Multidisciplinary System Design Optimization (MSDO)

Optimization Method Selection

Recitation 5

Andrew March

Today's Topics

- Review optimization algorithms
- Algorithm Selection
- Questions

Analytical Methods

- Gradient Based:
  - Steepest descent
  - Conjugate Gradient
  - Newton's Method
  - Quasi-Newton

- Direct Search:
  - Compass search
  - Nelder-Mead Simplex

- Note: The gradient methods have a constrained equivalent.
  - Steepest Descent/CG: Use projection
  - Newton/Quasi-Newton: SQP
  - Direct search typically uses barrier or penalty methods

Gradient Methods

- Compute descent direction, $\mathbf{d}_k$
- Compute step length α_k
- Take step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- Repeat until $\alpha_k \mathbf{d}_k \leq \varepsilon$

Steepest Descent

- Compute descent direction, $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$

- Compute step length, α_k
  - Exactly: $\alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
  - Inexactly: any α_k such that for a c_1, c_2 in $(0<c_1<c_2<1)$

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

- Take step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$

- Repeat until $\alpha_k \mathbf{d}_k \leq \varepsilon$

Conjugate Gradient

- Compute descent direction, $\mathbf{d}_k = -\nabla f(\mathbf{x}_k) + \beta_k \mathbf{d}_{k-1}$

$$\beta_k = \frac{\nabla f(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_{k-1})^T \nabla f(\mathbf{x}_{k-1})} \quad \text{or} \quad \beta_k = \frac{\nabla f(\mathbf{x}_k)^T (\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}))}{\nabla f(\mathbf{x}_{k-1})^T \nabla f(\mathbf{x}_{k-1})}$$

- Compute step length, α_k
  - Exactly: $\alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
  - Inexactly: any α_k such that for a c_1, c_2 in $(0<c_1<c_2<1)$

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \le f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \ge c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

- Take step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$

- Repeat until $\alpha_k \mathbf{d}_k \le \varepsilon$

Newton's Method

- Compute descent direction, $\mathbf{d}_k = -H^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$
- Compute step length, α_k
  - Try: $\alpha_k=1$, decrease? If not:
    - Exactly: $\alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha\mathbf{d}_k)$
    - Inexactly: any α_k such that for a c_1,c_2 in $(0<c_1<c_2<1)$

$$f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1\alpha_k\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$$
$$\nabla f(\mathbf{x}_k + \alpha_k\mathbf{d}_k)^T\mathbf{d}_k \geq c_2\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$$

    - Trust-region: $\alpha_k\mathbf{d}_k \leq \Delta_k$
- Take step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$
- Repeat until $\alpha_k\mathbf{d}_k \leq \varepsilon$

Quasi-Newton

- Compute descent direction, $\mathbf{d}_k = -B^{-1}(\mathbf{x}_k)\nabla f(\mathbf{x}_k)$

$$B(\mathbf{x}_k) \approx H(\mathbf{x}_k); \quad B(\mathbf{x}_k) \succ 0$$

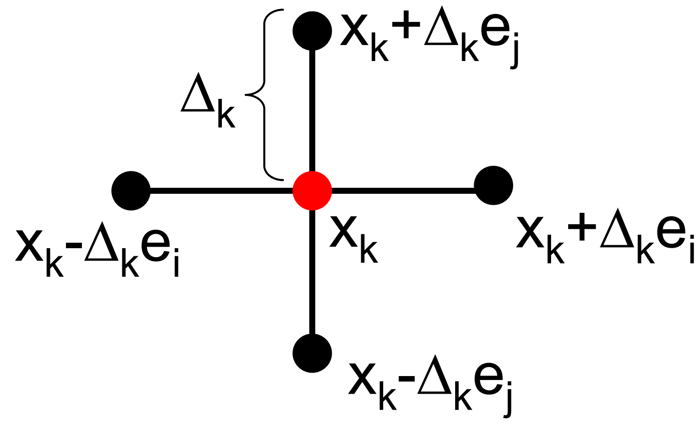
- Compute step length, α_k
  - Try: $\alpha_k=1$, decrease? If not:
    - Exactly: $\alpha_k = \arg\min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
    - Inexactly: any α_k such that for a c_1,c_2 in $(0<c_1<c_2<1)$

