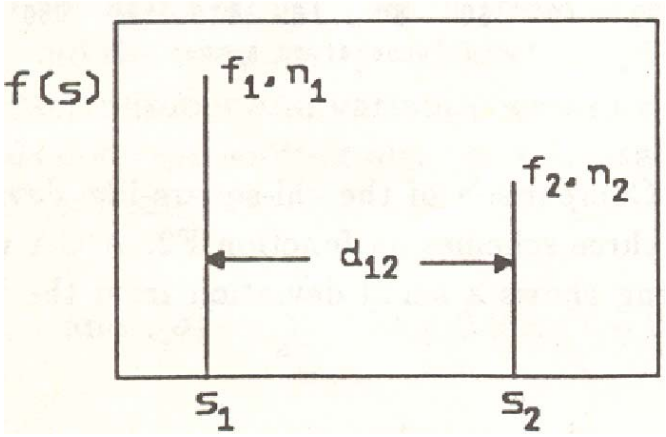


Genotopic Sharing

- d=Hamming distance (# of bit differences)
- q niches: $\frac{1}{2^l}\prod_{i=0}^{\sigma}\binom{l}{i}=\frac{1}{q}$
- Approximate: $\sigma=\frac{1}{2}\left(l+z^*\sqrt{l}\right)$
- z* found from cum. Normal distribution chart for 1/q

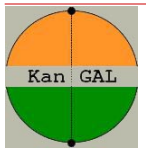
- Some difficulties
  - Need a minimum distance between optima ($\gamma=f_1/f_2$)

$$d\geq\left(1-\frac{1}{\gamma}\right)\sigma$$



Other Niching Approaches

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

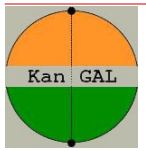


Clearing Strategy

- 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)

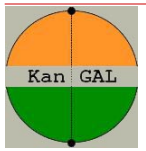
Clearing Strategy (cont.)

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



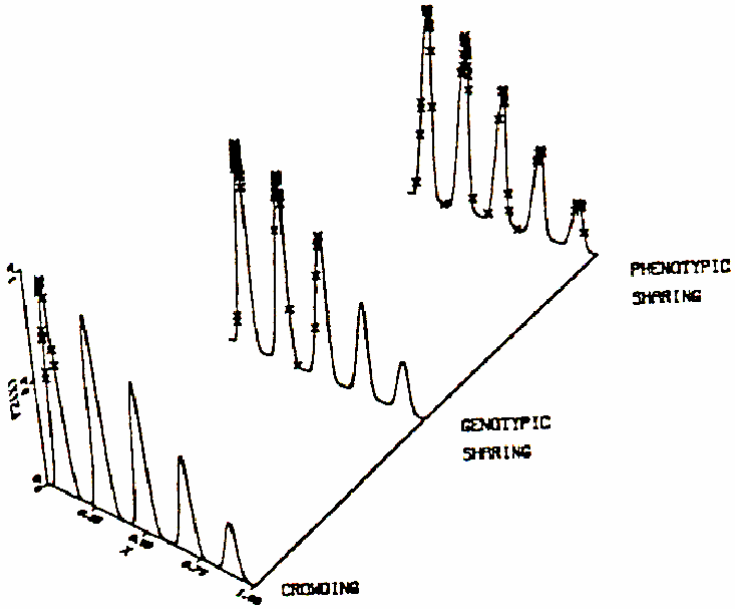
Clustering Approach

- 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)



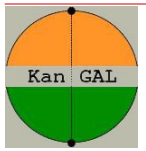
Crowding Approach

- 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)



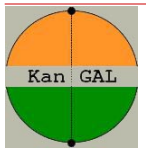
Preselection Procedure

- 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



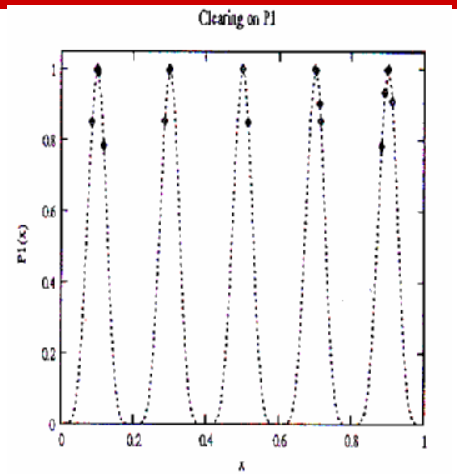
Continuously Updated Sharing Approach

- 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)

