Evolutionary Practical Optimization

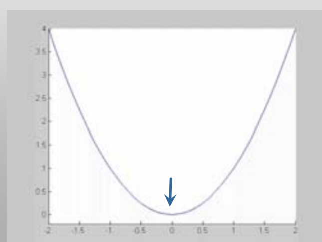
Kalyanmoy Deb*
Deva Raj Chair Professor
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, India
Email: deb@iitk.ac.in
http://www.iitk.ac.in/kangal/deb.htm

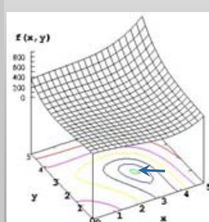
Outline of Tutorial

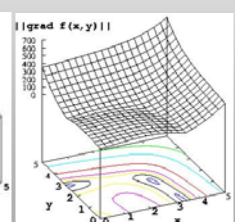
- Optimization fundamentals
- Scope of *optimization* in practice
- Classical *point-by-point* approaches
- Advantages of evolutionary *population-based* approaches
- Scope and flexibility of evolutionary approaches in different practical problem solving tasks
  - A case study
- Summary

Fundamentals of Optimization

- A generic name for minimization and maximization of a function *f(**x**)*
- Everyone knows: *df/dx=0* or $\nabla f(x) = 0$
- Curse of dimensionality, multiple optima



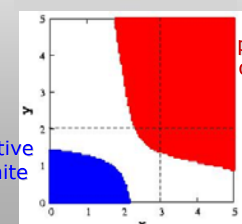




Fundamentals (cont.)

- Concept relates to mathematics
  - Second and higher-order derivatives
    $d^2f/dx^2>0$, minimum
    $d^2f/dx^2<0$, maximum

  if $\nabla^2 f$ is positive definite at x*, it is a minimum
  - Convex: One optimum



GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

Constrained Optimization Basics

- Decision variables: $\mathbf{x} = (x_1, x_2, \ldots, x_n)$
- Constraints restrict some solutions to be feasible

$$\begin{aligned}&\text{Min. } f(\mathbf{x})\\ &\text{s.t. } g_j(\mathbf{x}) \geq 0 \quad j = 1,2,\ldots,J\\ &\quad h_k(\mathbf{x})=0 \quad k = 1,2,\ldots,K\\ &\quad x_i^L \leq x_i \leq x_i^U \quad i = 1,2,\ldots,n\end{aligned}$$

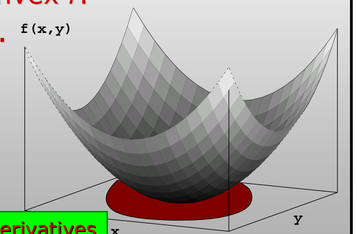
- Equality and inequality constraints
- Minimum of $f(\mathbf{x})$ need not be constrained minimum
- Constraints can be non-linear

Constrained Optimization Basics (cont.)

- Karush-Kuhn-Tucker (KKT) conditions for optimality
  - First-order necessary conditions
  - Convex search space, convex f: KKT point is minimum pt.

$$\nabla f(x) - \sum_{j=1}^{J} u_j \nabla g_j(x) - \sum_{k=1}^{K} v_k \nabla h_k(x) = 0$$

$$\begin{aligned}&g_j(\mathbf{x}) \geq 0 \quad j = 1,2,\ldots,J\\ &h_k(\mathbf{x})=0 \quad k = 1,2,\ldots,K\\ &u_j g_j(\mathbf{x})=0 \quad j = 1,2,\ldots,J\\ &u_j \geq 0 \quad j = 1,2,\ldots,J\end{aligned}$$



Involve Derivatives and solve for roots

Duality Theory in Optimization

- A primal problem has an equivalent dual problem
  - Dimension same as number of constraints
- Dual problem is always concave ($-f$ is convex)
  - But involves a nested optimization
- Theoretical results:
  - Convex problems and in some special cases:
    - Optimal primal and dual function values are same
  - Generic cases:
    - Optimal dual function value underestimates optimal primal function value

Theory is not practical, but prudent

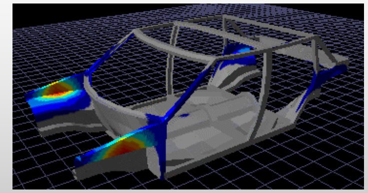
- Theory often not applicable in practice
  - Gradients do not always exist
  - Theory not pragmatic for generic problems
- But good to know
  - Know extent of theory
  - Know limitation of theory
  - Often may lead to better algorithm development
- `No Free Lunch' (NFL) theorem
- Need for *customization* for a problem

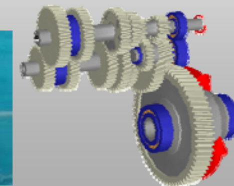
No Free Lunch (NFL) Theorem

- In the context of *optimization*
  - Wolpert and McCardy (1997)
  - Algorithms A1 and A2
  - All possible problems **F**
  - Performances P1 and P2 using A1 and A2 for a fixed number of evaluations
  - P1 = P2
- NFL breaks down for a narrow class of problems or algorithms
- Research effort: Find the best algorithm for a class of problems
  - Unimodal, multi-modal, quadratic etc.

Scope of Optimization in Practice

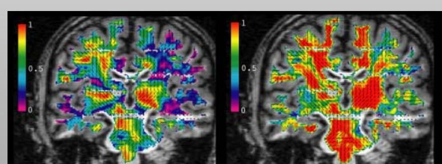
- *Optimal design & manufacturing* for desired goals
- Major application in engineering





Scope of Optimization (cont.)

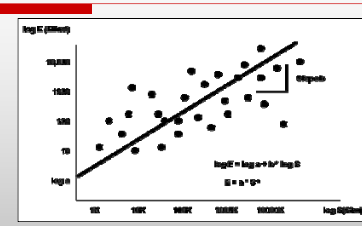
- *Inverse Problems*
  - Output known, find input
  - Often with a goal: minimize distortion, maintain physics, occum's razor (simplest) etc.
  - Tomography, reconstruction, 3D from 2D images
  - Lead to multiple solutions



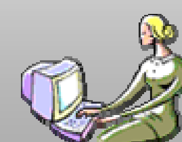
Reverse current in brain models (Johnson, 2006)

Scope of Optimization in Practice (cont.)

- *Parameter optimization* for optimal performance

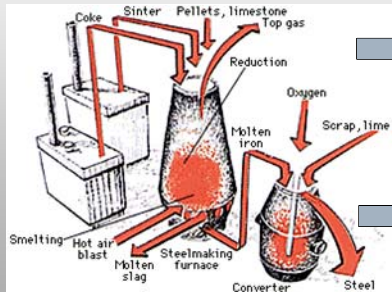


- Scientific experiments, computer experiments

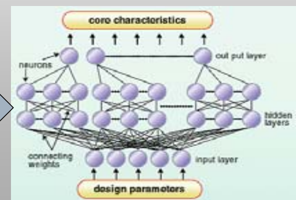


Scope of Optimization in Practice (cont.)

- *Modeling* (system, process)



$$(1)\quad \frac{\partial P}{\partial t} = -\gamma P + \beta\lambda I_A - \kappa P S_A$$
$$(2)\quad \frac{\partial I_A}{\partial t} = r I_A - g_A I_A + \kappa P S_A - \lambda I_A + mb_{B\to A} I_B - mb_{A\to B} I_A$$
$$(3)\quad \frac{\partial S_A}{\partial t} = r S_A - g_A S_A - \kappa P S_A + mb_{B\to A} S_B - mb_{A\to B} S_A$$
$$(4)\quad \frac{\partial I_B}{\partial t} = -g_B I_B - mb_{B\to A} I_B + mb_{A\to B} I_A$$
$$(5)\quad \frac{\partial S_B}{\partial t} = -g_B S_B - mb_{B\to A} S_B + mb_{A\to B} S_A$$



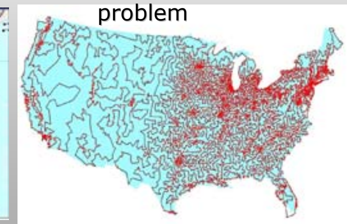
Scope of Optimization in Practice (cont.)

Timetabling

- *Scheduling and planning* (combinatorial optimization)

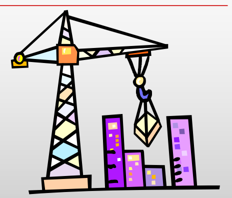
Routing & Scheduling

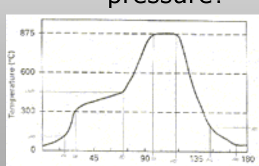
Traveling Salesperson problem

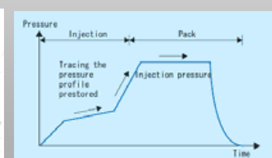


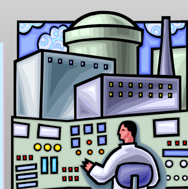
Scope of Optimization in Practice (cont.)

- *Optimal control*
- Time-variant profiles are to be found
  - How to lower load?
  - How to control temp, pressure?



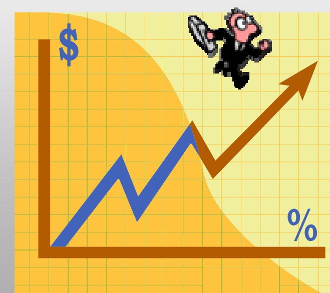






Scope of Optimization in Practice (cont.)

- *Forecasting and prediction*



Scope of Optimization in Practice (cont.)

- *Data mining* (classification, clustering, pattern recognition)

GECCO-08 Tutorial: `Evolutionary Practical Optimization' (K. Deb) 17

Scope of Optimization in Practice (cont.)

