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
Constrained Optimization Basics

- Decision variables: $x = (x_1, x_2, ..., x_n)$
- Constraints restrict some solutions to be feasible

Min. $f(x)$

s.t. $g_j(x) \geq 0$ \quad $j = 1, 2, ..., J$  
$h_k(x)=0$ \quad $k = 1, 2, ..., K$  
$x_l \leq x_i \leq x_u$ \quad $i = 1, 2, ..., n$

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

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 \( P_1 \) and \( P_2 \) using A1 and A2 for a fixed number of evaluations
  - \( P_1 = P_2 \)

- 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, Occam’s razor (simplest) etc.
  - Tomography, reconstruction, 3D from 2D images
  - Lead to multiple solutions

- Parameter optimization for optimal performance
- Scientific experiments, computer experiments

Reverse current in brain models (Johnson, 2006)
Scope of Optimization in Practice (cont.)

**Modeling** (system, process)

- Scheduling and planning (combinatorial optimization)
  - Routing & Scheduling
  - Traveling Salesperson problem

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

**Forecasting and prediction**
Scope of Optimization in Practice (cont.)

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

Scope of Optimization in Practice (cont.)

- **Machine learning**
  - Designing intelligent systems

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

Different Problem Complexities

- Mixed variables
- Multi-modal
- Robust solution
- Reliable solution
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

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

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

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
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

Advantages (cont.)

- 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

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
  - Mutation
  \[
  (x_1 x_2 \ldots x_n) \Rightarrow ?
  \]
  \[
  (y_1 y_2 \ldots y_n) \Rightarrow ?
  \]
- Simple exchanges are not adequate
Naive Recombination

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

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.345</td>
<td>3.569</td>
<td>11.142</td>
</tr>
<tr>
<td>21.142</td>
<td>5.687</td>
<td>26.345</td>
</tr>
</tbody>
</table>

- 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)

- Step 1: Choose a random number $u \in [0,1]$.
- Step 2: Calculate $\beta$:
  $$\beta = \frac{c_1 - c_2}{p_1 - p_2}$$
- Step 3: Compute two offspring:
  $$c_1 = 0.5 \left( (1 + \beta_1) p_1 + (1 - \beta_1) p_2 \right)$$
  $$c_2 = 0.5 \left( (1 - \beta_2) p_1 + (1 + \beta_2) p_2 \right)$$

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

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

\[ \bar{y} = \bar{x}_p + \sum_{i=1, i \neq p}^{\mu} \bar{w}_i \bar{D} e^{(i)} \]
Generalized Generation Gap (G3) Model (Steady-state approach)

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

- Parametric studies with \( \lambda \) and \( N \)

Three Test Problems

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

\[
\begin{align*}
F_{\text{elp}} &= \sum_{i=1}^{n} ix_i^2 \quad \text{(Ellipsoidal)} \\
F_{\text{sch}} &= \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2 \quad \text{(Schwefel)} \\
F_{\text{ros}} &= \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \quad \text{(Rosenbrock)}
\end{align*}
\]

Quasi-Newton Method

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

<table>
<thead>
<tr>
<th>Func.</th>
<th>FE</th>
<th>Best</th>
<th>Median</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_{\text{elp}} )</td>
<td>6,000</td>
<td>8.819(10^{-24})</td>
<td>9.718(10^{-24})</td>
<td>2.226(10^{-23})</td>
</tr>
<tr>
<td>( F_{\text{sch}} )</td>
<td>15,000</td>
<td>4.118(10^{-12})</td>
<td>1.021(10^{-10})</td>
<td>7.422(10^{-9})</td>
</tr>
<tr>
<td>( F_{\text{ros}} )</td>
<td>15,000</td>
<td>6.077(10^{-17})</td>
<td>4.046(10^{-10})</td>
<td>3.987</td>
</tr>
<tr>
<td>( F_{\text{elp}} )</td>
<td>8,000</td>
<td>5.994(10^{-24})</td>
<td>1.038(10^{-23})</td>
<td>2.226(10^{-23})</td>
</tr>
<tr>
<td>( F_{\text{sch}} )</td>
<td>18,000</td>
<td>4.118(10^{-12})</td>
<td>4.132(10^{-11})</td>
<td>7.422(10^{-9})</td>
</tr>
<tr>
<td>( F_{\text{ros}} )</td>
<td>26,000</td>
<td>6.077(10^{-17})</td>
<td>4.046(10^{-10})</td>
<td>3.987</td>
</tr>
</tbody>
</table>