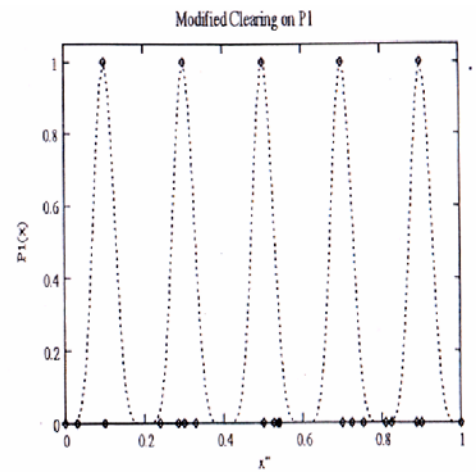


Comparison of Niching Methods

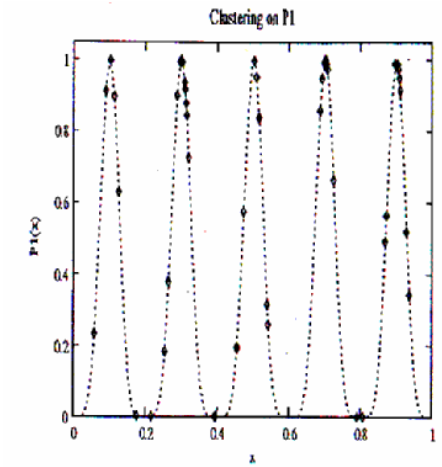
Clearing



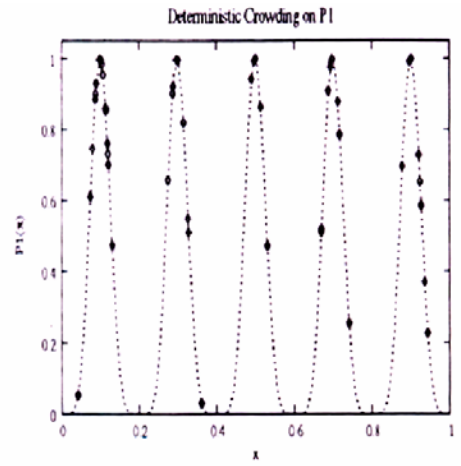
Modified Clearing



Clustering

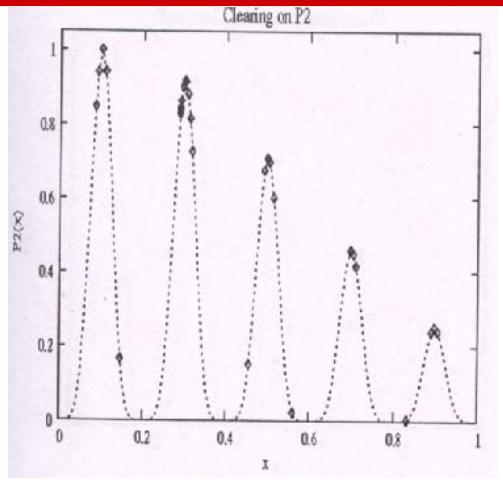


Crowding

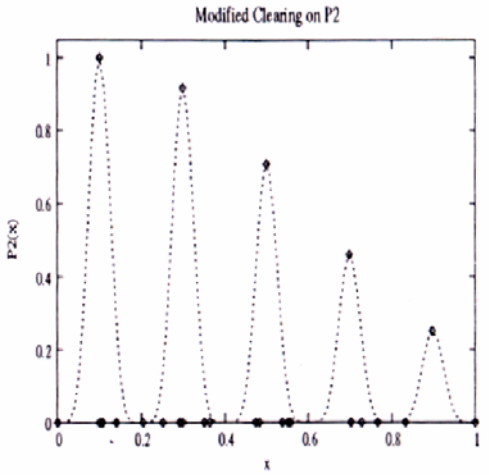


More Comparisons

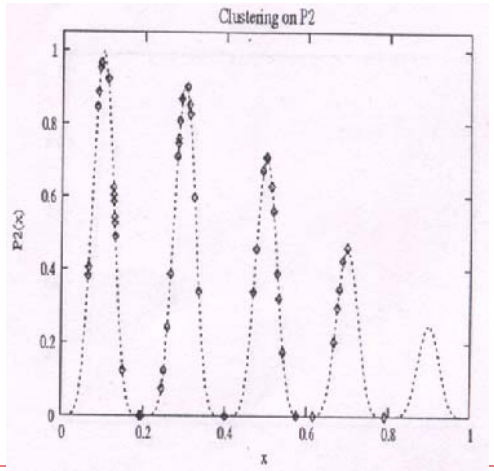
Clearing



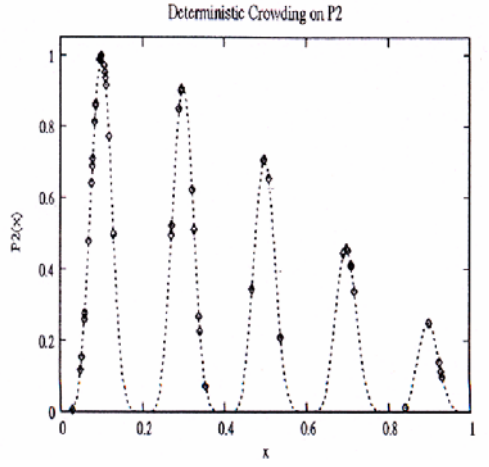
Modified Clearing



Clustering



Crowding



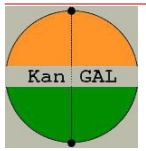
Comparison on a Scalable Problem

- 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

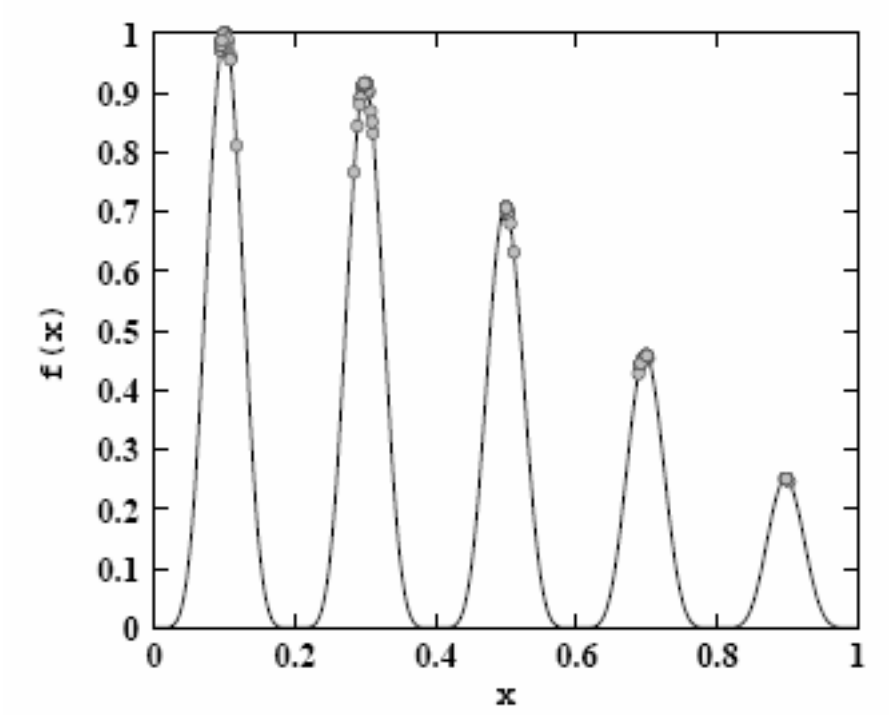
Mating Restriction: Speciation

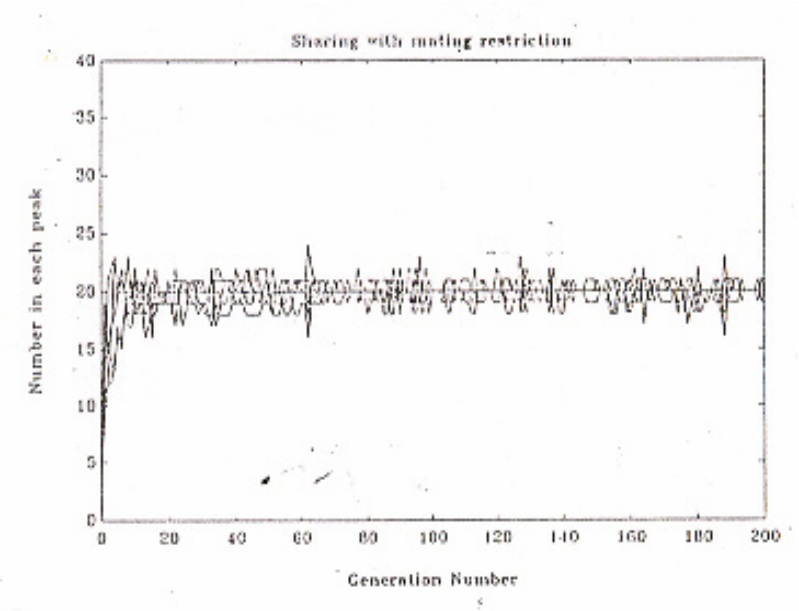
- Allow mating with intra-species individuals
- Solutions only within σ_{mating} mates to each other (restricted crossover)
  - Does not allow too many lethal solutions
- Causes stable subpopulations in many optima
- Other approaches:
  - Tag-template, taxon-exemplar etc.



Mating Restriction Results

A cleaner distribution





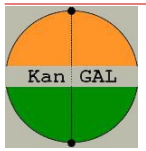
Tag-Template Scheme

- Tag-template scheme

  Tag Templ. Functional String

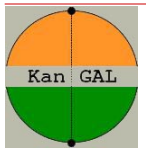
  0010 : **10: 010001101010110

- 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



Omni-Optimizer

- One algorithm for various problem-solving
  - Single-objective, single global solution
  - Single-objective, multiple global solns.
  - Multiple objectives, single-modal P-O solns.
  - Multiple objectives, multiple P-O solns.
- One algorithm degenerates to solve all above
- Should revolutionize act of optimization



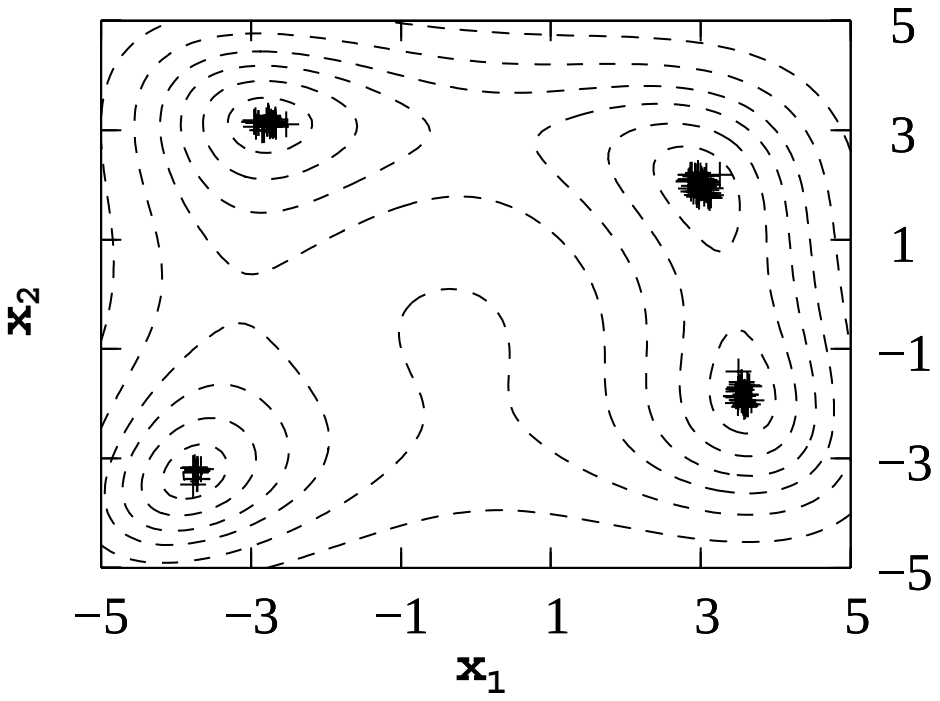
Motivation from Computation

- Adding a few integers
- N=1 is a degenerate case

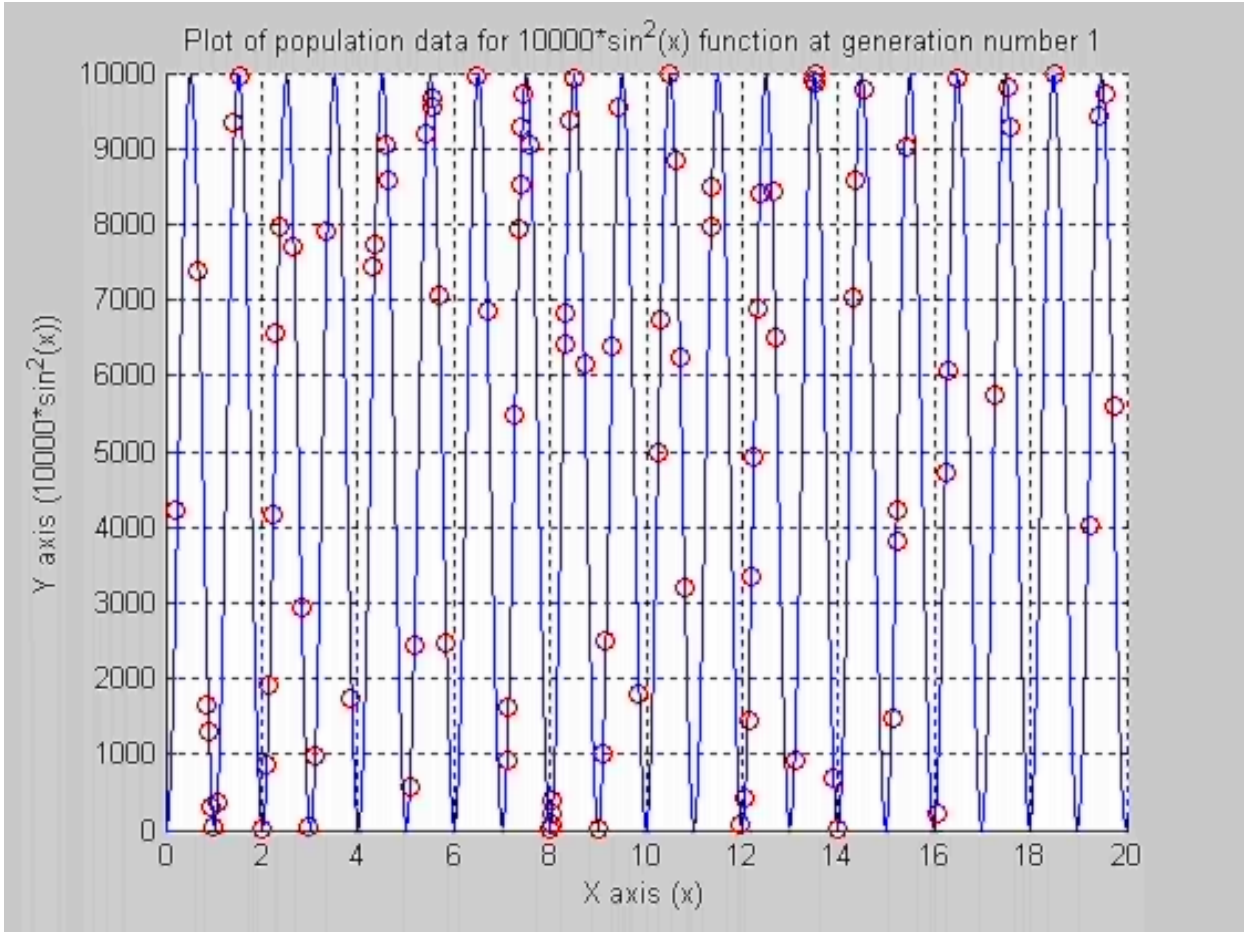
```
Add a few integers:
begin
   print 'enter number of integers to be added', read n
   sum = 0
   for i = 1 to n
      print 'enter integer i', read a[i]
      sum = sum + a[i]
   print 'Sum =' sum
end
```