- *Machine learning*
  - Designing intelligent systems

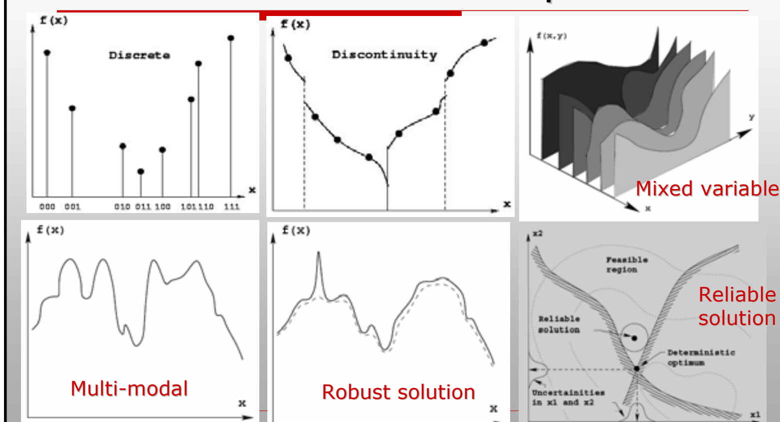
GECCO-08 Tutorial: `Evolutionary Practical Optimization' (K. Deb) 18

Properties of Practical Optimization Problems

- Non-differentiable functions and constraints
- Discontinuous search space
- Discrete search space
- Mixed variables (discrete, continuous, permutation)
- Large dimension (variables, constraints, objectives)
- Non-linear constraints
- Multi-modalities
- Multi-objectivity
- Uncertainties in variables
- Computationally expensive problems
- Multi-disciplinary optimization

GECCO-08 Tutorial: `Evolutionary Practical Optimization' (K. Deb) 19

Different Problem Complexities



GECCO-08 Tutorial: `Evolutionary Practical Optimization' (K. Deb) 20

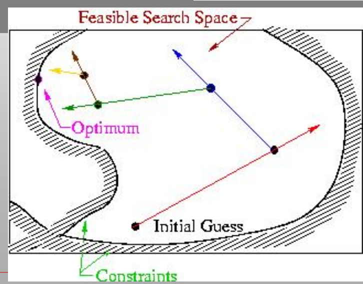
Classical Optimization Methods and Past

- Exact differentiation & root-finding method
  - Intractable and not sufficient for practical problems
- Numerical algorithms
  - Iterative and deterministic
  - Directions based on gradients (mostly)
- Point-by-point approaches

Linear regression: y=mx+b

$$m = \frac{n\sum(xy) - \sum x \sum y}{n\sum(x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - m\sum x}{n}$$

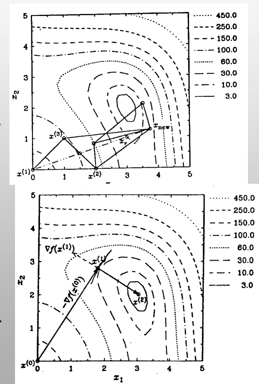


Classical Methods (cont.)

- Direct and gradient based methods
- Convexity assumption
  - No guarantee otherwise
- Local perspectives
- Discreteness cause problems
- Non-linear constraints
- Large-scale application time-consuming
- Serial in nature

Simplex search

Steepest descent



Practical Optimization

With a point in each iteration, scope is limited

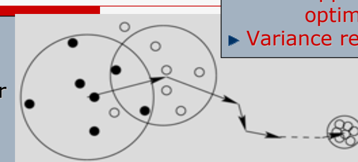
Non-classical, population-based optimization methods may be more flexible and useful

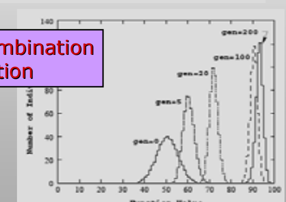
Evolutionary Algorithm (EA) as an Optimizer

- Mean approaches optimum
- Variance reduces

```
begin
Solution Representation
t := 0;  // generation counter
Initialization P(t);
Evaluation P(t);
while not Termination
do
  P'(t)  := Selection (P(t));
  P''(t) := Variation (P'(t));
  Evaluation P''(t);
  P(t+1):= Survivor (P(t),P''(t));
  t := t+1;
od
end
```

Recombination
Mutation





Advantages of EAs

- Applicable in problems where no (good) method is available
  - Discontinuities, non-linear constraints, multi-modalities
  - Discrete variable space
  - implicitly defined models (*if-then-else)*
  - Noisy problems
- Most suitable in problems where multiple solutions are sought
  - Multi-modal optimization problems
  - Multi-objective optimization problems

Taking adv. of operator flexibility

Taking adv. of population

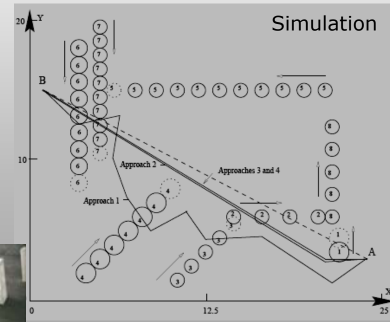
Advantages (cont.)

Robot Navigation

Rulebase:

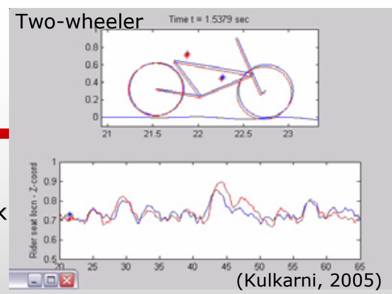
distance \ angle	L	AL	A	AR	R
VN	A	AR	AL	AL	A
N	A	A	AL	A	A
F	A	A	AR	A	A
VF	A	A	A	A	A

- Concept Development
  - An EA solution is a *recipe* or a procedure
  - Example: Seed and how-to-grow principles
  - Not the whole solution, but a construction procedure
  - Evaluation generates the whole solution
- Parallel implementation is easier

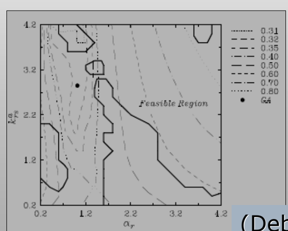


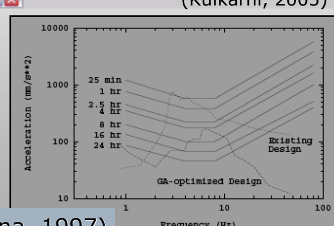
Suspension Design

- Practice is full of non-linearities
- Opt. softwares may get stuck
- Orders of magnitude better than existing design possible



(Kulkarni, 2005)





(Deb and Saxena, 1997)

EAs for Real-parameter optimization

- Decision variables are coded directly, instead of using binary strings
- Recombination and mutation need structural changes
- Selection operator remains the same

Recombination

$$\begin{matrix}(x_1 x_2 \ldots\ldots\ldots\ldots x_n) \\ (y_1 y_2 \ldots\ldots\ldots\ldots y_n)\end{matrix} \Rightarrow ?$$

Mutation

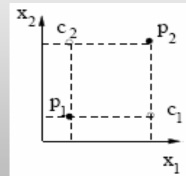
$$(x_1 x_2 \ldots\ldots\ldots\ldots\ldots x_n) \Rightarrow ?$$

- Simple exchanges are not adequate

Naive Recombination

- Crossing at boundaries do not constitute adequate search
  - Least significant digits are taken too seriously

15.345	3.569	→	11.142	3.587
21.142	5.687		25.345	5.669



- Two Remedies:
  - Parent values (variable-wise) need to be blended to each other
  - Vector-wise recombination

Variable-wise Blending of Parents

- Use a probability distribution to create child
- Different implementations since 1991:
  - Blend crossover (BLX-α), 1991
  - Simulated binary crossover (SBX-β), 1995
  - Fuzzy recombination (FR-d), 1995
  - Self-adaptive evolution strategy (ES-τ), 1987
  - Differential evolution (DE-CR-F), 1996
  - Particle swarm optimization (PSO-param.)
- Main feature: Difference between parents used to create children
  - Provides a self-adaptive property

Simulated Binary Crossover (SBX)

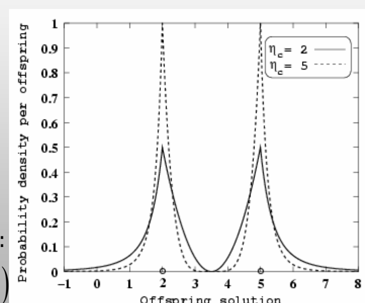
$$\beta = \left|\frac{c_1 - c_2}{p_1 - p_2}\right|$$

- Step 1: Choose a random number

$$u \in [0,1].$$

- Step 2: Calculate β_q:

$$\beta_q = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \le 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases}$$



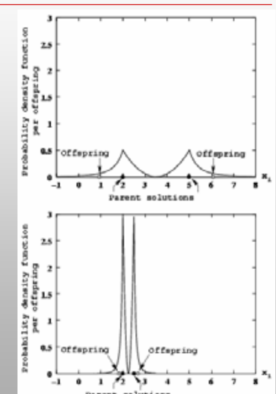
- Step 3: Compute two offspring:

$$c_1 = 0.5\left((1+\beta_q)p_1 + (1-\beta_q)p_2\right)$$

$$c_2 = 0.5\left((1-\beta_q)p_1 + (1+\beta_q)p_2\right)$$

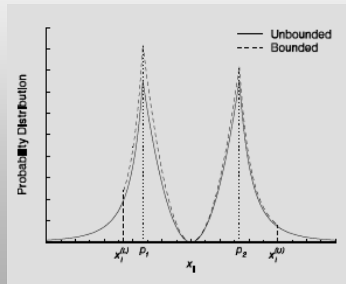
Properties of SBX Operator

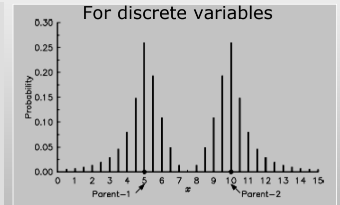
- If parents are distant, distant offspring are likely
- If parents are close, offspring are close to parents
- Self-adaptive property



Variations of SBX

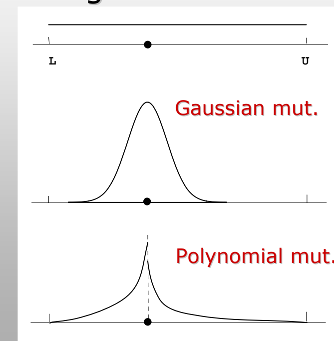
- For bounded and discrete variables





Real-Parameter Mutation Operators

- Idea: Create a neighboring solution
  - Random mutation
  - Normally distributed mutation
  - Non-uniform mutation
  - Extensions to bounded and discrete cases exist