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \le f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \ge c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

- Take step: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
- Repeat until $\alpha_k \mathbf{d}_k \le \varepsilon$

Direct Search, Compass Search



- Evaluate $f\left(\mathbf{x}_k \pm \Delta_k \mathbf{e}_i\right), \quad \forall i$
- If $f\left(\mathbf{x}_k \pm \Delta_k \mathbf{e}_i\right) < f\left(\mathbf{x}_k\right)$
  - Move to minimum of: $f\left(\mathbf{x}_k \pm \Delta_k \mathbf{e}_i\right), \quad \forall i$
- Else
  - $\Delta_{k+1} = \frac{1}{2}\Delta_k$

Direct Search, Nelder-Mead

Generate n+1 points in $\Re^n$, $\{x_1,\ldots,x_{n+1}\}$

Iterate:

- $\mathbf{x}_l = \arg\min_{\mathbf{x}_i} f(\mathbf{x})$
- $\mathbf{x}_h = \arg\max_{\mathbf{x}_i} f(\mathbf{x})$
- $\bar{\mathbf{x}} = \text{centroid}\{\mathbf{x}_1,\ldots,\mathbf{x}_{n+1}\}$
- Reflect (α>0): $\mathbf{x}_r = (1+\alpha)\bar{\mathbf{x}} - \alpha\mathbf{x}_h$
- $\text{if}\,(f(\mathbf{x}_l) < f(\mathbf{x}_r) \text{ and } f(\mathbf{x}_r) < f(\mathbf{x}_h)),\, \mathbf{x}_h = \mathbf{x}_r, \text{ return}$
- $\text{if}\,(f(\mathbf{x}_r) < f(\mathbf{x}_l))$, Expand ($\gamma$>1): $\mathbf{x}_e = \gamma\mathbf{x}_r + (1-\gamma)\bar{\mathbf{x}}$
- $\quad \text{if}\,(f(\mathbf{x}_e) < f(\mathbf{x}_l)),\, \mathbf{x}_h = \mathbf{x}_e, \text{ return}$
- $\quad \textit{else},\, \mathbf{x}_h = \mathbf{x}_r, \text{ return}$
- $\text{if}\,(f(\mathbf{x}_r) > f(\mathbf{x}_h))$, Contract (0<$\beta$<1): $\mathbf{x}_c = \beta\mathbf{x}_h + (1-\beta)\bar{\mathbf{x}}$
- $\quad \text{if}\,(f(\mathbf{x}_c) \le \min\{f(\mathbf{x}_h), f(\mathbf{x}_r)\}),\, \mathbf{x}_h = \mathbf{x}_c, \text{ return}$
- $\quad \text{else},\, \mathbf{x}_i = (\mathbf{x}_i + \mathbf{x}_l)/2,\ \forall i$

J. A. Nelder and R. A. Mead, *A simplex method for function minimization*, Computer Journal, Vol. 7, pp 308-313, 1965.

Heuristic Methods

- Simulated Annealing
- Genetic Algorithms
- Particle Swarm Optimization (next lecture)
- Tabu Search (next lecture)
- Efficient Global Optimization

Simulated Annealing

- **Terminology:**
  - X (or R or Γ) = Design Vector (i.e. Design, Architecture, Configuration)
  - E = System Energy (i.e. Objective Function Value)
  - T = System Temperature
  - Δ = Difference in System Energy Between Two Design Vectors

- **The Simulated Annealing Algorithm**

  1) Choose a random X_i, select the initial system temperature, and specify the cooling (i.e. annealing) schedule

  2) Evaluate $E(X_i)$ using a simulation model

  3) Perturb X_i to obtain a neighboring Design Vector (X_{i+1})