Omni-Optimizer Developed and Tested

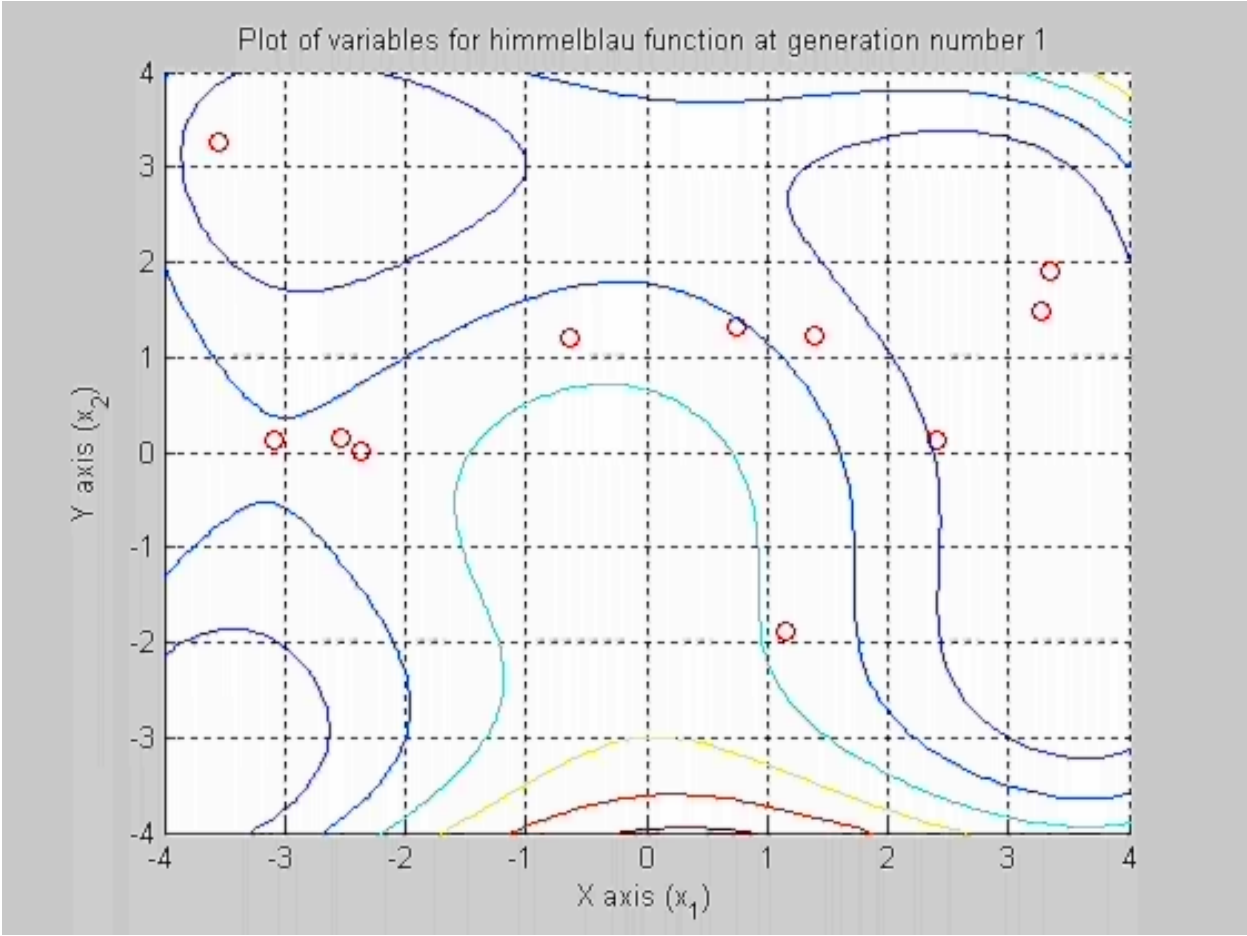
- Basic structure similar to NSGA-II
- Degenerates to an efficient GA (like CHC) for single-objective problems
  - Domination reduces to '<'
  - Elite preservation present
- Multiple global solutions are found



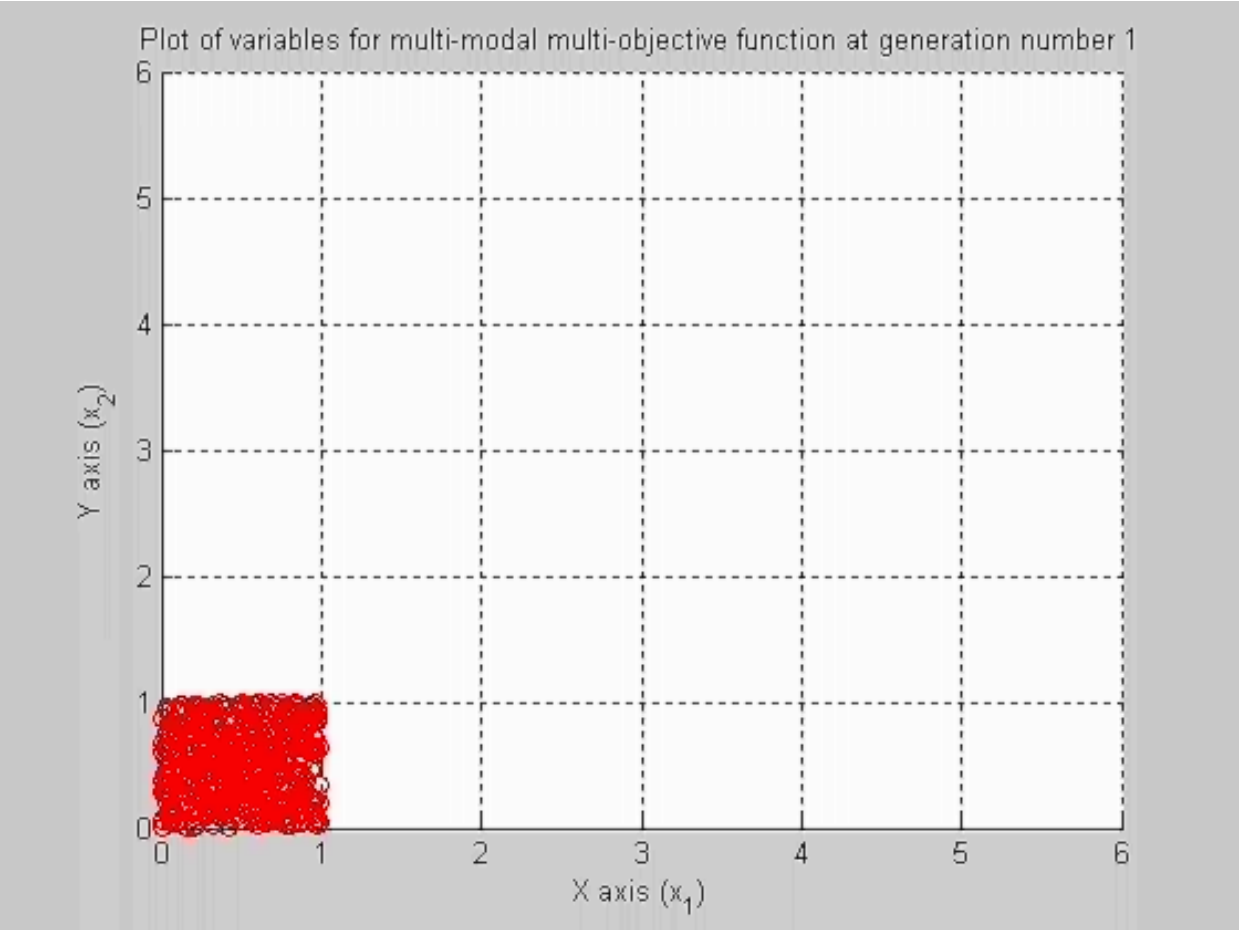
$10^4 Sin^2(x)$


Plot of population data for 10000*sin2(x) function at generation number 1
Y axis (10000*sin2(x))
X axis (x)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
2
4
6
8
10
12
14
16
18
20

Himmelblau's Function



Multi-Modal Multi-Objective Problem

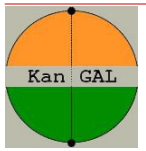


Function Approximation

- Evaluation of most real-world problems is computationally expensive
- Optimization algorithm run into days
- To save time, use approximate models of objective function and constraints
- Different techniques
  - A fixed model
  - Updating the model with iteration

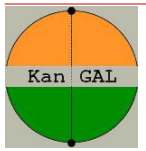
Metamodel

- Model of models (Kleijnen, 1987)
- Approximations in models
  - Error of approximating functions
  - Measurement (random) errors
- Computer simulation produce determinisitic results
  - Random effect is missing