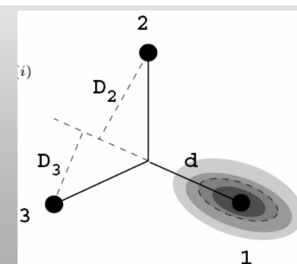
Vector-Wise Recombination Operators

- Variable-wise recombination cannot capture nonlinear interactions
- Recombine parents as vectors
  - Parent-centric recombination (PCX)
  - Unimodal normally-distributed crossover (UNDX)
  - Simplex crossover (SPX)
- Difference between parents is used to create offspring solutions
- DE, PSO, CMA-ES, and others

Parent Centric Crossover Operator

- μ parents create a offspring
- Each parent has its turn
- $e^{(i)}$ orthonormal bases spanning the subspace perpendicular to $d^{(p)}$
- w_ς and w_η are user-defined parameters, controlling extent of search

$$\vec{y} = \vec{x}_p + w_\varsigma |\vec{d}^{(p)}| + \sum_{i=1,\ i\neq p}^{\mu} w_\eta \bar{D} \vec{e}^{(i)}$$



Generalized Generation Gap (G3) Model (Steady-state approach)

1. Select the best parent and μ-1 other parents randomly
2. Generate λ offspring using a recombination scheme
3. Choose two parents at random from the population
4. Form a combination of two parents and λ offspring, choose best two solutions and replace the chosen two parents

- Parametric studies with λ and N

Three Test Problems

- 20-variable problems, x_i = [-10,-5] for all i
- 50 runs performed
- F^t_{elp}=0.5(10^6), F^t_{sch}=1(10^6), F^t_{ros}=1(10^6)

$$F_{\rm elp} = \sum_{i=1}^{n} i x_i^2 \quad \text{(Ellipsoidal)}$$

$$F_{\rm sch} = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \quad \text{(Schwefel)}$$

$$F_{\rm ros} = \sum_{i=1}^{n-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right) \quad \text{(Rosenbrock)}$$

Quasi-Newton Method

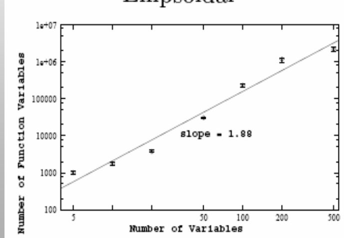
- Accuracy obtained by G3+PCX is 10^{-20}

Func.	FE	Best	Median	Worst
$F_{\rm elp}$	6,000	$8.819(10^{-24})$	$9.718(10^{-24})$	$2.226(10^{-23})$
$F_{\rm sch}$	15,000	$4.118(10^{-12})$	$1.021(10^{-10})$	$7.422(10^{-9})$
$F_{\rm ros}$	15,000	$6.077(10^{-17})$	$4.046(10^{-10})$	3.987
$F_{\rm elp}$	8,000	$5.994(10^{-24})$	$1.038(10^{-23})$	$2.226(10^{-23})$
$F_{\rm sch}$	18,000	$4.118(10^{-12})$	$4.132(10^{-11})$	$7.422(10^{-9})$
$F_{\rm ros}$	26,000	$6.077(10^{-17})$	$4.046(10^{-10})$	3.987

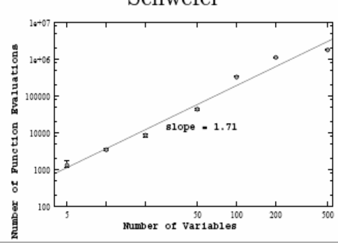
Scalability Study

- Accuracy 10^{-10} is set

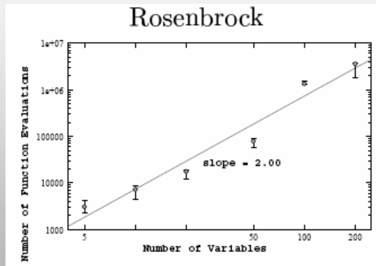
Ellipsoidal



Schwefel



Scalability Study (cont.)



- All polynomial complexity $O(n^{1.7})$ to $O(n^2)$ similar to those reported by CMA-ES approach (Hansen and Ostermeier, 2001)

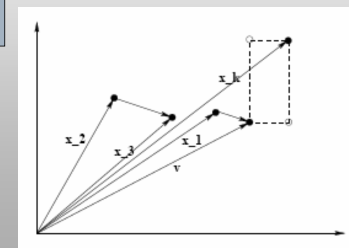
Differential Evolution (DE)

1. Start with a pool of random solutions
2. Create a child v
3. x_k and v are recombined with *p*

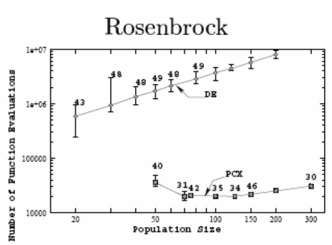
- Difference of parents in creating a child is important
- A number of modifications exist

– $\mathbf{v} = \mathbf{x}^{(1)} + \lambda(\mathbf{x}^{(2)} - \mathbf{x}^{(3)})$

– $y_i = \begin{cases} v_i, & \text{with a prob. } p \\ \mathbf{x}_i^{(k)}, & \text{else} \end{cases}$



DE Results



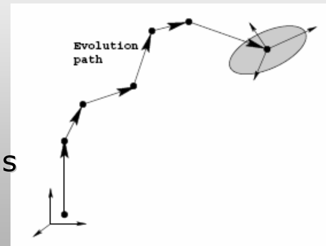
	Best	Median	Worst
		F_{elp}	
DE	9,660	12,033	20,881
G3	5,826	6,800	7,728
		F_{sch}	
DE	102,000	119,170	185,590
G3	13,988	15,602	17,188
		F_{ros}	
DE	243,800	587,920	942,040
G3	16,508	21,452	25,520

Particle Swarm Optimization (PSO)

- Kennedy and Eberhart, 1995
- Particles fly through the search space
- Velocity dynamically adjusted
- $x_i = x_i + v_i$
- $v_i = v_i + c_1 rnd()(p_{i,best} - x_i) + c_2 rnd()(p_g - x_i)$
- p_i: best position of i-th particle
- p_g: position of best particle so far
  - 1st term: momentum part (history)
  - 2nd term: cognitive part (private thinking)
  - 3rd term: social part (collaboration)

CMA-ES
(Hansen & Ostermeier, 1996)

- Selecto-mutation ES is run for n iterations
- Sucessful steps are recorded
- They are analyzed to find uncorrelated basis directions and strengths
- Required $O(n^3)$ computations to solve an eigenvalue problem
- Rotation invariant



CMA-ES On Three Test Problems

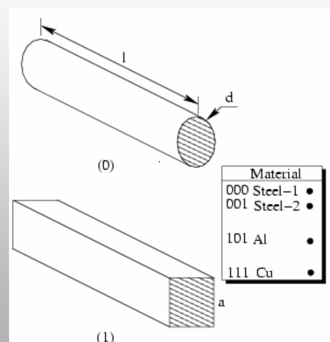
EA	F_{elp} Best	F_{elp} Median	F_{elp} Worst	F_{sch} Best	F_{sch} Median	F_{sch} Worst
CMA-ES	8,064	8,472	8,868	15,096	15,672	16,464
DE	9,660	12,033	20,881	102,000	119,170	185,590
G3+PCX	5,826	6,800	7,728	13,988	15,602	17,188
	F_{ros}					
CMA-ES	29,208	33,048	41,076			
DE	243,800	587,920	942,040			
G3+PCX	16,508	21,452	25,520			

Accuracy 1×10^{-20}

Mixed-Variable Optimization:
Handling mixed type of variables

- Treat type of cross-sections, materials, etc. as decision variables
- A mixed representation:

  (1) 14 23.457 (101)
  - (1): circular or square cross-section
  - 14: diameter/side
  - 23.457: length
  - (101): material
- Permutation + real + cont.
- Complete optimization
- Deb and Goel, ASME-JMD, 1997



Constraint Handling:
Handling non-linear constraints

- Inequality constraints $(g_j(x) \geq 0)$ penalized for violation:

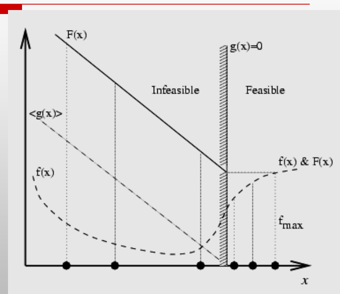
$$F(x) = f(x) + \sum_{j=1}^{J} R_j \langle g_j(x) \rangle^2$$

  - <a> = a if a is −ve, 0 otherwise
- Performance sensitive to penalty parameters

R	≤ 50%	Infeasible	Best	Median	Worst
10^0	12	13	2.41324	7.62465	483.50177
10^1	12	0	3.14206	4.33457	7.45453
10^3	1	0	3.38227	5.97060	10.65891
10^6	0	0	3.72929	5.87715	9.42353

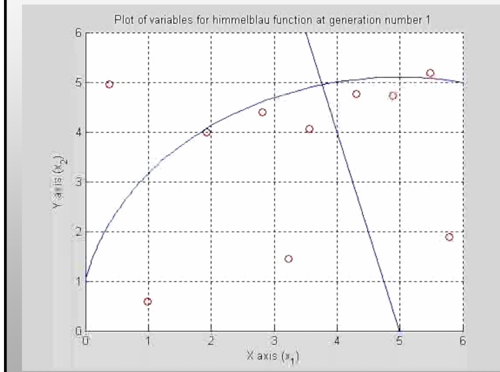
A Penalty-Parameter-less Population-based Approach

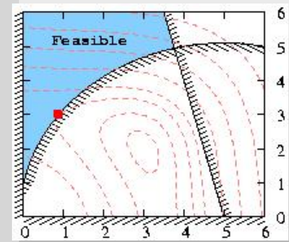
- Modify tournament sel.:
  - A feasible is better than an infeasible
  - For two feasibles, choose the one with better *f*
  - For two infeasibles, choose the one with smaller constraint violation $\left(\sum_j \langle g_j(x) \rangle\right)$
  - (Deb, CMAME 2000)



$$F(x) = \begin{cases} f(x), & \text{if } g_j(x) \geq 0, \forall j \in J \\ f_{\max} + \sum_{j=1}^{J} \langle g_j(x) \rangle, & \text{otherwise} \end{cases}$$