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. $x_i$ and $v$ are recombined with $p$

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

Differential Evolution (DE)

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

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

Particle Swarm Optimization (PSO)

- Kennedy and Eberhart, 1995
- Particles fly through the search space
- Velocity dynamically adjusted
- $x_i = x_i + v_i$
- $v_i = v_i + c_1 \text{rand}((p_{i,\text{best}} - x_i) + c_2 \text{rand}((p_{g} - x_i))$
- $p_i$: 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)

<table>
<thead>
<tr>
<th>DE</th>
<th>Best</th>
<th>Median</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>9,660</td>
<td>12,033</td>
<td>20,881</td>
</tr>
<tr>
<td>G3</td>
<td>5,826</td>
<td>6,800</td>
<td>7,728</td>
</tr>
<tr>
<td>DE</td>
<td>102,000</td>
<td>119,170</td>
<td>185,590</td>
</tr>
<tr>
<td>G3</td>
<td>13,988</td>
<td>15,602</td>
<td>17,188</td>
</tr>
<tr>
<td>DE</td>
<td>243,800</td>
<td>587,920</td>
<td>942,040</td>
</tr>
<tr>
<td>G3</td>
<td>16,508</td>
<td>21,452</td>
<td>25,520</td>
</tr>
</tbody>
</table>

DE Results

Particle Swarm Optimization (PSO)
CMA-ES
(Hansen & Ostermeier, 1996)
- Selecto-mutation ES is run for n iterations
- Successful 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

<table>
<thead>
<tr>
<th>EA</th>
<th>Fe_{opt}</th>
<th>F_{ech}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
</tr>
<tr>
<td>CMA-ES</td>
<td>8,064</td>
<td>8,472</td>
</tr>
<tr>
<td>DE</td>
<td>9,660</td>
<td>12,033</td>
</tr>
<tr>
<td>G3+PCX</td>
<td>5,826</td>
<td>6,800</td>
</tr>
</tbody>
</table>

Accuracy 1X10^{-20}

Mixed-Variable Optimization:
Handling mixed type of variables
- Treat type of cross-sections, materials, etc. as decision variables
- A mixed representation:
  - (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:
  \[
  F(x) = f(x) + \sum_{j=1}^{J} R \left( g_j(x) \right)^2
  \]
- \( a \) = a if a is -ve, 0 otherwise
- Performance sensitive to penalty parameters
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 $\sum g_j(x)$  
  - (Deb, CMAME 2000)

\[
F(x) = \begin{cases} 
  f(x), & \text{if } g_j(x) \geq 0, \forall j \in J \\
  f_{\text{max}} + \sum' g_j(x), & \text{otherwise}
\end{cases}
\]

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 Problem (cont.)

- Maximum metal utilization:
  \[ \frac{1}{H} \sum_{k=1}^{K} \frac{w_k x_k}{W} \]

- Constraints:
  - Demand satisfaction: \( \sum_{i=1}^{H} x_k = r_k \) for \( k = 1, \ldots, K \)
  - Capacity constraint: \( \sum_{i=1}^{N} w_i x_i \leq W \) 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.)

Off-The-Shelf EA Results

<table>
<thead>
<tr>
<th>Number of</th>
<th>Binary-coded GAs</th>
<th></th>
<th>Real-coded GAs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>Population Size</td>
<td>Efficiency</td>
<td>Function</td>
<td>Population Size</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>96.15</td>
<td>13,800</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>300</td>
<td>98.01</td>
<td>1,42,300</td>
<td>200</td>
</tr>
<tr>
<td>300</td>
<td>1,000</td>
<td>90.11</td>
<td>14,12,400</td>
<td>700</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>N</th>
<th># Heats</th>
<th># Heats</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>211</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>204</td>
<td>200</td>
</tr>
<tr>
<td>12</td>
<td>201</td>
<td>200</td>
</tr>
<tr>
<td>16</td>
<td>201</td>
<td>200</td>
</tr>
</tbody>
</table>