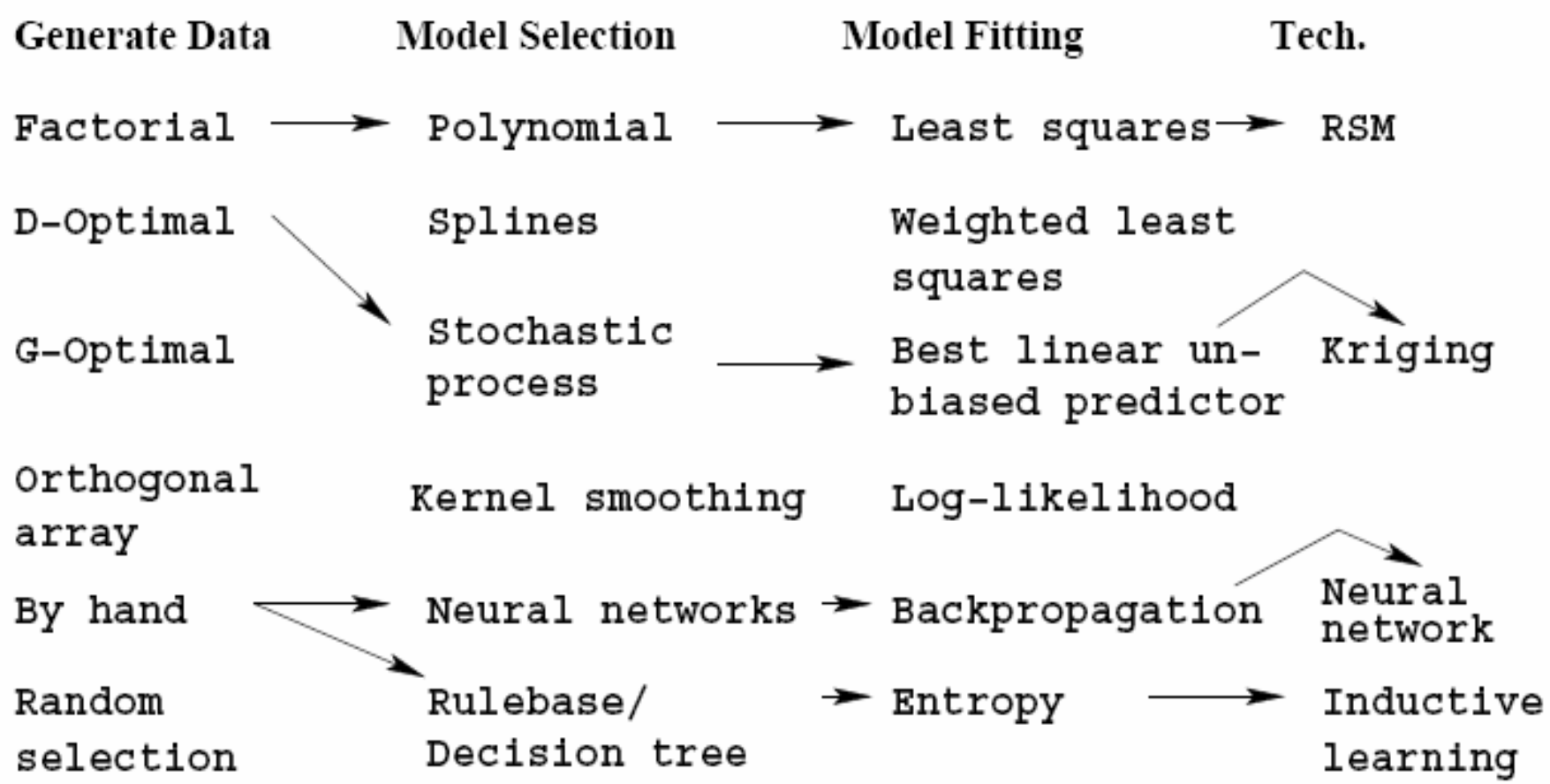


Metamodeling Techniques

- 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



Metamodeling Techniques (cont.)



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

Method of Least Squares

- Responses $\mathbf{y}$ can be created from a design $\mathbf{x} \in \mathbb{R}^n$
- Say the model: $\mathbf{y} = \sum_{i=1}^{k} \phi_i(\mathbf{x})\beta_i$
- Say an experiment at design $\hat{\mathbf{x}}$ gives $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}}_e = \boldsymbol{\phi}^T \boldsymbol{\beta} + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- An example: $\boldsymbol{\phi} = [1 \quad x_1 \quad x_1 x_2 \quad x_1^2]$:

$$\begin{pmatrix} y_{e1} \\ \vdots \\ y_{en} \end{pmatrix} = \begin{pmatrix} 1 & x_{1-e1} & x_{1-e1}x_{2-e1} & x_{1-e1}^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1-eN} & x_{1-eN}x_{2-eN} & x_{1-eN}^2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{pmatrix}$$

Method of Least Squares (cont.)

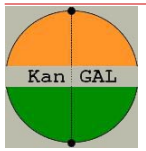
- Find $\mathbf{b}$ which minimizes error: $\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}$
- Performing $\frac{d(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon})}{d\mathbf{b}}$ yields $\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}^T\mathbf{y}_e$
- Also $E(\mathbf{b}) = \boldsymbol{\epsilon}$
- And $Var(\mathbf{b}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$, where $Var(\mathbf{y}) = \mathbf{I}\sigma^2$
- σ is fixed for an experimental technique, $(\mathbf{X}^T\mathbf{X})^{-1}$ depends on location of $\mathbf{x}_i$
- Predicting capability can be enhanced by suitable choice of $\mathbf{x}_i$ using a DOE

Aim in Choosing Points

- $(X^TX)^{-1}$ small
- $(X^TX)^{-1}$ diagonal to make uncorrelated variance
- $(X^TX)^{-1}$ real, symmetric matrix -> Eigenvalues v_i
- A-Optimality: Min $tr((X^TX)^{-1}) = \Sigma v_i$
- D-Optimality: Min $|(X^TX)^{-1}| = \Pi v_i$
- E-Optimality: Min Max v_i
- G-Optimaility: Min $\varphi^T (X^TX)^{-1} \varphi$

RSM Tips

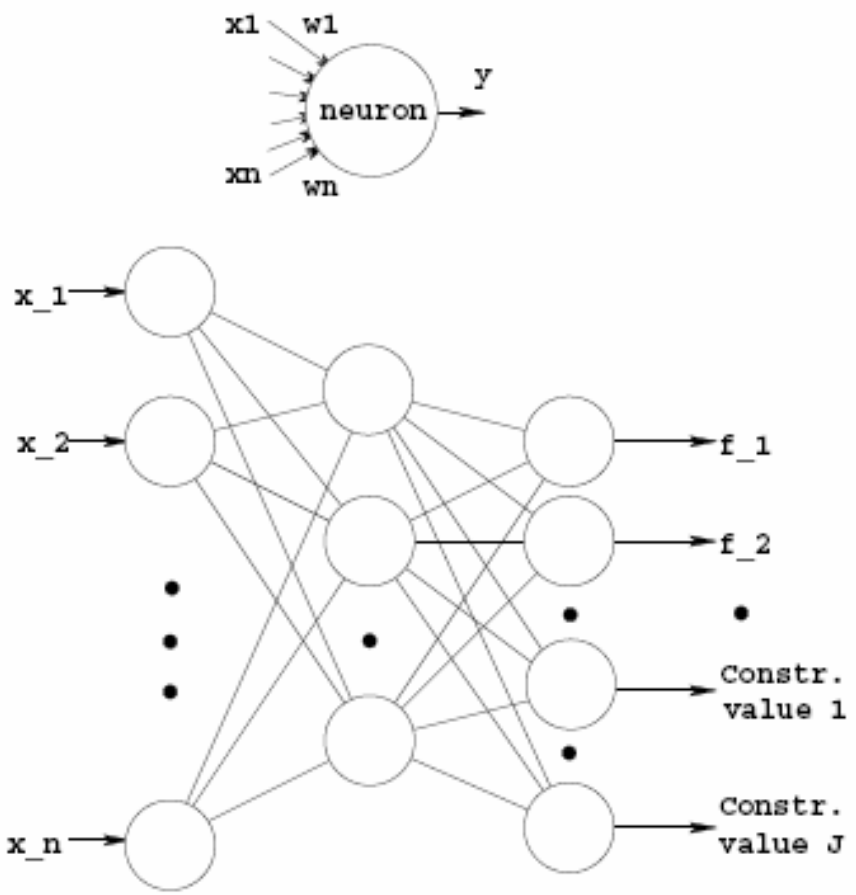
- Popular and most widely used
- Best suited in applications with random error
- However, limited handling on non-linearity
- Usually applied for k<10
- Sequencial RSM with move limits and trust region approach



Artificial Neural Networks (ANN)

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

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



ANN (cont.)

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

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

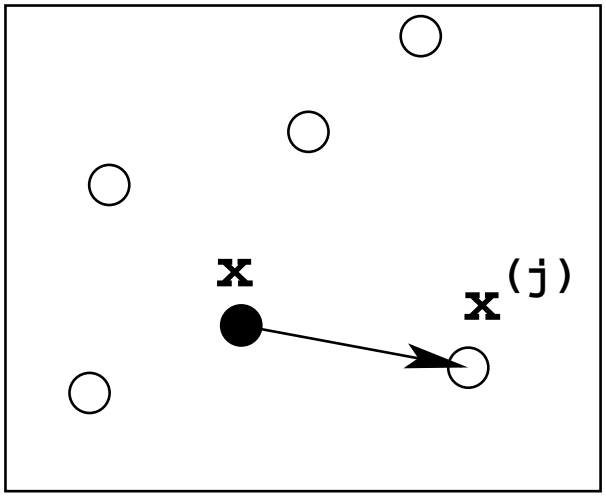
- A non-linear regression method
- Suitable for large (104) parameters
- Best suited for deterministic problems (no random error)
- Computationally expensive procedure

Kriging Procedure

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

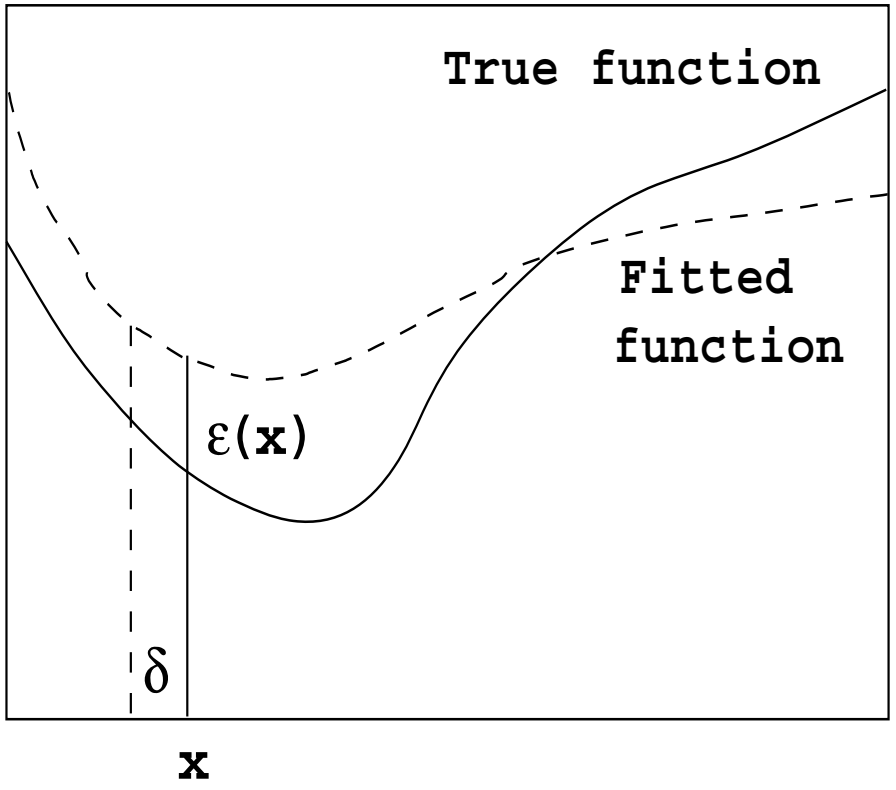
$$f(\mathbf{x}) = \sum_{i=1}^{M} \lambda_i y(\mathbf{x}^{(j)})$$