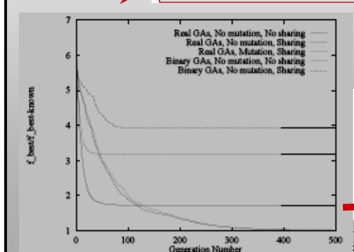
A Computer Simulation





Welded-Beam Design Problem

Method	Muta-tion	shar ing	ε ≤ 1%	≤ 2%	≤ 5%	≤ 10%	≤ 20%	≤ 50%	> 50%
Maximum generations = 500									
TS-B	No	No	0	0	0	0	0	0	50
TS-B	No	Yes	0	0	0	0	0	0	50
TS-R	No	No	0	0	1	4	8	16	34
TS-R	No	Yes	28	36	44	48	50	50	0
Maximum generations = 4,000									
TS-R	No	Yes	28	37	44	48	50	50	0
TS-R	Yes	Yes	50	50	50	50	50	50	0



Case	Best	Median	Worst
1	4.939027	7.183082	12.084255
2	3.820984	8.899963	14.298933
3	2.442714	3.834121	7.444246
4	2.381191	2.392892	2.645833
5	2.381187	2.392025	2.645833
6	2.381447	2.382634	2.383554

Large-Scale Optimization:
Handling a large number of variables

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

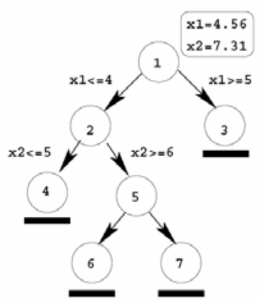
Casting Scheduling

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 1	2	3	4	5	6	7	8	9	10	Utilization/ Cruc. Size	Efficiency (%)
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: Population Size	Efficiency	Function Eval.	Real-coded GAs: 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

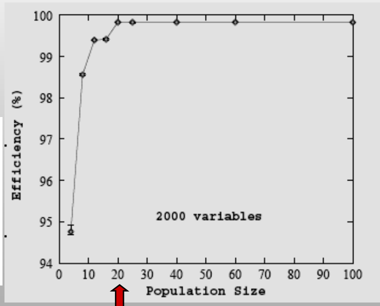
A Customized EA

- 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

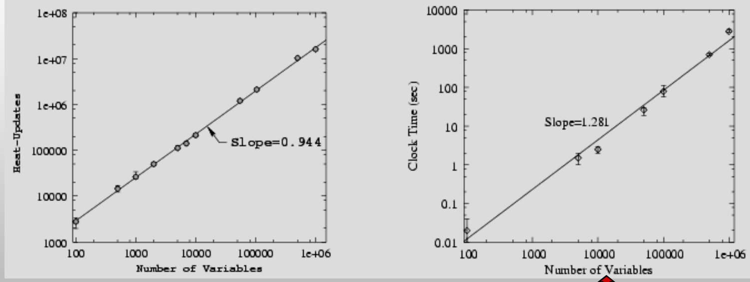
A Customized GA: Optimal Population Size

- N=2,000 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



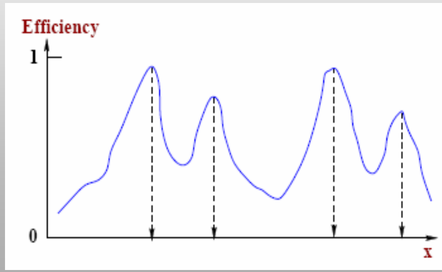
Scale-Up Results



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

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

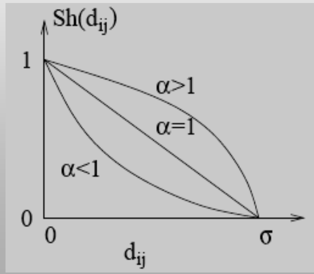


Niching and Sharing Function

- Goldberg and Richardson (1997)

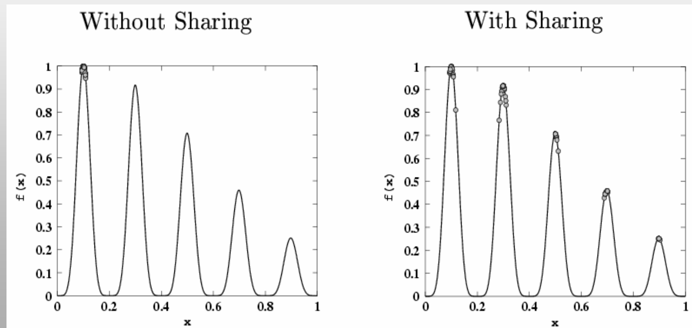
$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma}\right)^{\alpha} & \text{if } d_{ij} \leq \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

- 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 operator



Simulation Results

- Inclusion of niche-formation strategy



Other Niching Approaches

- Clearing approach
- Clustering approach
- Crowding approach
- Pre-selection approach
- Restrictive tournament selection approach
- All require at least one tunable parameter
- Some parameter-less procedures suggested recently
  - More application to real problems needed

Optimization with Meta-Models:
Handling computationally expensive problems

- 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

Response Surface Method (RSM)

- Box and Wilson (1951)
- Model: Error is independent of x
- Usually a parametric linear or quadratic approx:

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

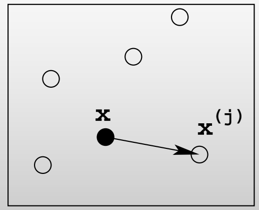
- β_i determined by least-square regression from observed data
  - Optimize to minimize error: Find mean β_i
  - Variance of β_i determine predictive capability
- Usually applied for k<10

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

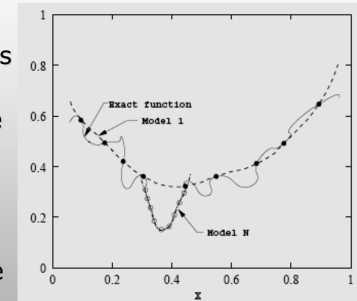


$$Y(\mathbf{x}) = \sum_{k=1}^{K} \beta_k \mathbf{f}_k(\mathbf{x}) + Z(\mathbf{x})$$

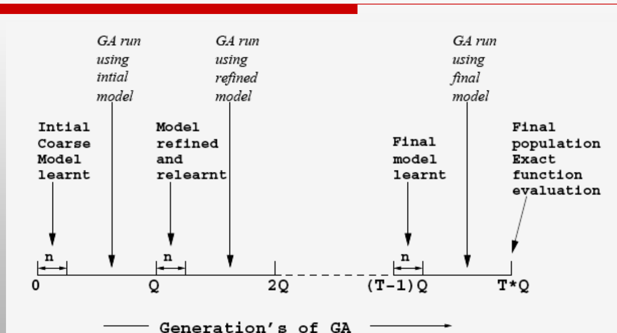
- Local variation Z(x) and β computed through spatial coorelation fun.
- Maximum likelihood fun. is optimized

Successive Modeling 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



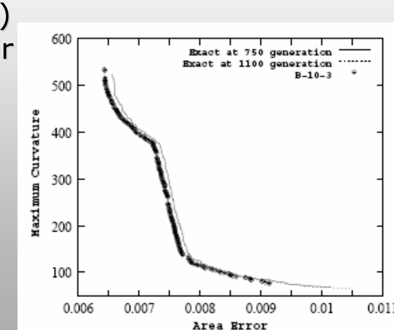
Generation-wise Sketch of Proposed Approach



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

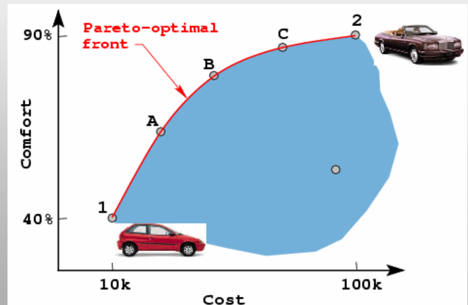
A Case Study: Saving Function Evaluations

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



Multi-Objective Optimization:
Handling multiple conflicting objectives

- We often face them



Evolutionary Multi-Objective Optimization (EMO)

Step 1 :

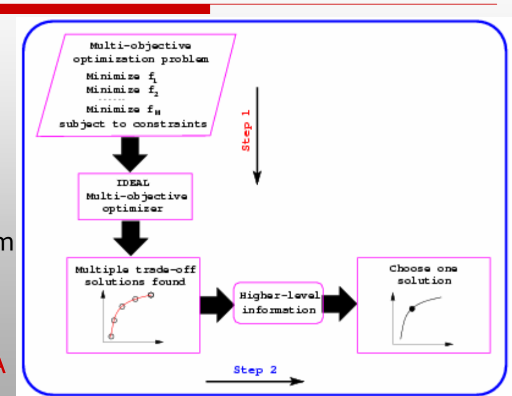
Find a set of Pareto-optimal solutions

Step 2 :

Choose one from the set

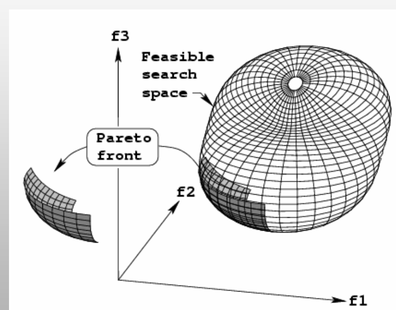
(Deb, 2001)

- Ideal for an EA



Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II)

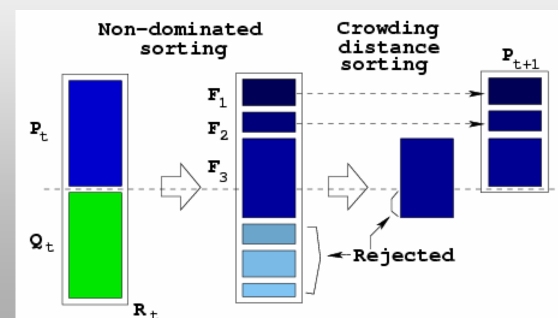
- NSGA-II can extract Pareto-optimal frontier
- Also find a well-distributed set of solutions
- iSIGHT and modeFrontier adopted NSGA-II



Fast-Breaking Paper in Engineering by ISI Web of Science (Feb'04)

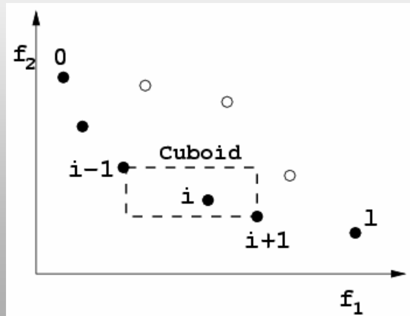
NSGA-II Procedure

Elites are preserved
Non-dominated solutions are emphasized



NSGA-II (cont.)