  4) Evaluate $E(X_{i+1})$ using a simulation model

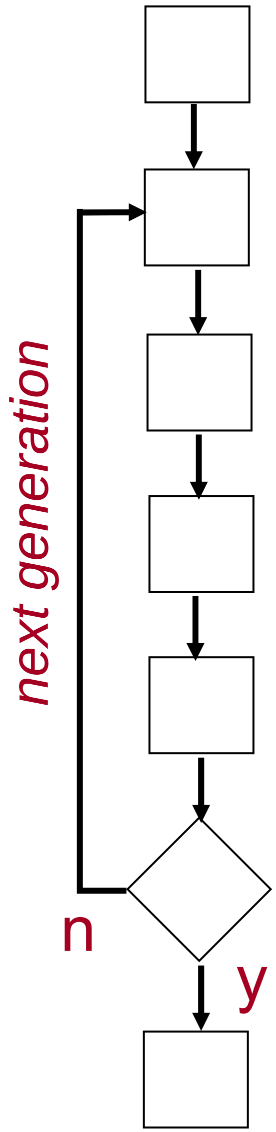
  5) If $E(X_{i+1}) < E(X_i)$, X_{i+1} is the new current solution

  6) If $E(X_{i+1}) > E(X_i)$, then accept X_{i+1} as the new current solution with a probability $e^{(-\Delta/T)}$ where $\Delta = E(X_{i+1}) - E(X_i)$.

  7) Reduce the system temperature according to the cooling schedule.

  8) Terminate the algorithm.

Genetic Algorithm



Initialize Population (initialization)

Select individual for mating (selection)

Mate individuals and produce children (crossover)

Mutate children (mutation)

Insert children into population (insertion)

Are stopping criteria satisfied ?

Finish

Ref: Goldberg (1989)

Particle Swarm

- Birds go in a somewhat random direction, but also somewhat follow a swarm
- Keep checking for “better” locations
  - Generally continuous parameters only, but there are discrete formulations.

Tabu Search

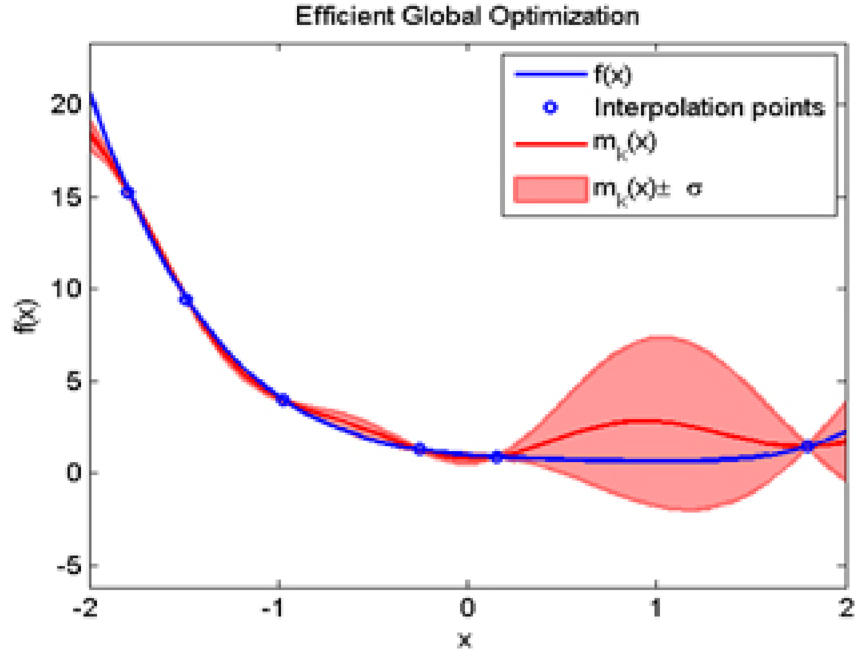
- Keep a list of places you've visited
- Don't return, keep finding new places

Efficient Global Optimization

- Started by Jones 1998
- Based on probability theory
  - Assumes:

$$f(\mathbf{x}) \approx \beta^T \mathbf{x} + N\left(\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right)$$