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



Fundamentals of Kriging

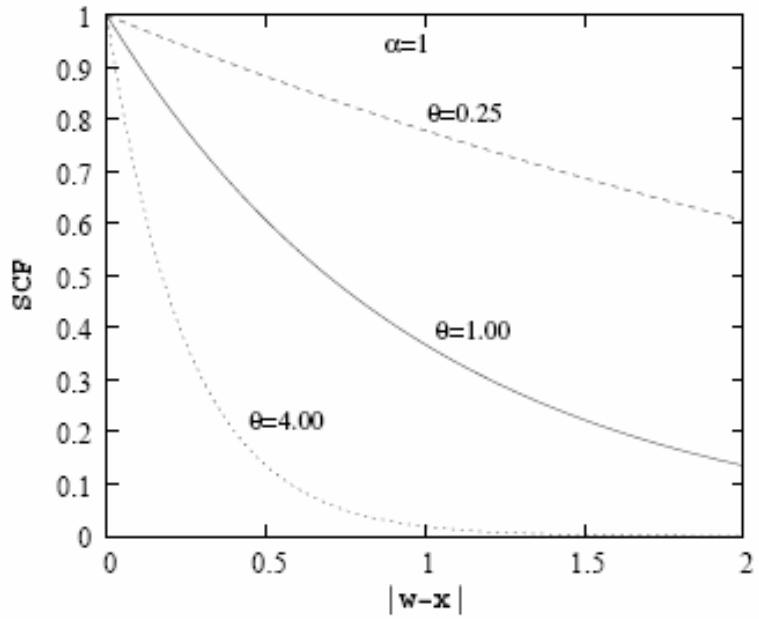
- F(x) is true function
- f(x) is fitted function
- ε(x) = F(x)-f(x)
- If error at x is large, it is reasonable to believe that
  - Error at x+δ is also large



Fundamentals of Kriging (cont.)

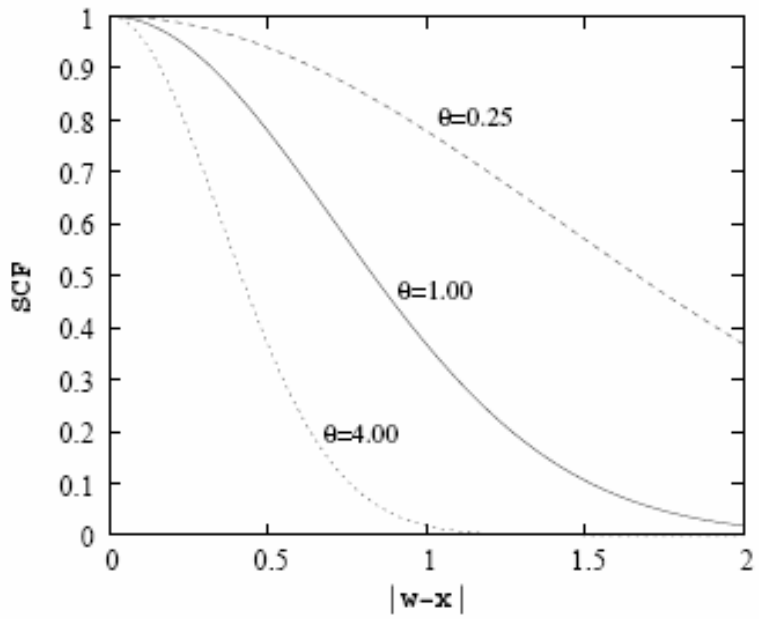
- Say, M data points: $(\mathbf{x}^{(j)}, y_j), \quad j = 1, 2, \ldots, M$
- Assume model: $Y(\mathbf{x}) = \sum_{k=1}^{K} \beta_k \mathbf{f}_k(\mathbf{x}) + Z(\mathbf{x})$
- $\mathbf{f}_k(\mathbf{x})$ assumed global model, can be taken as constant, polynomial or others
- $Z(\mathbf{x})$ is a local variation and stochastic:
  - $E(Z(\mathbf{x})) = 0$
  - $Cov(Z(\mathbf{w}), Z(\mathbf{x})) = \sigma_Z^2 R(\mathbf{w}, \mathbf{x})$
  - σ_Z^2 is the process variance
  - $R(\mathbf{w}, \mathbf{x})$ is spatial correlation function (SCF)
  - $R(\mathbf{w}, \mathbf{x}) = \Pi_{j=1}^{n} R_j(\mathbf{w}_j, \mathbf{x}_j)$
  - SCF (R_j) is function of distance between $\mathbf{w}_i$ and $\mathbf{x}_i$

Commonly Used SCFs



Exponential:

$$\exp\left(-\theta_j \mid \mathbf{w}_j - \mathbf{x}_j \mid^{\alpha}\right) \quad 0 < \alpha \leq 2$$



Gaussian:

$$\exp\left(-\theta_j \mid \mathbf{w}_j - \mathbf{x}_j \mid^{2}\right)$$

Kriging Procedure

Step 1: Normalize $\mathbf{x}^{(j)}$ and $y^{(j)}$ and set $\mathbf{Y} = [y^{(j)}]$

Step 2: Maximize log-likelihood function:

$$\max_{\boldsymbol{\theta}} \quad -\frac{1}{2}\left[M \log \sigma_Z^2 + \log R(\boldsymbol{\theta})\right]$$

and obtain $\boldsymbol{\theta}^*$

- $R(\boldsymbol{\theta})$ is a matrix: $\Pi_{j=1}^{n} R_j(\mathbf{w}_j, \mathbf{x}_j;\ \theta_j)$
- MLE of σ_Z^2: $\sigma_Z^2 = \frac{1}{M}\left(\mathbf{Y} - \mathbf{I}\hat{\beta}(\boldsymbol{\theta})\right)^T R(\boldsymbol{\theta})^{-1}\left(\mathbf{Y} - \mathbf{I}\hat{\beta}(\boldsymbol{\theta})\right)$
- MLE of $\hat{\beta}$: $\hat{\beta}(\boldsymbol{\theta}) = (\mathbf{I}^T R(\boldsymbol{\theta})^{-1}\mathbf{I})^{-1}\mathbf{I}^T R(\boldsymbol{\theta})^{-1}\mathbf{Y}$

Step 3: Compute $\hat{\beta}^* = \hat{\beta}(\boldsymbol{\theta}^*)$, $R(\boldsymbol{\theta}^*)$, and ${\sigma_Z^2}^*$

Kriging Procedure (cont.)

Step 4: Compute fitted function at any point $\mathbf{x}$:

$$\hat{y}(\mathbf{x}) = \hat{\beta}^* + r^T(\mathbf{x}) R(\boldsymbol{\theta}^*)^{-1} \left(\mathbf{Y} - \mathbf{I}\hat{\beta}^* \right)$$

where $r(\mathbf{x}) = [R(\mathbf{x}, \mathbf{x}^{(i)})] \quad i = 1, 2, \ldots, M$

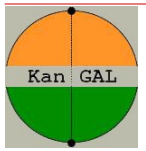
Step 5: The corresponding MSE (error):

$$MSE(\mathbf{x}) = {\sigma_Z^2}^* \left[1 - \begin{pmatrix} 1 & r^T(\mathbf{x}) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{I}^T \\ \mathbf{I} & R(\boldsymbol{\theta}^*) \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ r(\mathbf{x}) \end{pmatrix} \right]$$

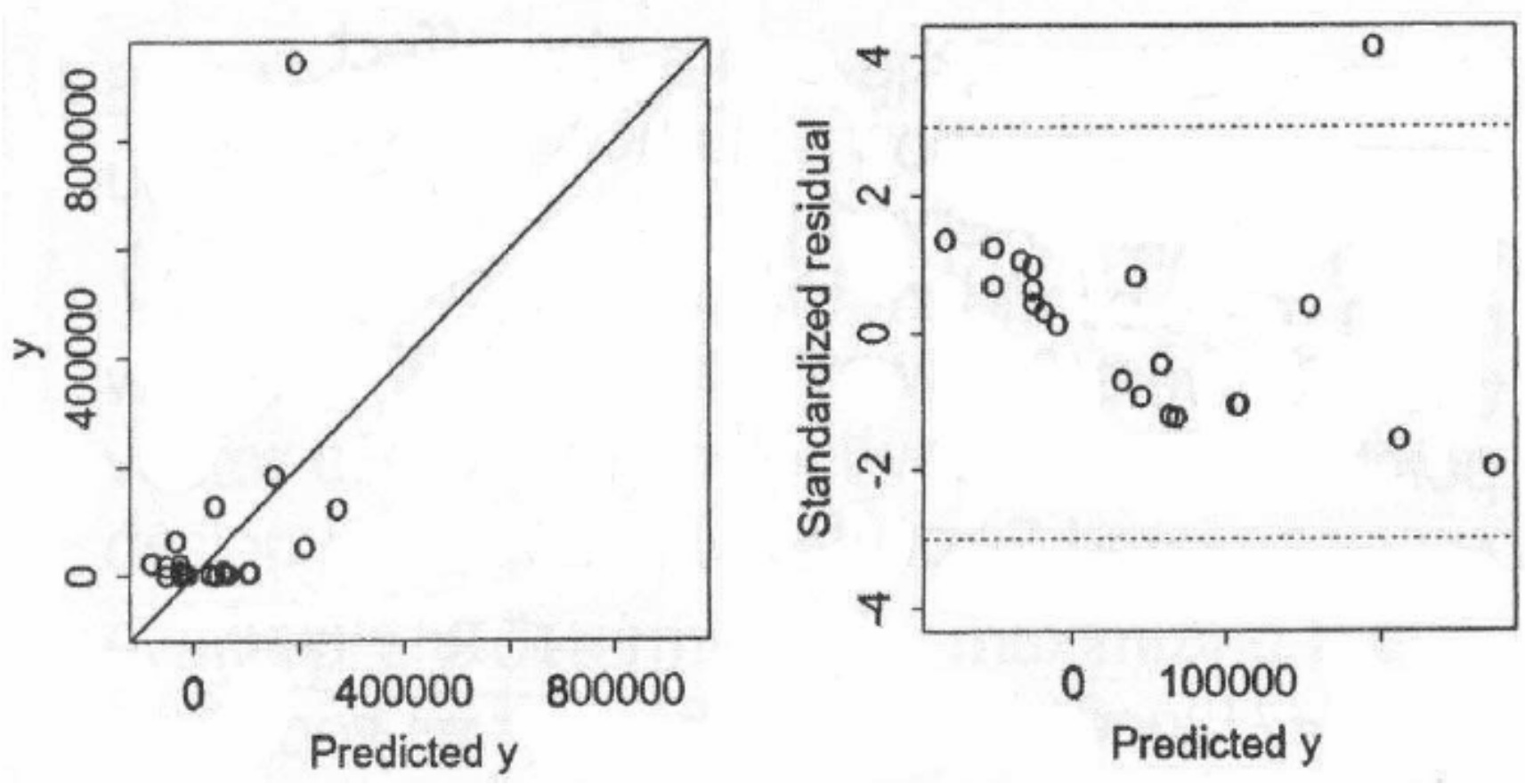
- If $\mathbf{x}$ is closer to an observed point, the MLE will be small