Diversity is maintained

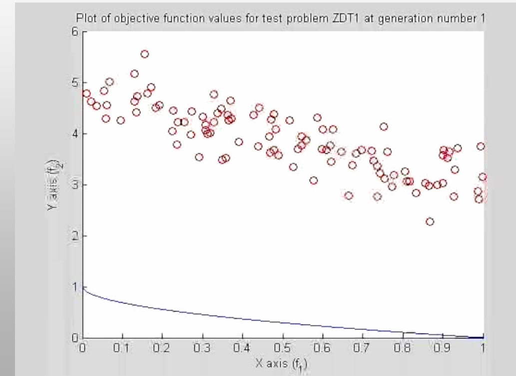


Overall Complexity
$O(N \log^{M-1} N)$

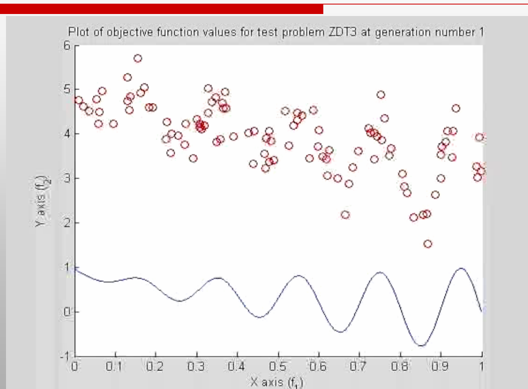
Improve diversity by

- k-mean clustering
- Euclidean distance measure
- Other techniques

Simulation on ZDT1



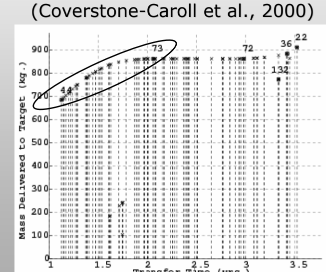
Simulation on ZDT3

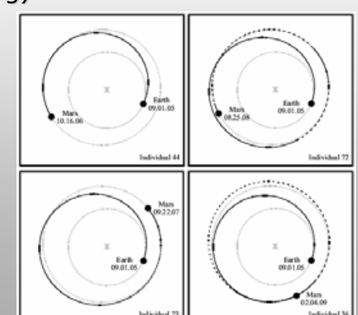


EMO Applications

- Identify different trade-off solutions for choosing one (Better decision-making)
- Inter-planetary trajectory

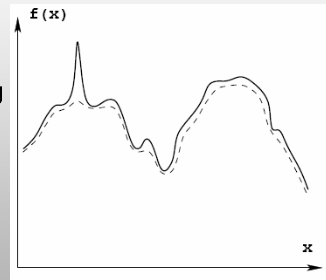
(Coverstone-Caroll et al., 2000)





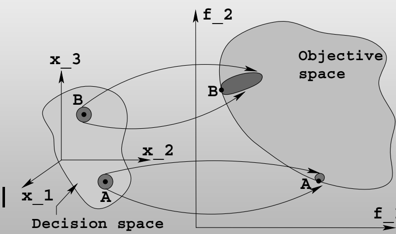
Robust Optimization
Handling uncertainties in variables

- Parameters are uncertain and sensitive to implementation
  - Tolerances in manufacturing
  - Material properties are uncertain
  - Loading is uncertain
- Who wants a sensitive optimum solution?
- Single-objective robust EAs (Branke and others)



Multi-Objective Robust Solutions

- Solutions are averaged in δ-neighborhood
- Not all Pareto-optimal points may be robust
- A is robust, but B is not
- Decision-makers will be interested in knowing robust part of the front



Multi-Objective Robust Solutions of Type I and II

- Similar to single-objective robust solution of type I

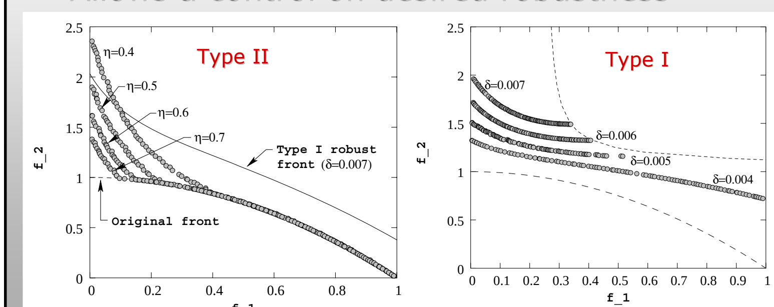
$$\begin{array}{ll} \textit{Minimize} & (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x})), \\ \textit{subject to} & \mathbf{x} \in \mathcal{S}, \end{array}$$

- Type II

$$\begin{array}{ll} \textit{Minimize} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \textit{subject to} & \frac{\|\mathbf{f}^p(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|}{\|\mathbf{f}(\mathbf{x})\|} \le \eta, \\ & \mathbf{x} \in \mathcal{S}. \end{array}$$

Robust Frontier for Two Objectives

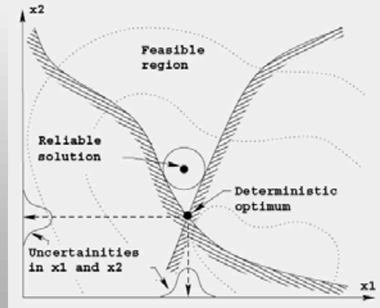
- Identify robust region
- Allows a control on desired robustness



Reliability-Based Optimization:
Making designs safe against failures

- Deterministic optimum is not usually reliable
- Reliable solution is an interior point
- Chance constraints with a given reliability

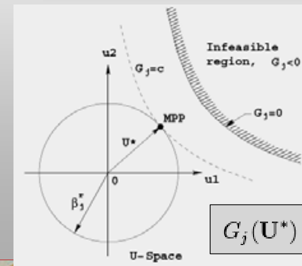
Minimize $\mu_f + k\sigma_f$
Subject to $Pr(g_j(x)\geq 0) \geq \beta_j$
β_j is user-supplied



Statistical Procedure:
Check if a solution is reliable

- PMA approach

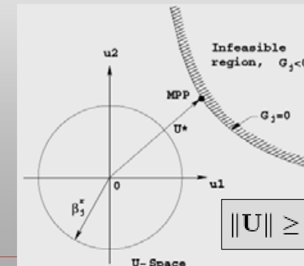
$$\text{Minimize } G_j(\mathbf{U}), \quad \text{Subject to } \|U\| = \beta_j^r,$$



$G_j(\mathbf{U}^*) \geq 0.$

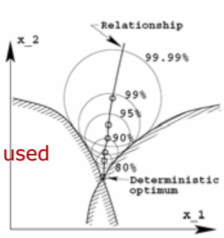
- RIA approach

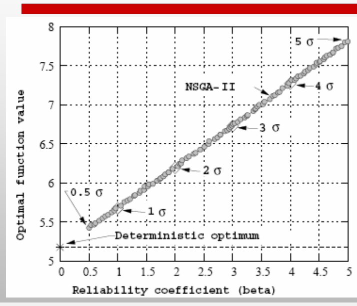
$$\text{Minimize } \|\mathbf{U}\|, \quad \text{Subject to } G_j(\mathbf{U}) = 0.$$



$\|\mathbf{U}\| \geq \beta_j^r.$

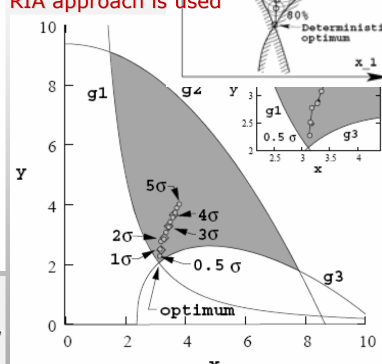
Multiple Reliability Solutions:
Get a better insight





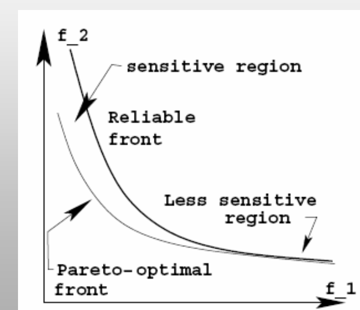
Maximize $x + y$,
Subject to $g_1(x,y) \equiv \frac{1}{20}x^2y - 1 \geq 0,$
$g_2(x,y) \equiv \frac{1}{30}(x+y-5)^2 + \frac{1}{120}(x-y-12)^2 - 1 \geq 0,$
$g_3(x,y) \equiv \frac{80}{x^2+8y+5} - 1 \geq 0,$
$0 \leq x, y \leq 10.$

RIA approach is used



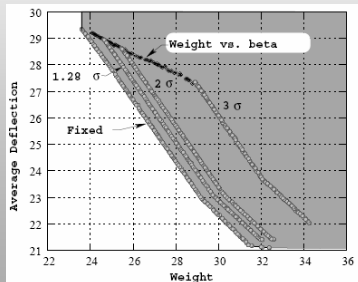
Multi-Objective Reliable Frontier

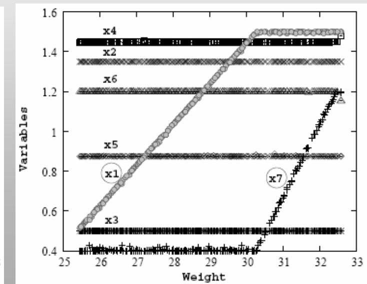
- Instead of finding deterministic Pareto-optimal front, find reliable front
- Chance constraints
- Objectives as they are
- PMA approach is used



Multi-Objective Reliability-Based Optimization

- Reliable fronts show rate of movement
- What remains unchanged and what gets changed!