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)
- $d$ is a distance measure between two solutions.
  - Phenotypic distance: $d(x_i, x_j)$, $x$: variable
  - Genotypic distance: $d(s_i, s_j)$, $s$: string
- Calculate niche count, $n_c = \sum Sh(d_{ij})$
- Shared fitness: $f'_i = f_i / n_c$
- Use proportionate selection operator

$Sh(d_{ij}) = \begin{cases} 1 - \left( \frac{d_{ij}}{\alpha} \right) \alpha & \text{if } d_{ij} \leq \alpha, \\ 0 & \text{otherwise}. \end{cases}$
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:
  $$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j$$
- $\beta$, determined by least-square regression from observed data
  - Optimize to minimize error: Find mean $\beta_i$
  - Variance of $\beta_i$ 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
  \[ f(x) = \sum_{i=1}^{M} \lambda_i y(x^{(j)}) \]
  \[ Y(x) = \sum_{k=1}^{K} \beta_k f_k(x) + Z(x) \]
- \( \lambda_i \) depends on distance of \( X \) from observed points
- Flexible, but complex
- Suited for \( k < 50 \), deterministic problems

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

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 (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 $\delta$-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

**Minimize** $(f_1^{\text{eff}}(x), f_2^{\text{eff}}(x), \ldots, f_M^{\text{eff}}(x))$

**subject to** $x \in S$,

**Type II**

**Minimize** $f(x) = (f_1(x), f_2(x), \ldots, f_M(x))$

**subject to** $\frac{\|f(x) - f(x)\|}{\|f(x)\|} \leq \eta, x \in S$.

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 $Pr(g_j(x) \geq 0) \geq \beta_j$,
$\beta_j$ is user-supplied

Statistical Procedure: Check if a solution is reliable

- PMA approach
  Minimize $G_j(U)$,
  Subject to $\|U\| = \beta_j$,

- RIA approach
  Minimize $G_j(U^*)$,
  Subject to $\|U\| \geq \beta_j^*$

Multiple Reliability Solutions: Get a better insight

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!

In Search of Common Optimality Properties

Fritz-John Necessary Condition:

Solution $\mathbf{x}^*$ satisfy

1. $\sum_{m=1}^{M} \lambda_m \nabla f_m(x^*) - \sum_{j=1}^{J} \mu_j \nabla g_j(x^*) = 0$, and
2. $\mu_j g_j(x^*) = 0$ 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

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!

Brushless DC Permanent Magnet Motor Design

- Five variables (all discrete), three constraints
- Non-convex, disconnected P-O front

Innovations:

- Connection: Y (betw. 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

Overhead Crane Maneuvering

- Minimize time of operation
- Minimize operating energy

NSGA-II finds trade-off and interesting properties

- Push early
- Lower later

E = mg(1 - \cos \theta)t

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

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

- Thermal power generation $P_{th}$ and hydroelectric power generation $P_{he}$ are variables

\[ \text{Minimize } f_1(x) = \sum_{m=1}^{N} \left[ \sum_{t=1}^{T} \left( \alpha_t (P_{th} - P_{he}) + \beta_t (P_{he} - P_{th}) + \gamma_t (P_{he} - P_{th})^2 + \delta_t \sin \left( \phi_t (P_{he} - P_{th}) \right) \right) \right] \]

\[ \text{Minimize } f_2(x) = \sum_{m=1}^{N} \left[ \sum_{t=1}^{T} \left( \alpha_t (P_{th} - P_{he}) + \beta_t (P_{he} - P_{th}) + \gamma_t (P_{he} - P_{th})^2 + \delta_t \sin \left( \phi_t (P_{he} - P_{th}) \right) \right) \right] \]

subject to

\[ \sum_{t=1}^{T} \left( \alpha_t (P_{th} - P_{he}) + \beta_t (P_{he} - P_{th}) + \gamma_t (P_{he} - P_{th})^2 + \delta_t \sin \left( \phi_t (P_{he} - P_{th}) \right) \right) \leq 0, \quad m = 1, 2, \ldots, M, \]