Model Verification

- Leave-one-out-cross-validation scheme
- Leave out one observation (say $x^{(J)}$ and $y^{(J)}$)
- Find the kriging model with (M-1) points and find $\hat{y}^{(J)}$
- Compute the error $s^J = \sqrt{(MSE(x^{(J)}))}$
- If the normalized error $(y^{(J)} - \hat{y}^{(J)})/s^J$ is within [-3,3], the model is acceptable

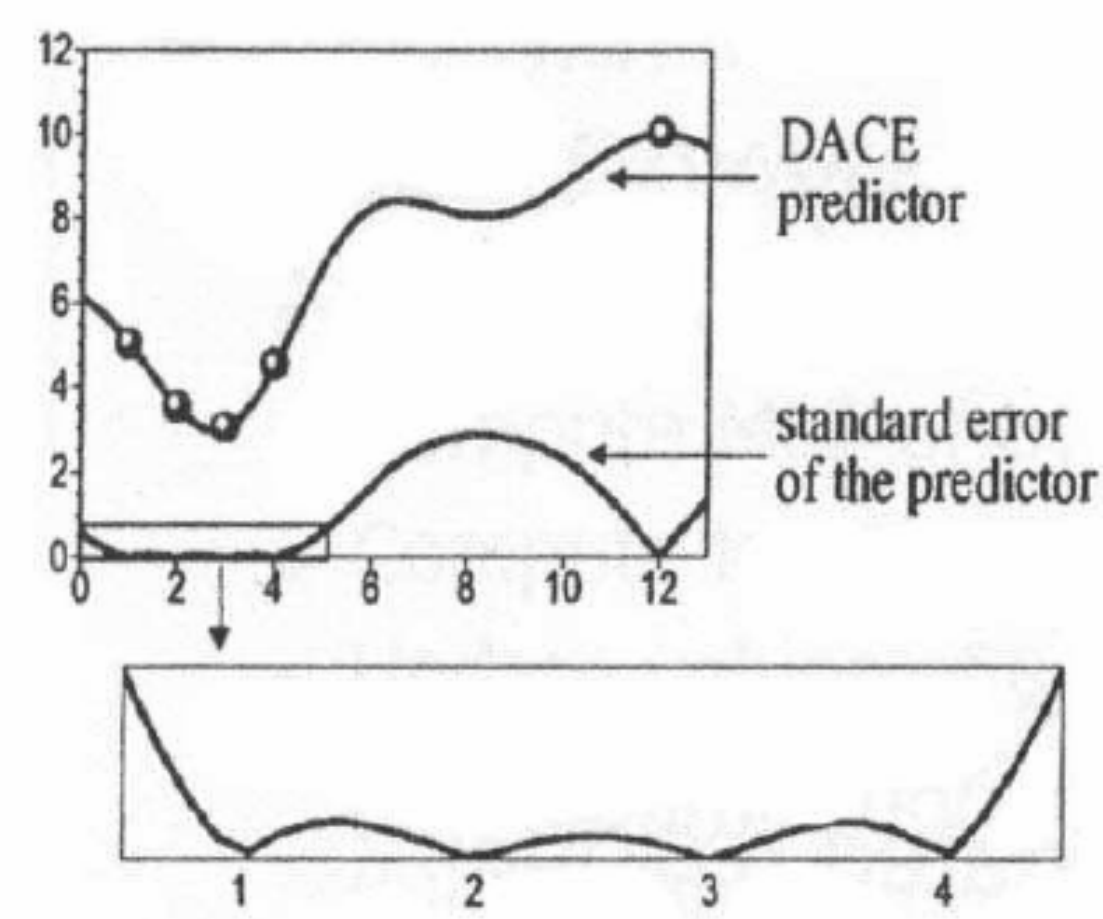


Model Verification: An Example



Properties of a Kriging Model

- Near more samples, error is low
- Take more samples where error is high

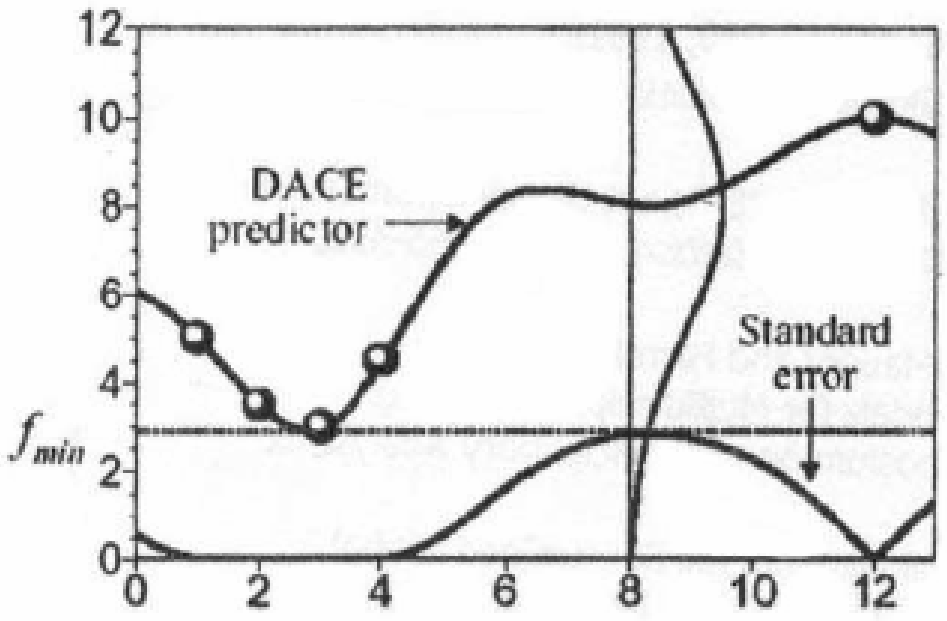


Kriging and Optimization

- Estimate error distribution at any x: $N(\hat{y}(\mathbf{x}), \sqrt{MSE(\mathbf{x})})$
- Compute the prob. of improvement:

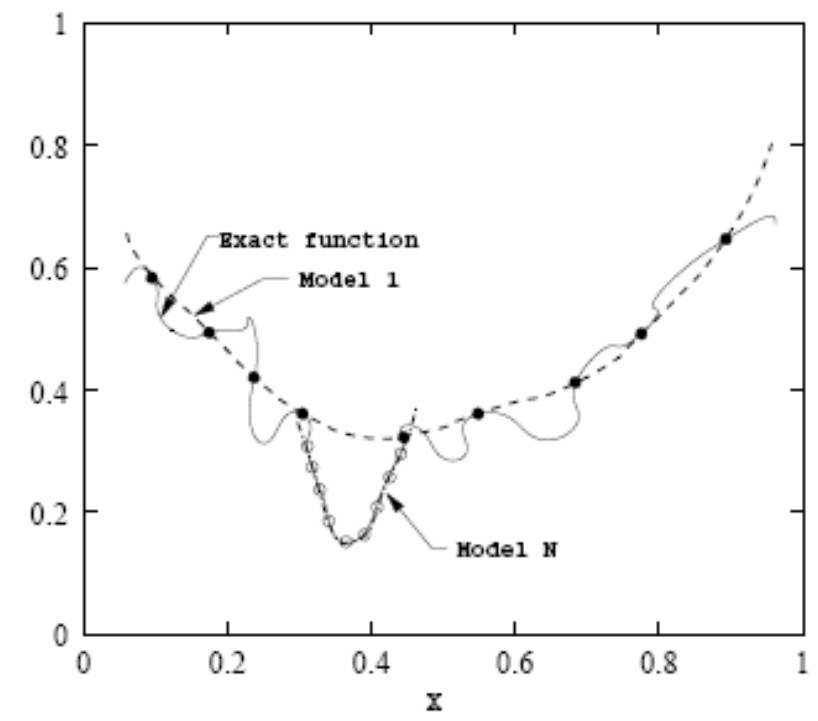
$$\int_{-\infty}^{f_{\min}} N(\hat{y}, \sqrt{MSE}) d\hat{y}$$