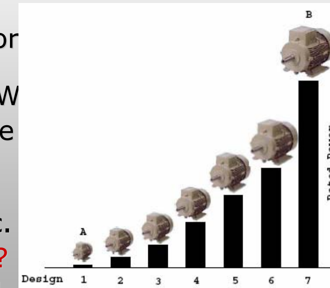




Innovization:

Discovery of Innovative design principles through optimization

- Understand important design principles in a routine design scenario
  - Example: Electric motor design with varying ratings, say 1 to 10 kW
    - Each will vary in size and power
    - Armature size, number of turns etc.
  - How do solutions vary?
    - Any common principles!



In Search of Common Optimality Properties

Fritz-John Necessary Condition:

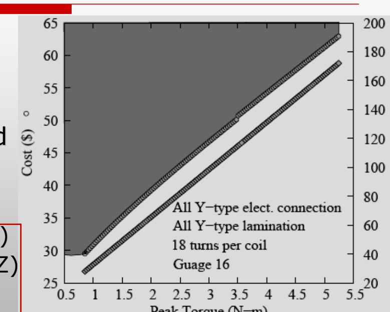
Solution x^* satisfy

1. $\sum_{m=1}^{M} \lambda_m \nabla f_m(x^*) - \sum_{j=1}^{J} u_j \nabla g_j(x^*) = 0,$ and
2. $u_j g_j(x^*) = 0$ for all $j = 1, 2, 3, \ldots, J$
3. $u_j \geq 0$, $\lambda_j \geq 0$, for all j and $\lambda_j > 0$ for at least one j

- To use above conditions requires differentiable objectives and constraints
- Yet, it lurks existence of some properties among Pareto-optimal solutions

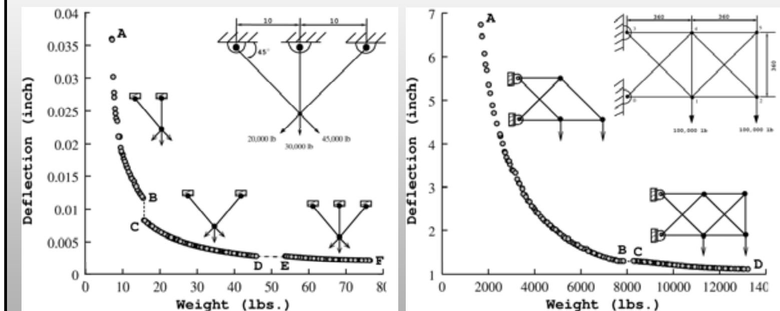
Brushless DC Permanent Magnet Motor Design

- Five variables (all discrete), three constraints
- Non-convex, disconnected P-O front
- *Innovizations:*
  - Connection: Y (betn. Y & Δ)
  - Lamination Type: Y (X, Y, Z)
  - 1 out of 16 wire guages
  - 18 turns per coil (10,80)
  - *Increase length by linearly adding more laminations*



Recipe for design

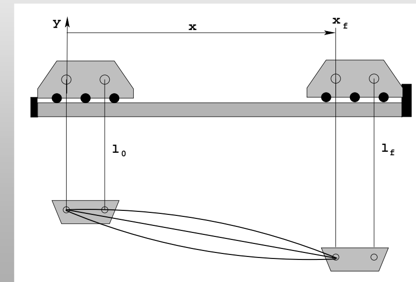
Revealing Salient Insights:
Truss Structure Design

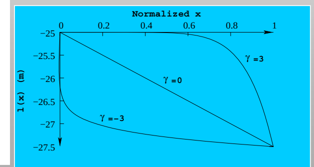


(Deb, Khan and Jindal, 2000)

Overhead Crane Maneuvering

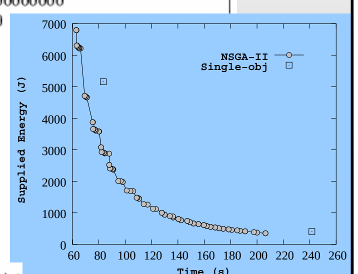
- Minimize time of operation
- Minimize operating energy





Innovization Results

γ	Force (N)	Pattern
2.98	149.26	1111111100100010000000000000000000000000000000000
3.00	187.76	111111110010001000000000000000000000000000
3.00	238.56	1111111100100010000000000000000000
3.00	327.86	1111101100100110000000000000
2.98	427.93	11111011001001110000000
3.00	544.94	1111111100100100
2.87	629.62	111111110010001
2.88	818.99	110111010010001

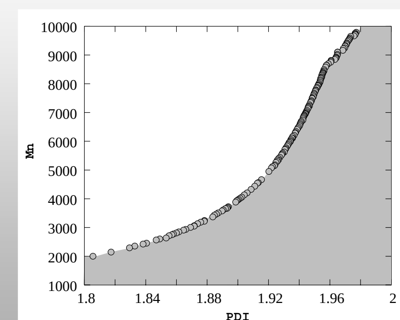


- NSGA-II finds trade-off and interesting properties
  - Push early
  - Lower later $E = mg(1 - \cos\theta)l$

Epoxy Polymerization

- Three ingredients added hourly
- 54 ODEs solved for a 7-hour simulation
- Maximize chain length (Mn)
- Minimize polydispersity index (PDI)
- Total 3x7 or 21 variables

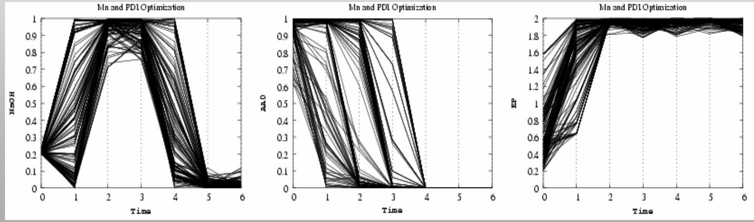
(Deb et al., 2004)



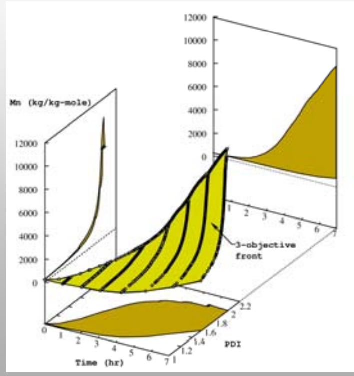
A non-convex frontier

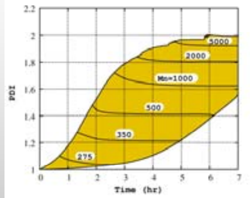
Epoxy Polymerization (cont.)

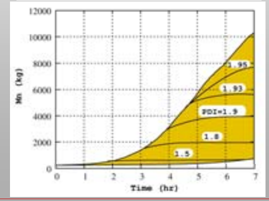
- Patterns emerge among obtained solutions
- Chemical significance unveiled



Innovized Principles: *An Optimal Operating Chart*







A Case Study for Practical Optimization: *A Hydro-Thermal Power Dispatch Problem*

- Thermal power generation P^s_{sm} and hydroelectric power generation P^h_{hm} are variables

$$\begin{aligned}
\text{Minimize} \quad & f_1(\mathrm{x}) = \textstyle\sum_{m=1}^{M} \sum_{s=1}^{N_s} t_m [a_s + b_s P^{\mathrm{s}}{}_{sm} + c_s (P^{\mathrm{s}}{}_{sm})^2 + |d_s \sin(e_s(P^{\mathrm{s}}{}_{s,\min} - P^{\mathrm{s}}{}_{sm}))|], \\
\text{Minimize} \quad & f_2(\mathrm{x}) = \sum_{m=1}^{M} \sum_{s=1}^{N_s} t_m [\alpha_s + \beta_s P^{\mathrm{s}}{}_{sm} + \gamma_s (P^{\mathrm{s}}{}_{sm})^2 + \eta_s \exp(\delta_s P^{\mathrm{s}}{}_{sm})], \\
\text{subject to} \quad & \textstyle\sum_{s=1}^{N_s} P^{\mathrm{s}}{}_{sm} + \sum_{h=1}^{N_h} P^{\mathrm{h}}{}_{hm} - P_{Dm} - P_{Lm} = 0, \quad m = 1, 2 \ldots, M, \\
& \textstyle\sum_{m=1}^{M} t_m (a_{0h} + a_{1h} P^{\mathrm{h}}{}_{hm} + a_{2h} (P^{\mathrm{h}}{}_{hm})^2) - W_h = 0, \quad h = 1, 2, \ldots, N_h. \\
& P^{\mathrm{h}}{}_{h,\min} \le P_{hm} \le P^{\mathrm{h}}{}_{h,\max}, \quad h = 1, 2, \ldots, N_h, m = 1, 2, \ldots, M, \\
& P^{\mathrm{s}}{}_{s,\min} \le P_{sm} \le P^{\mathrm{s}}{}_{s,\max}, \quad s = 1, 2, \ldots, N_s, m = 1, 2, \ldots, M.
\end{aligned}$$

- Minimize Cost (non-differentiable) and NOx emission
- Power balance and water head limits
- Known power demand P_{Dm} with time m

GA Coding and Constraint Handling

- Four thermal, two hydroelectric, M time steps

$$((\underbrace{P^{\mathrm{h}}{}_{11}\ P^{\mathrm{h}}{}_{21}}_{\mathcal{H}_1}\ \underbrace{P^{\mathrm{s}}{}_{11} P^{\mathrm{s}}{}_{21} P^{\mathrm{s}}{}_{31} P^{\mathrm{s}}{}_{41}}_{\mathcal{T}_1})(\underbrace{P^{\mathrm{h}}{}_{12}\ P^{\mathrm{h}}{}_{22}}_{\mathcal{H}_2}\ \underbrace{P^{\mathrm{s}}{}_{12} P^{\mathrm{s}}{}_{22} P^{\mathrm{s}}{}_{32} P^{\mathrm{s}}{}_{42}}_{\mathcal{T}_2}) \cdots (\underbrace{P^{\mathrm{h}}{}_{1M}\ P^{\mathrm{h}}{}_{2M}}_{\mathcal{H}_M}\ \underbrace{P^{\mathrm{s}}{}_{1M} P^{\mathrm{s}}{}_{2M} P^{\mathrm{s}}{}_{3M} P^{\mathrm{s}}{}_{4M}}_{\mathcal{T}_M})).$$