\[ \sum_{t=1}^{T} \left( \alpha_t (P_{th} - P_{he}) + \beta_t (P_{he} - P_{th}) + \gamma_t (P_{he} - P_{th})^2 + \delta_t \sin \left( \phi_t (P_{he} - P_{th}) \right) \right) \leq 0, \quad h = 1, 2, \ldots, N_h. \]

\[ P_{he,min} \leq P_{he} \leq P_{he,max}, \quad h = 1, 2, \ldots, N_h, \quad m = 1, 2, \ldots, M. \]

\[ P_{th,min} \leq P_{th} \leq P_{th,max}, \quad s = 1, 2, \ldots, N_s, \quad m = 1, 2, \ldots, M. \]

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

Innovized Principles: An Optimal Operating Chart

- Four thermal, two hydroelectric, M time steps

\[ \left( P_{th}, P_{he} \right) \text{ subject to } (P_{th}, P_{he}) \leq \text{bounds} \]

Two sets of constraints:

- Handle water availability constraint first
- Quadratic constraint, repair to find $P_{he}$

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

GA Coding and Constraint Handling

- Four thermal, two hydroelectric, M time steps

\[ \left( P_{th}, P_{he} \right) \text{ subject to } (P_{th}, P_{he}) \leq \text{bounds} \]

Two sets of constraints:

- Handle water availability constraint first
- Quadratic constraint, repair to find $P_{he}$

- If not within bounds, penalize
- Similarly, handle other constraint set
NSGA-II Simulation and Verifications
- Single-objective opt. for extreme solns.
- Intermediate c-constraint optimizations
- Better than a SA-approach with naive penalty
- Min-cost soln. difficult to optimize

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

NSGA-II Simulation and Verifications

Bounds Too Tight (250 & 500 MW)
- Increase upper bounds to 350, 600 MW
- Better behaved P-O frontier

Verify Obtained Solutions Through KKT Conditions

\[ \lambda_1 \nabla f_1(x) + \lambda_2 \nabla f_2(x) + \sum_{i=1}^{M} \nu_i \nabla h_i^1(x) + \sum_{j=1}^{N_c} \nu_j \nabla h_j^2(x) = 0. \]

\[ f_j \text{ broken into two parts, } s_i \text{ in } [-1,1] \]

KKT conditions reduce to following: (Ay=b)

Verify Obtained Solutions Through KKT Conditions

Error Minimization and KKT Results
- Solutions are close to KKT points, if error is close to zero, \( \lambda \geq 0, s_i \text{ in } [-1,1] \)

Error Minimization and KKT Results
Solutions in Earlier Generations are Not Close to KKT Points

- $\lambda < 0$, error high

<table>
<thead>
<tr>
<th>Solution</th>
<th>$(f_1, f_2)$</th>
<th>$t$ near kink</th>
<th>$s$</th>
<th>$\delta(s)$</th>
<th>$\lambda_1/\lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(8.2940.02, 3.6760.00)</td>
<td>3</td>
<td>0</td>
<td>0.0015</td>
<td>0.006</td>
</tr>
<tr>
<td>2</td>
<td>(8.2937.15, 2.5924.30)</td>
<td>1</td>
<td>12</td>
<td>0.9469</td>
<td>0.051</td>
</tr>
<tr>
<td>3</td>
<td>(31521.44, 26028.55)</td>
<td>4</td>
<td>-11</td>
<td>0.3512</td>
<td>-0.078</td>
</tr>
</tbody>
</table>

Robust Frontier

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

Innovations

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

---

Dynamic Optimization

- Assume a static 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.

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>Emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50%</td>
<td>74239.67</td>
<td>2514.44</td>
</tr>
<tr>
<td>10-50%</td>
<td>66354.73</td>
<td>27689.08</td>
</tr>
<tr>
<td>0-100%</td>
<td>87196.50</td>
<td>23916.00</td>
</tr>
</tbody>
</table>

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 $\varepsilon$
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 database; choose an suitable one for an application
- Calls for collaborations

Example:

![Example Graph]

Design Variables | Objective Values | Cost | Max. Reflection
--- | --- | --- | ---
0.997 | 1.059 | 9.556 | 1.072 | 12.839 | 0.001571
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!