- If prob. is greater than a threshold, get x evaluated
- More in Lecture L6



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

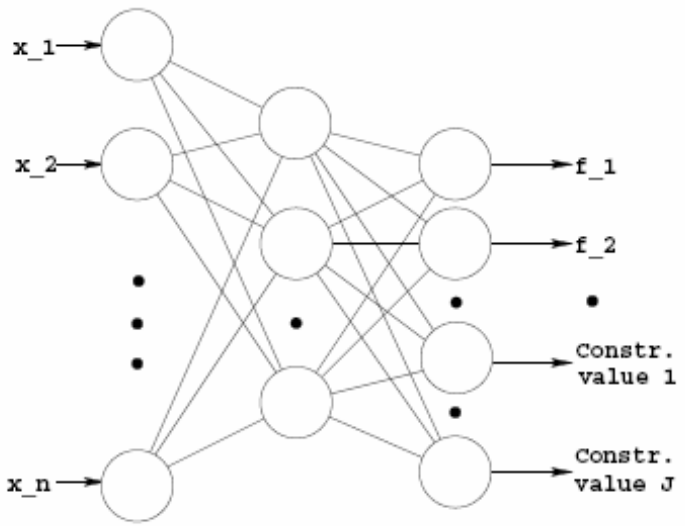


Approximation Models

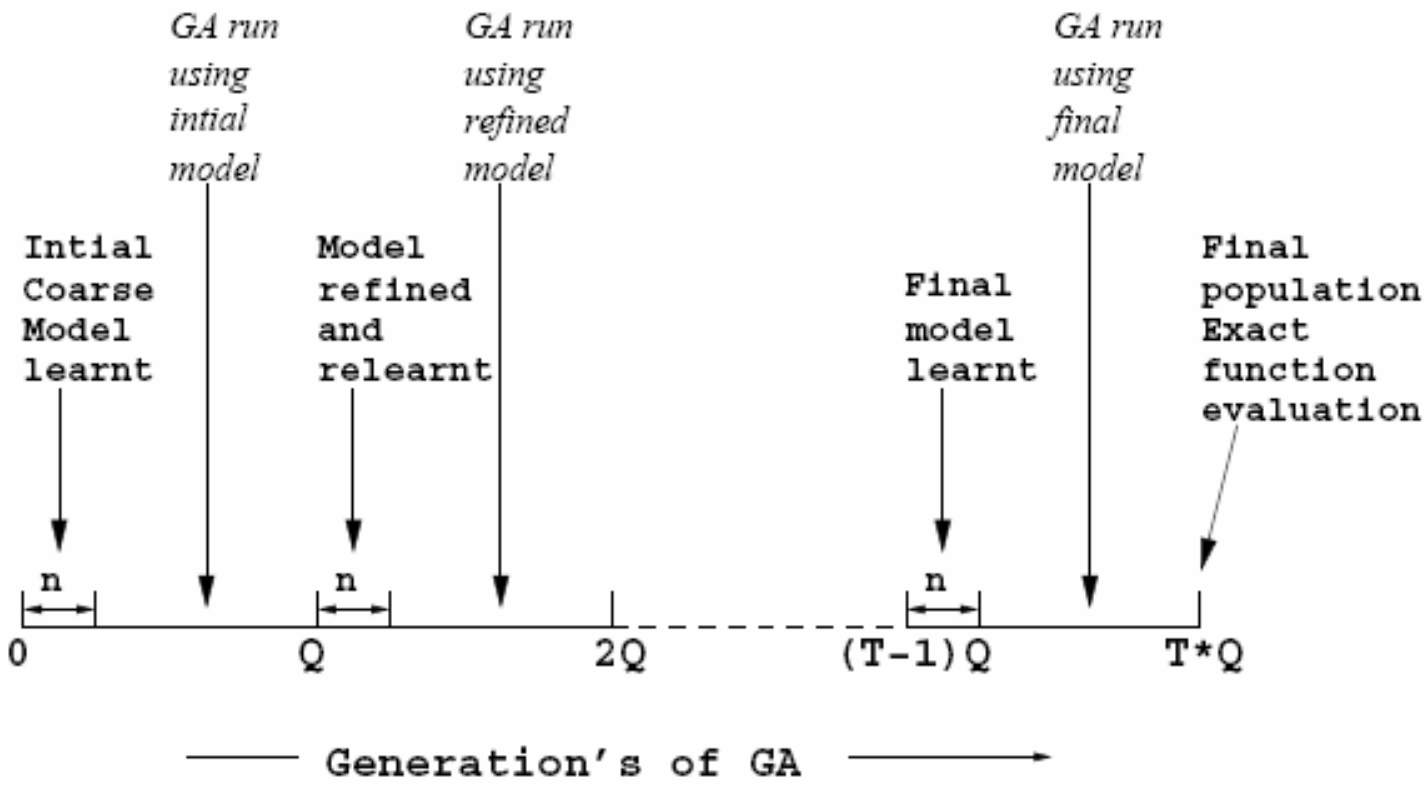
- Through linear or quadratic apprximation of functions
- Through a set of chosen basis functions
- These two are problem-specific
- Through regression using a set of points
  - An artificial neural network can be used
  - Advantage: Multiple functions (objective and constraints) can be modeled concurrently
  - A generic procedure

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



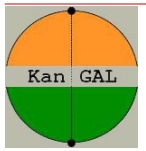
Generation-wise Sketch of Proposed Approach



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

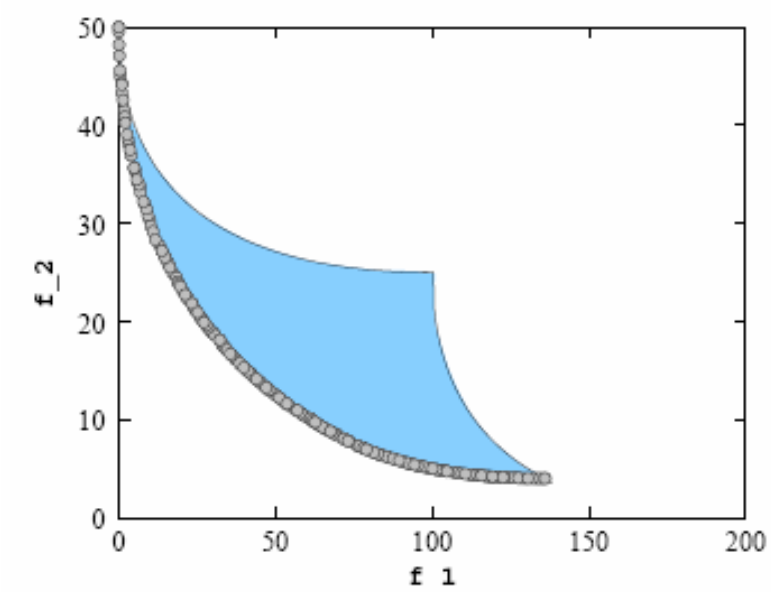
Flip Side: Parameters Involved

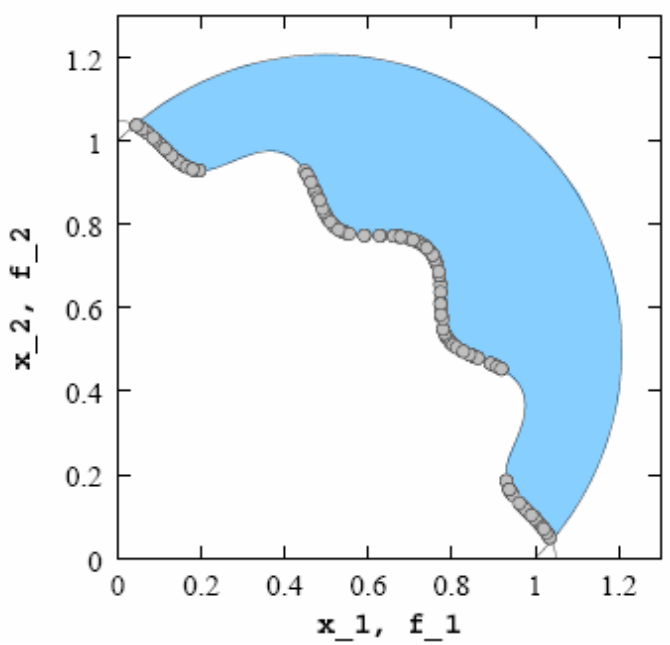
- Frequency of ANN modeling
- Number of training points (can reduce with iteration)
- Learning rate in training
- Final RMS error in ANN training
- Learning models (incremental or batch)
- ANN architecture (connectivity and hidden layers



Constraint Optimization Problems

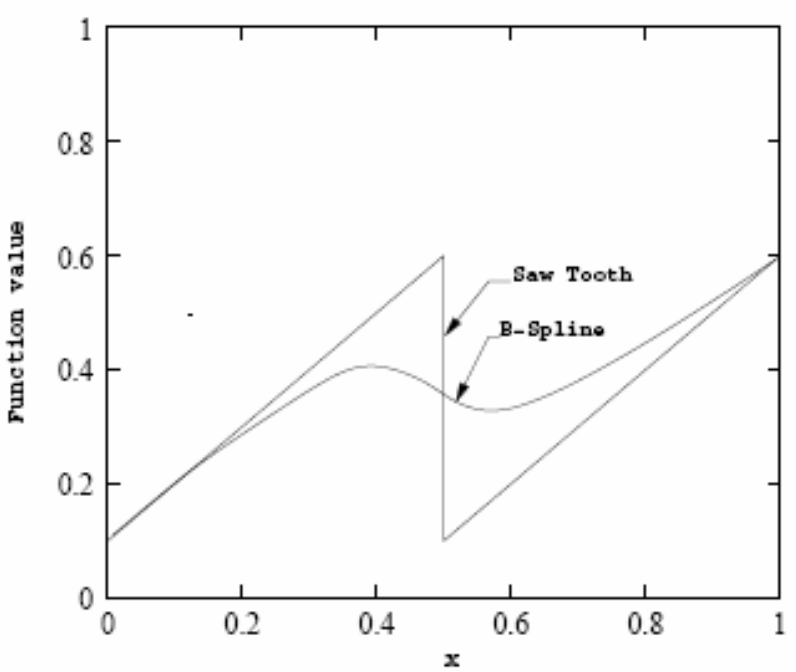
- BNH and TNK test problems having two constraints each
- B-10-2 model with 3-9-4 ANN architecture





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)



Formulation of the Problem

- A B-spline curve S having 41 control points in [0,1]
- Numerical computation of the integral with 400 intermediate points:

$$f_1(\mathcal{S}) = \int_{x=0}^{x=1} \mid f_{\text{saw-tooth}} - \mathcal{S} \mid dx$$

- Differentials are computed exactly

$$f_2(\mathcal{S}) = \max_{x=0}^{x=1} \frac{\frac{d^2\mathcal{S}}{dx^2}}{\left[1 + \left(\frac{d\mathcal{S}}{dx}\right)^2\right]^{3/2}}$$

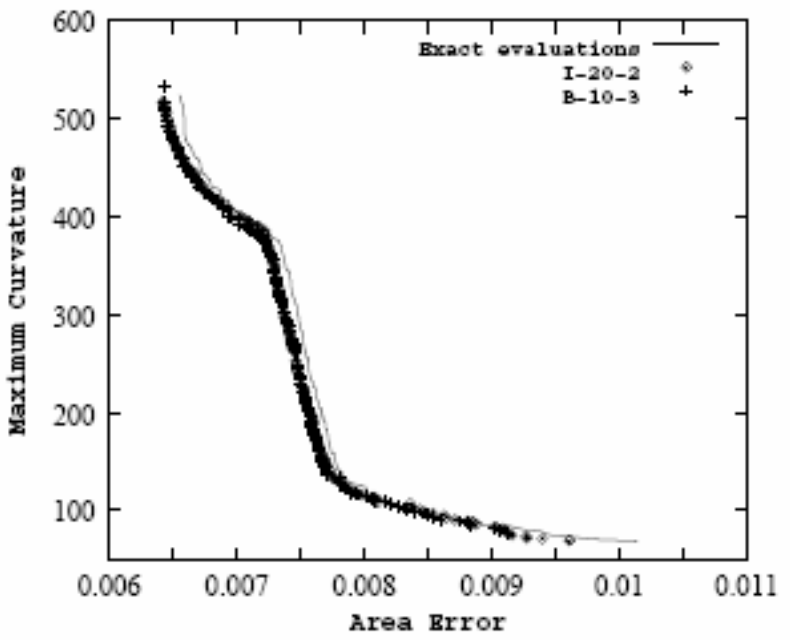
Parameter Values Adopted

40 input neurons	2 output neurons	One hidden layer
Moment. factor 0.1	Sigmoidal logistic activation function	
RMS error 0.005	Input data [0.1,0.9]	Output data [0,1]
Training cases $200n$ to $50n$ reduced linearly		
Population size 200	SBX crossover ($\eta_c = 10$)	
Polynomial mutation ($\eta_m = 50$)		