- Two sets of constraints:
  - Handle water availability constraint first
  - Quadratic constraint, repair to find $P^h_{h\mu}$

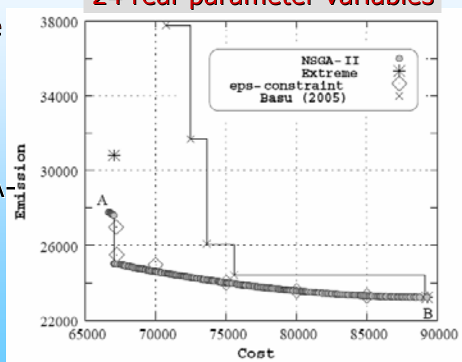
$$(P^{\mathrm{h}}{}_{h\mu})^2 + \frac{a_{1h}}{a_{2h}} P^{\mathrm{h}}{}_{h\mu} + \frac{1}{t_\mu a_{2h}} \left(-W_h + a_{0h} T + \sum_{\substack{m=1 \\ m \ne \mu}}^{M} t_m a_{1h} P^{\mathrm{h}}{}_{hm} + \sum_{\substack{m=1 \\ m \ne \mu}}^{M} t_m a_{2h} (P^{\mathrm{h}}{}_{hm})^2 \right) = 0.$$

  - If not within bounds, penalize
  - Similarly, handle other constraint set

NSGA-II Simulation and Verifications

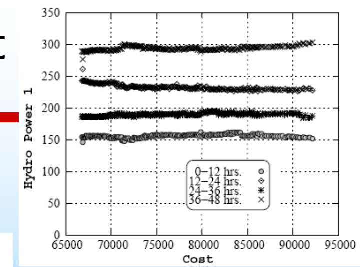
4 time steps of 12 hr. each, 24 real-parameter variables

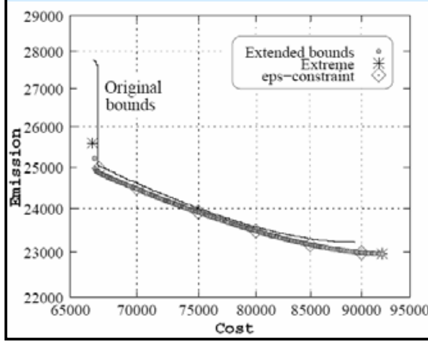
- Single-objective opt. for extreme solns.
- Intermediate ε-constraint optimizations
- Better than a SA-approach with naive penalty
- Min-cost soln. difficult to optimize

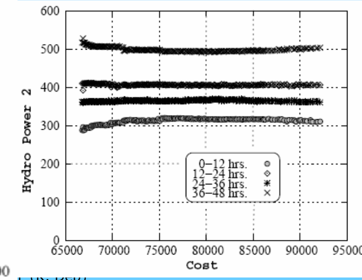


Bounds Too Tight (250 & 500 MW)

- Increase upper bounds to 350, 600 MW
- Better behaved P-O frontier







Verify Obtained Solutions Through KKT Conditions

$$\begin{aligned} \lambda_1\nabla f_1(\mathbf{x}) + \lambda_2\nabla f_2(\mathbf{x}) + \textstyle\sum_{k=1}^{M} v_k^{(1)}\nabla h_k^{(1)}(\mathbf{x}) + \sum_{l=1}^{N_h} v_l^{(2)}\nabla h_l^{(2)}(\mathbf{x}) + \\ \textstyle\sum_{j=1}^{M(N_h+N_s)} u_j^{(L)}\nabla g_j^{(L)}(\mathbf{x}) + \sum_{j=1}^{M(N_h+N_s)} u_j^{(U)}\nabla g_j^{(U)}(\mathbf{x}) &= 0, \\ h_k(\mathbf{x}) &= 0, \\ g_j(\mathbf{x}) &\le 0, \\ u_j g_j(\mathbf{x}) &= 0, \\ u_j &\ge 0. \end{aligned}$$

- f_1 broken into two parts, s_i in [-1,1]

$$\nabla(f_1)_i^1(\mathbf{x}) = \begin{cases} t_m\left(b_s + 2c_s x_i\right), & \text{for } i \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases}$$

$$\nabla(f_1)_i^2(\mathbf{x}) = \begin{cases} -s_i\left(t_m e_s d_s \cos(e_s(x^{\min} - x_i))\right), & \text{for } i \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases}$$

- KKT conditions reduce to following: (Ay=b)

$$\lambda_1\left(\nabla f_1^1(\mathbf{x}) + s_i D_i\right) + \lambda_2\nabla f_2(\mathbf{x}) + \sum_{k=1}^{M} v_k^{(1)}\nabla h_k^{(1)}(\mathbf{x}) + \sum_{k=1}^{N_h} v_k^{(2)}\nabla h_k^{(2)}(\mathbf{x}) = 0.$$

Error Minimization and KKT Results

- Solutions are close to KKT points, if error is close to zero, λ≥0, s_i in [-1,1]

$$\text{Minimize}_{\mathbf{s}} \quad \tilde{e}(\mathbf{s}) = \|b - A(\mathbf{s})\ \bar{y}(\mathbf{s})\|/\|b\|,$$
$$\text{subject to} \quad s_i \in [-1,1], \quad \forall i.$$

Soln.	(f_1, f_2)	|s|	i near kink	s	$\tilde{e}(\mathbf{s})$	λ_1/λ_2
Feasible Solutions at Final Generation						
1	(92120.45, 22963.09)	0	–	–	0.0024	0.000
2	(89929.82, 22991.40)	0	–	–	0.0169	0.019
3	(84997.68, 23169.00)	2	(5,21)	(0.99, -0.60)	0.0141	0.051
4	(79966.72, 23488.10)	4	(5,9,17,21)	(0.69, -0.92, 0.87, -0.38)	0.0126	0.077
5	(75023.50, 23920.95)	7	(4,5,9,11, 17,21,24)	(0.88, 0.45, -0.86, 0.88, 0.56, -0.63, -0.90)	0.0207	0.099
6	(70018.99, 24467.52)	11	(3,4,5,9, 11,12,13,15, 17,21,24)	(-0.85, 0.92, 0.45, -0.52 0.89, -0.92, -0.86, 0.90 0.57, -0.20, -0.55)	0.0241	0.125

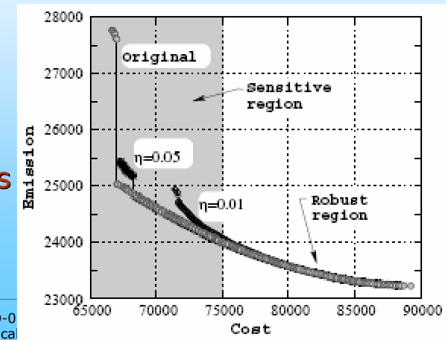
Solutions in Earlier Generations are Not Close to KKT Points

- λ<0, error high

Soln.	(f_1, f_2)	|s|	i near kink	s	$\tilde{e}$(s)	λ_1/λ_2
Feasible Solutions at Generation 10						
1	(83900.92, 24760.04)	0	–	–	0.2915	0.066
2	(82273.16, 25924.39)	1	(12)	(-1)	0.3469	0.051
3	(81281.44, 26298.55)	4	(5,6,9,22)	(-1,1,1,1)	0.3512	–0.078
Feasible Solutions at Generation 30						
1	(83145.39, 23610.54)	1	(6)	(-1)	0.1265	0.056
2	(73140.35, 24720.39)	4	(3,6,12,18)	(-0.57,1.00,1.00,1.00)	0.1931	0.097
3	(71869.35, 25234.02)	7	(5,6,11,12 16,18,23)	(1.00 -0.64 0.31 -1.00 1.00,-1.00,1.00)	0.2338	0.105
Feasible Solutions at Generation 70						
1	(81952.12, 23425.20)	2	(3,23)	(-0.42, 1.00)	0.0515	0.065
2	(74994.34, 24045.11)	4	(6,11,15,17)	(-1.00,0.42,-0.81,0.88)	0.0713	0.096
3	(70424.45, 24922.91)	8	(5,6,11,16 17,18,23,24)	(-0.93,-0.46,0.52,1.00 1.00,-1.00,1.00,-1.00)	0.1459	0.107

Robust Frontier

- Generator values are difficult to obtain exactly
- Power generation ±5 MW
- Max_i $\Delta f_i/f_i \leq \eta$
- Less η means more strict requirement
- Min-cost solutions are sensitive

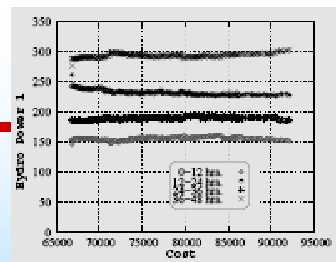


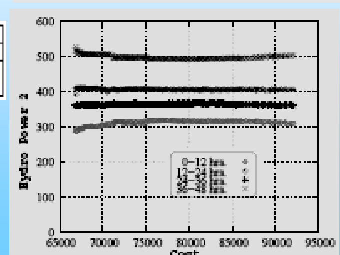
Innovizations

- Hydroelectric power generation almost constant
- Higher power demand, more generation of hydroelectric power

	0-12 Hrs.	12-24 Hrs.	24-36 Hrs.	36-48 Hrs.
Demand P_{dm} (MW)	900.00	1100.00	1000.00	1300.00
$P^h{}_{1m}$ (MW)	155.45	232.70	189.29	293.39
$P^h{}_{2m}$ (MW)	311.68	406.30	364.37	498.69

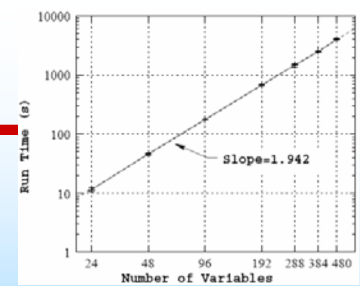
- Thermal power causes trade-off
- Other details in original study

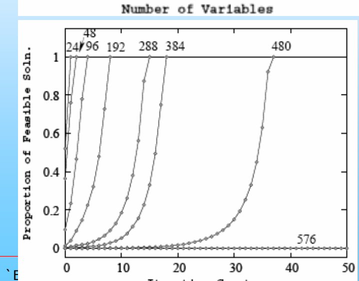




Scale-up Study

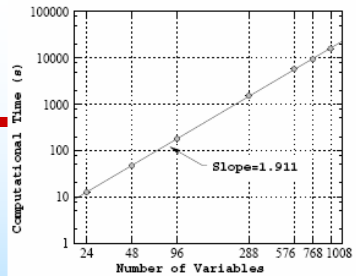
- More time steps
- *N=8n*
- Polynomial scaling up to 480 real variables
- Feasible solutions are difficult to find for higher number variables
- Need problem information through customization

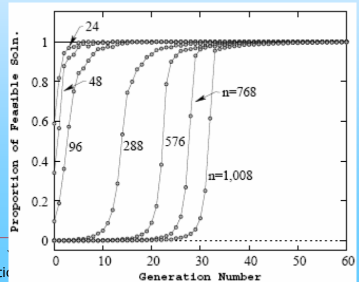




Scale-up Study (cont.)