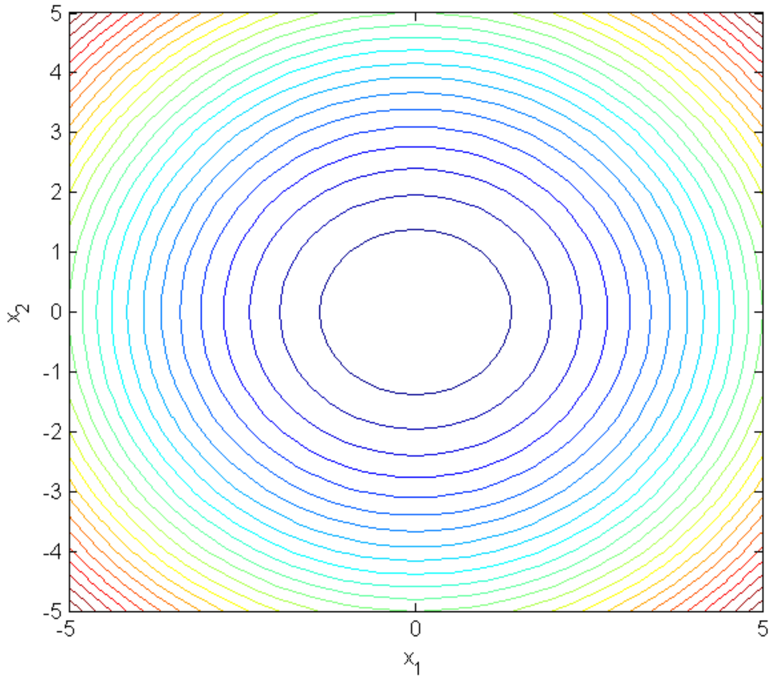
- $\beta^T \mathbf{x}$, true behavior, regression
- $N\left(\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right)$, error from true behavior is normally distributed, with mean $\mu(\mathbf{x})$, and variance $\sigma^2(\mathbf{x})$



- Estimate function values with a Kriging model (radial basis functions)
  - Predicts mean and variance
  - Probabilistic way to find optima
- Evaluate function at "maximum expected improvement location(s)" and update model

What's good algorithm?

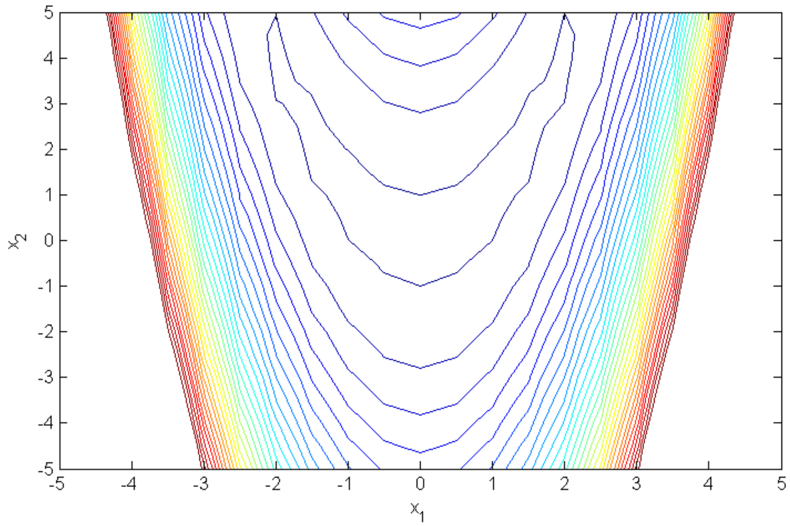
Objective Contours:



- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

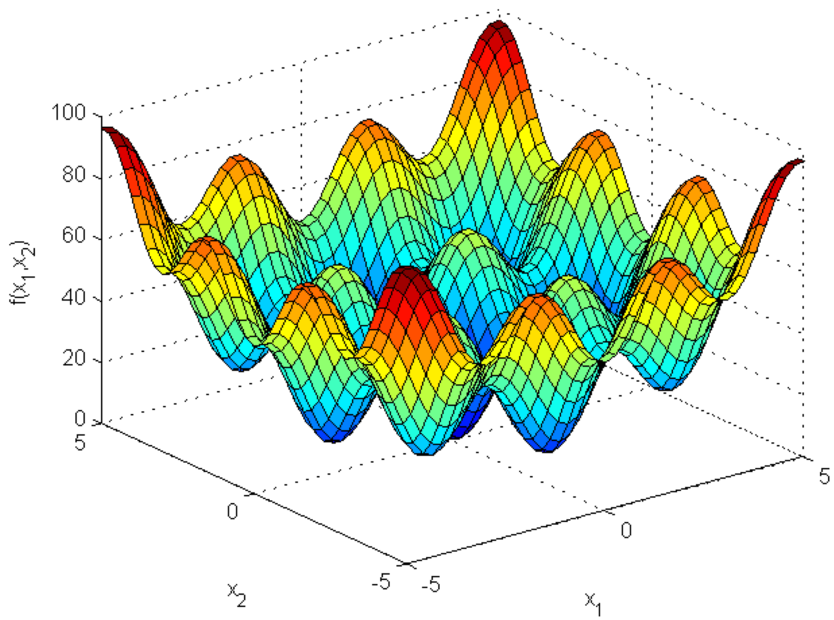
$$f(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$$



- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

Objective Function:



a) Find quick improvement?

b) Find global optima?

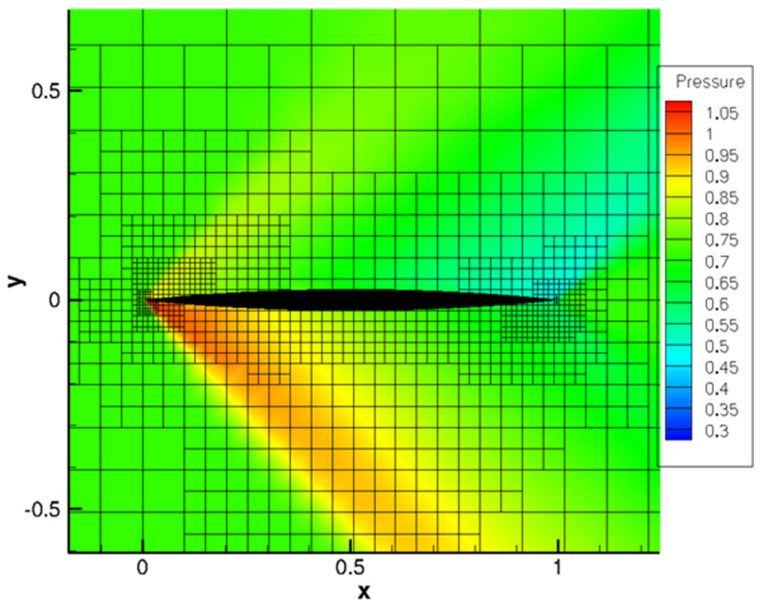
- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

- $x_1=\{1,2,3,4\}$
- $x_2 \in \Re$
- $\min f(x_1,x_2)$

- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?