- n, Q learning rate and number of hidden neurons varied

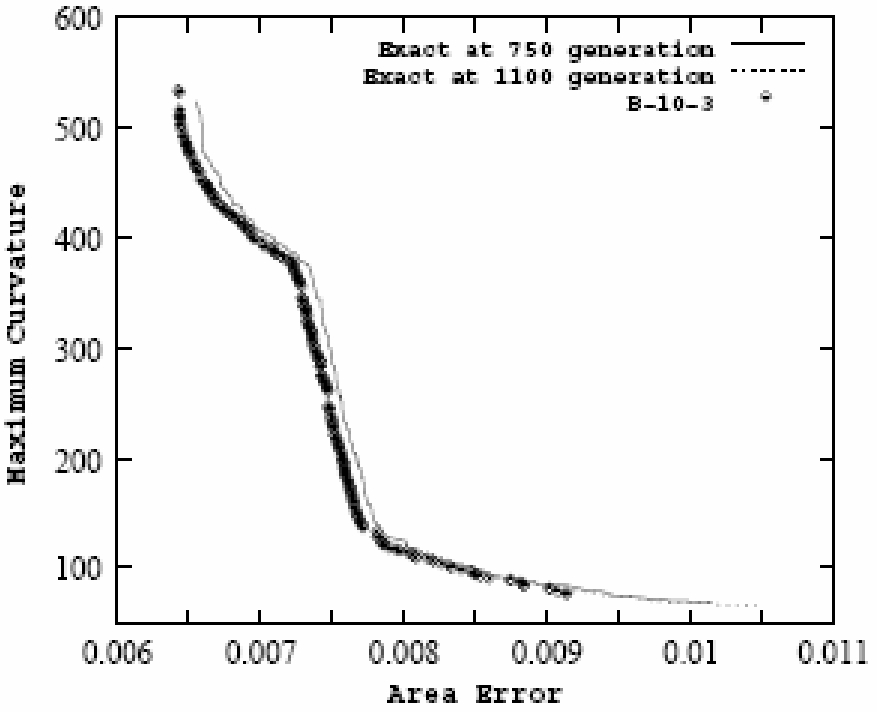
Optimal Parameter Values and Results

- ANN with Incremental learning after every 20 generations with 2N data points (I-20-2)
- ANN with Batch learning after every 10 generations with 3N data points (B-10-3)
- Compared with NSGA-II alone (750 generations, 200 members

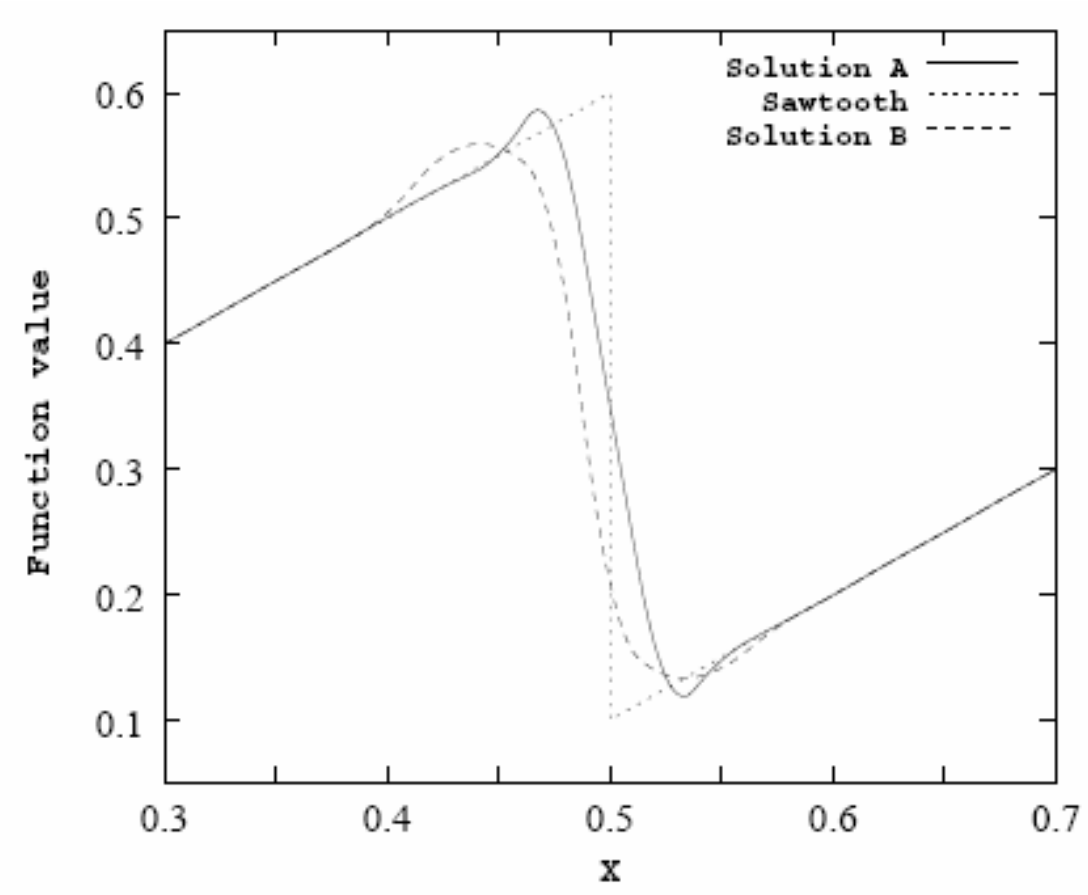


Savings in Function Evaluations

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

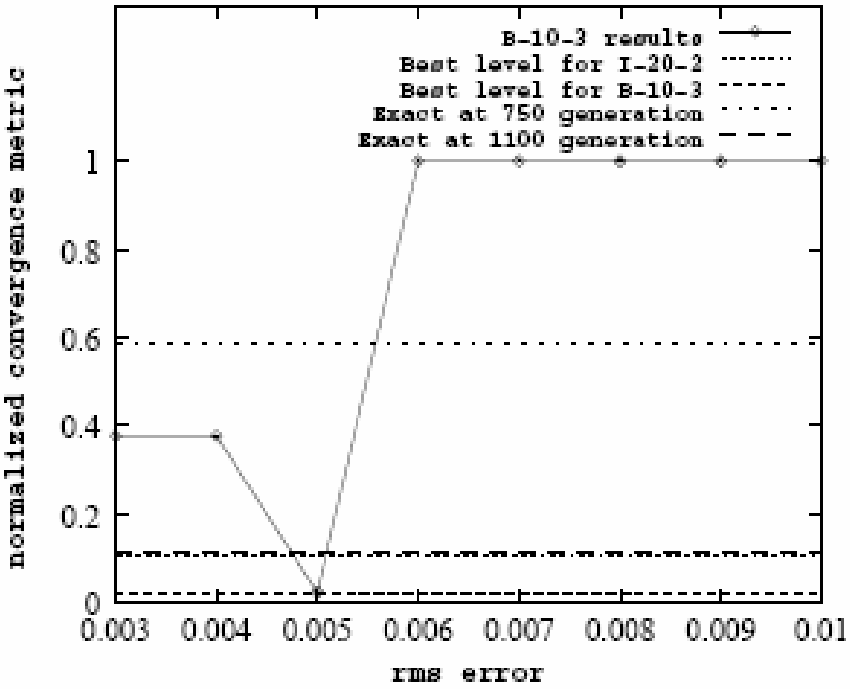


Extreme Trade-off Solutions



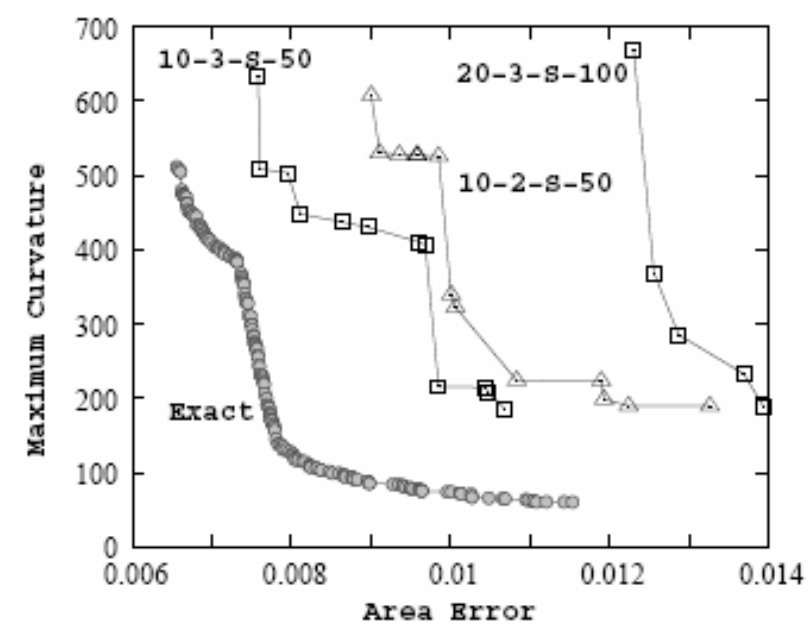
Sensitivity of Chosen RMS Error in ANN

- Convergence metric: Average distance of a front from the best derived
- front of all simulations
- RMS error of 0.005 is the best for B-10-3 model
- Best of I-20-2 is also close



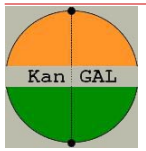
Multi-Objective Kriging

- Perform kriging for each objective independently
- Curve-fitting problem
- Points are chosen randomly for kriging
- Kriging results are not good
- Judicious choice of points necessary



Comments on Approximate Function Computation

- The use of ANN to approximate objective functions from a set of points
- found useful
- Frequency of ANN modeling and number of training points are two
- important parameters
- Batch model of learning is better
- For the test problem, at most 32% savings in eval. observed
- Constraints can also be modeled in a similar fashion
- Further immediate studies are needed in this direction



Conclusions of Lecture 5

- Practice calls for handling
  - Approximate modeling
  - Multiple optima and objectives (later)
  - Robustness and stochasticities (later)
- Above are strong points in favor of EAs
- Current practices are far from being ideal
- Optimization books should also be rewritten