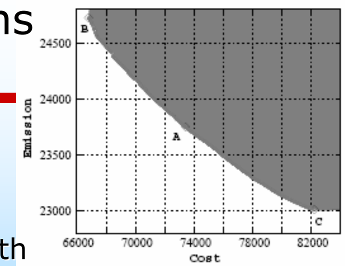
- Problem information in generating initial population
- One solution with equal Ts and Th for each time period
- Feasible solutions are easy to find
- Performance is better (1,008 real variables, *17.2 minutes vs. 12 hourly change in demand*)

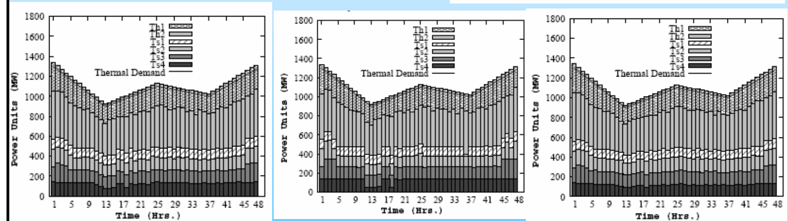




Trade-off Solutions (4 Time Steps)

- Demand is met
- Min-cost solution is abrupt and isolated
- Min-emission sol. smooth

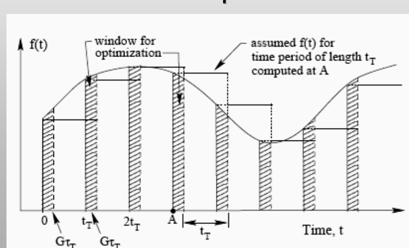


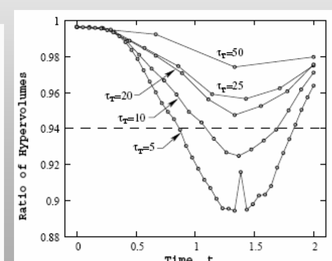


Operator Advantage

Dynamic Optimization

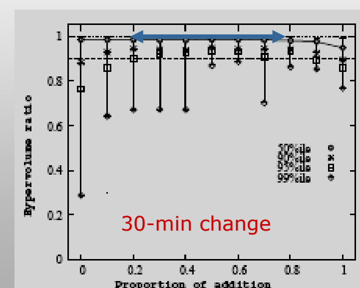
- Assume a statis in problem for a time step
- Find a critical frequency of change
- FDA2 test problem

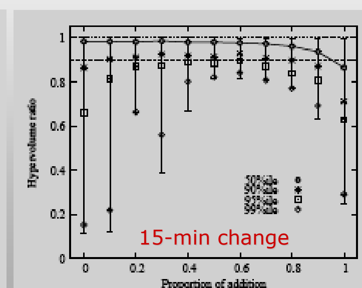




Dynamic Multi-Objective Hydro-Thermal Power Scheduling

- Addition of random or mutated points at changes
- 30-min change found satisfactory

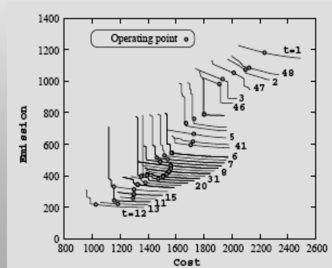


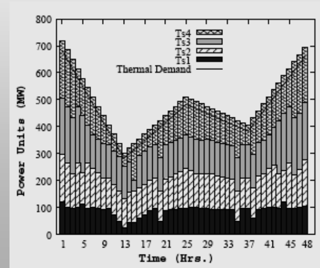


Dynamic EMO with Decision-Making

- Needs a fast decision-making
- Use an automatic procedure
  - Utility function, pseudo-weight etc.

Case	Cost	Emission
50-50%	74239.07	25314.44
100-0%	69354.73	27689.08
0-100%	87196.50	23916.09



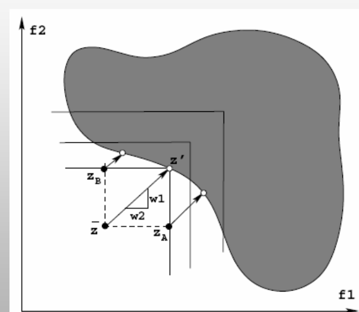


EMO for Decision-Making

- Use where multiple, repetitive applications are sought
- Use where, instead of a point, a trade-off region is sought
- Use for finding points with specific properties (nadir point, knee point, etc.)
- Use for robust, reliable or other fronts
- Use EMO for an idea of the front, then decision-making (I-MODE)

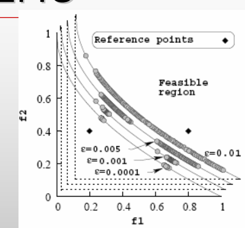
Reference Point Based EMO

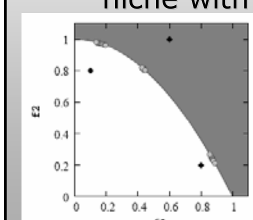
- Wierzbicki, 1980
- A P-O solution closer to a reference point
  - Multiple runs
  - Too structured
- Extend for EMO
  - Multiple reference points in one run
  - A distribution of solutions around each reference point

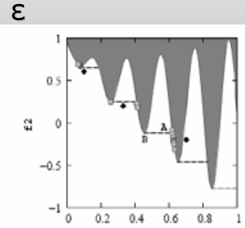


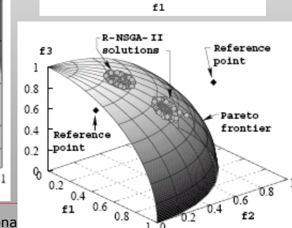
Making Decisions:
Reference Point Based EMO

- Ranking based on closeness to each reference point
- Clearing within each niche with ε

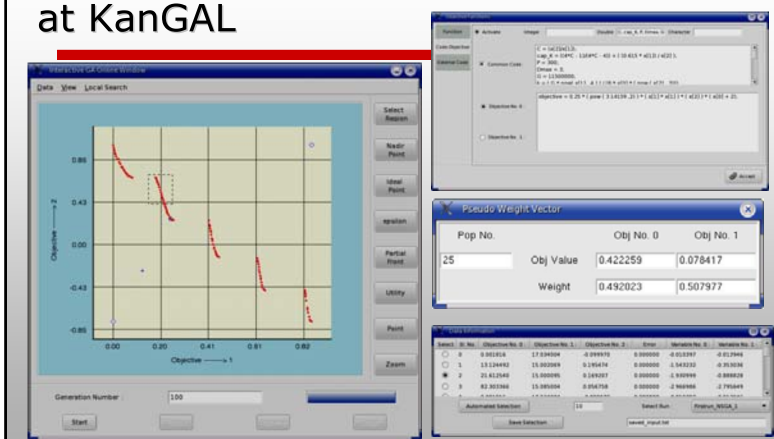




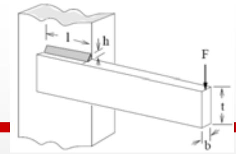


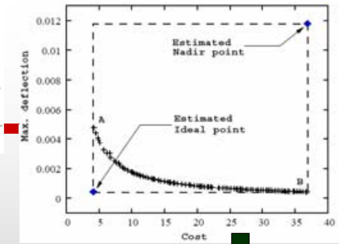


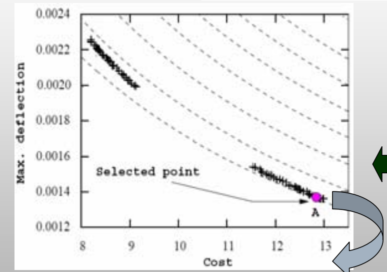
I-MODE Software Developed at KanGAL



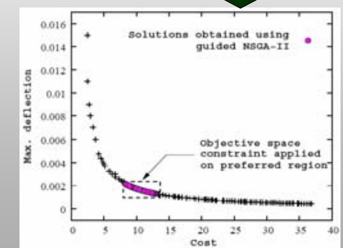
Example:







Decision Variables				Objective Values	
h (in)	l (in)	t (in)	b (in)	Cost	Max. Deflection (in)
0.917	1.009	9.856	1.672	12.838	0.001371



Summary

- Most application activities require optimization routinely
- Classical methods provide foundation
  - If applicable, good accuracy is achievable
- Evolutionary methods enable applicability to near-optimality
  - Try when classical methods fail
  - Parallel search ability
- A good optimization task through EAs and local search
- EAs for knowledge discovery -- *Innovization*

Summary (cont.)

- Seems impossible to have one algorithm for many practical problems
  - Needs a customized optimization
- A successful application requires
  - Domain-specific knowledge
  - Thorough knowledge on optimization basics and algorithms
  - Good computing background
- Record successful show-cases in a data-base; choose an suitable one for an application
  - Calls for collaborations

Thank You for Your Attention

- Acknowledgement:
  - KanGAL students, staff and collaborators
  - GM, GE, Honda R&D, STMicroelectronics
  - Governmental Research Labs

For further information:

http://www.iitk.ac.in/kangal/pub.htm
Email: deb@iitk.ac.in

Wishing you have a productive GECCO-2008!