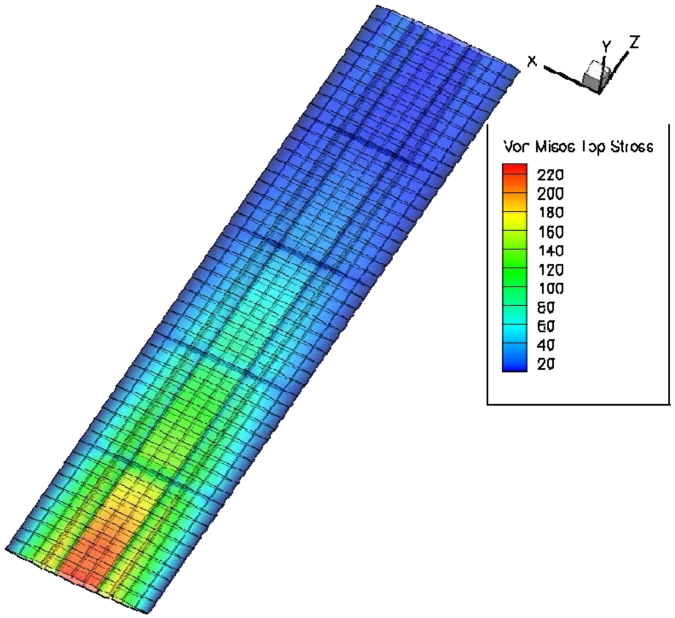
- Airfoil design with CFD
  - Run-time~3 hours

a) Without an adjoint solution?

b) With an adjoint solution?

- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

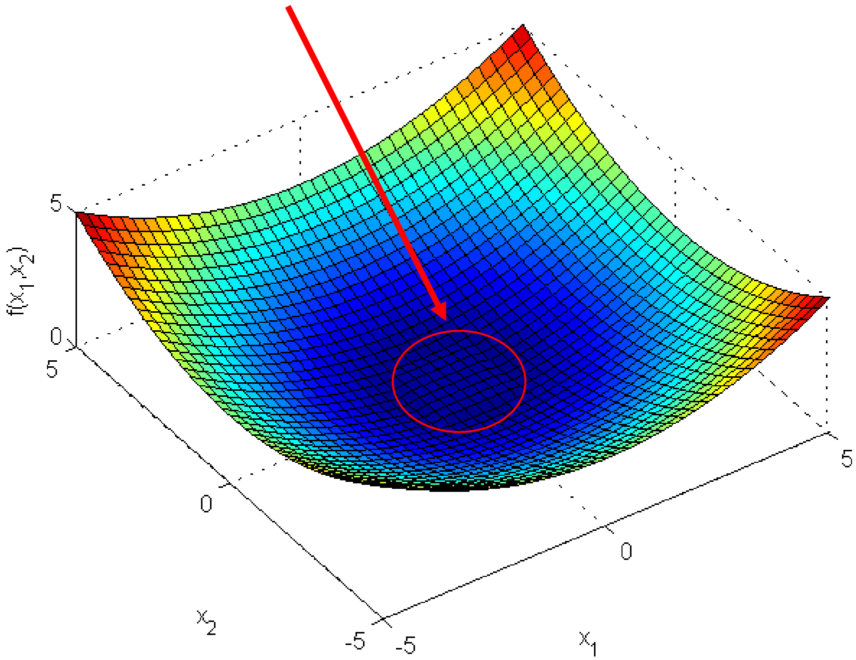


- Minimize weight
  - s.t. stress<σ_{max}
- Natran output
  - Stress=3.500x10^4
  - (finite precision)

- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

- Flat section:



- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

min $c^T\mathbf{x}$

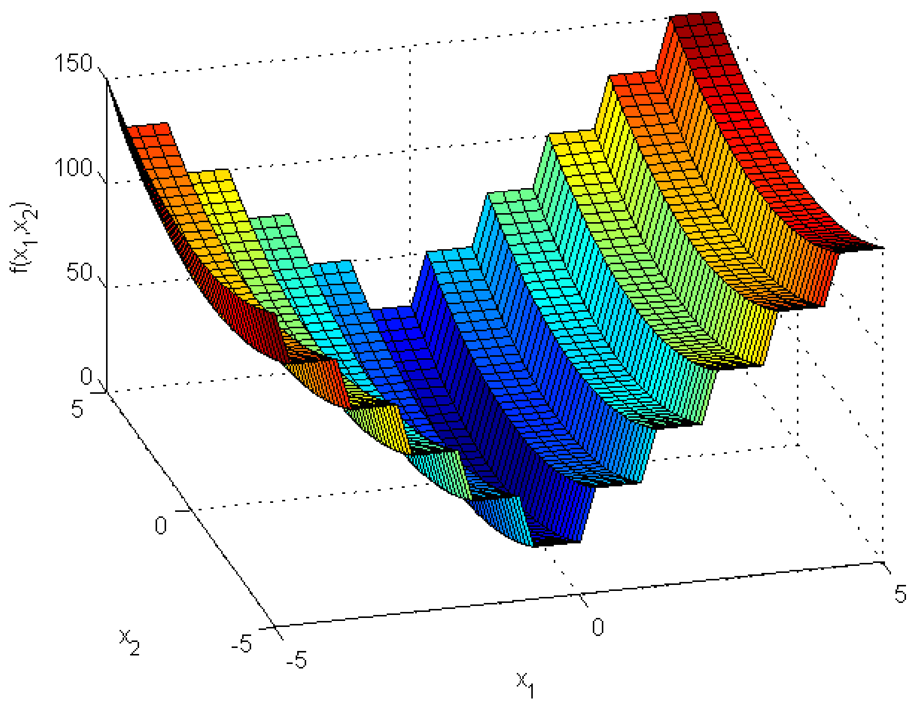
s.t. $A\mathbf{x}=b$

- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

16.888
ESD.77

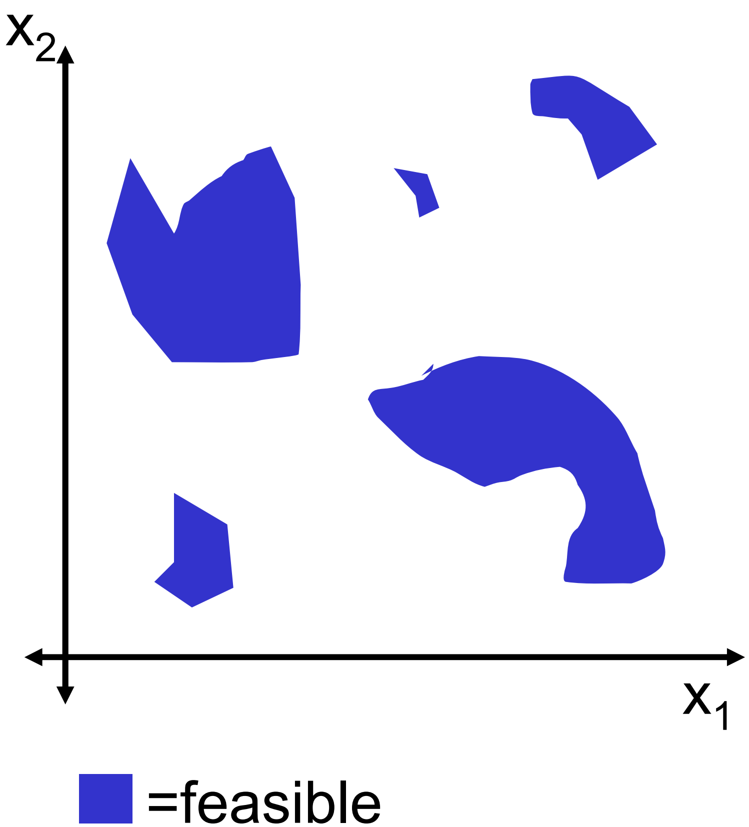
Nonsmooth objective:



- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

What's good algorithm?

Islands of feasibility:



- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

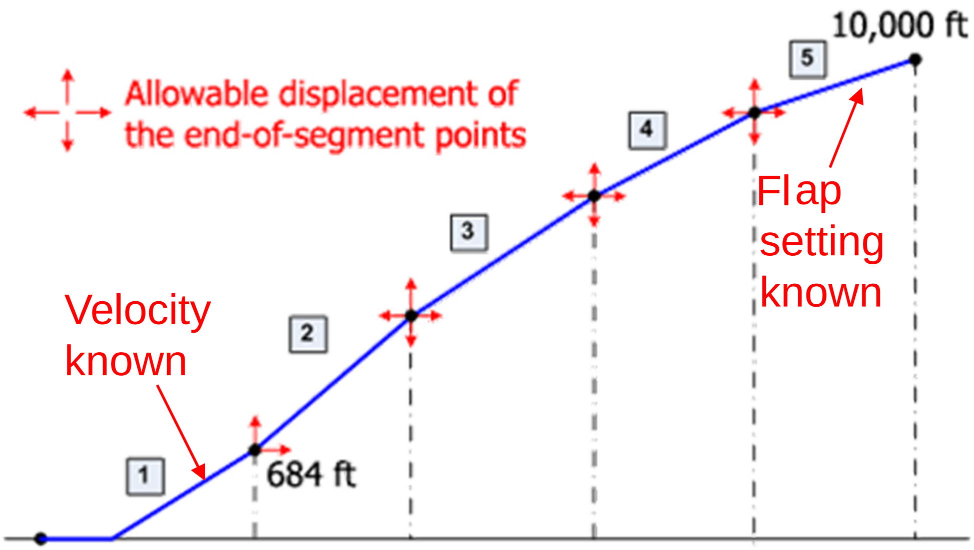
What's good algorithm?

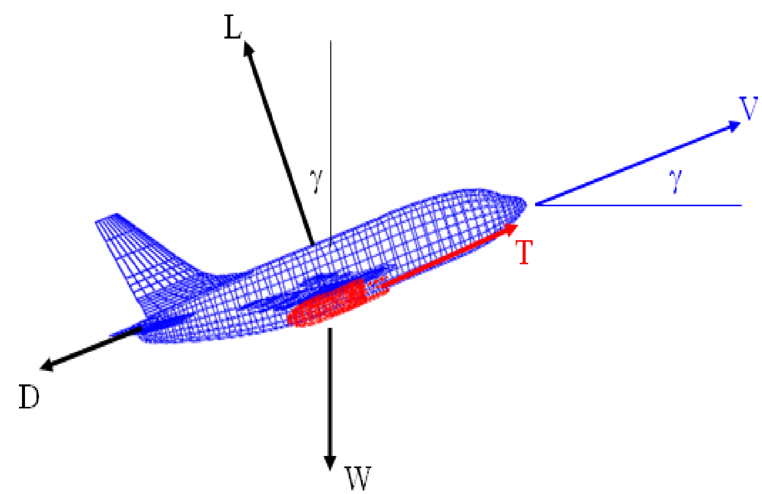
- Problem aspects:
  - Islands of feasibility
  - Many local minima
  - Mixed discrete/continuous variables
  - Many design variable scales ($10^{-1} \rightarrow 10^{4}$)
  - Long function evaluation time (~2 minutes)

- Steepest descent
- Conjugate gradient
- Newton's Method
- Quasi-Newton
- SQP
- Compass-Search
- Nelder-Mead Simplex
- SA
- GA
- Tabu
- PSO
- EGO

Example: Operational Design Space

- Objectives
  - Time to Climb, Fuel Burn, Noise, Operating Cost
- Parameters
  - Flap setting
  - Throttle setting
  - Velocity
  - Transition Altitude
  - Climb gradient*
  - **18 Total**
- Constraints:
  - Regulations
    - No pilot input below 684 ft
    - Initial climb at V_2+15kts
  - Flap settings
  - Velocity
    - Min: stall
    - Max: max q
  - Throttle
    - Min: engine idle or positive rate of climb
    - Max: full power





Example:Design Space Exploration Methods

- Exploration Challenges
  – Islands of feasibility
  – Many local minima
  – Mixed discrete/continuous variables
  – Many design variable scales ($10^{-1} \rightarrow 10^{4}$)
  – Long function evaluation time (~2 minutes with noise)

- Sequential Quadratic Programming [Climb time: 312 s]
  – Stuck at local minima
  – Can't handle discrete integers
- Direct Search (Nelder-Mead) [Climb time: 319 s]
  – Similar problems as SQP, but worse results
- Particle Swarming Optimization [Climb time: 319 s]
  – Slow running (8-12 hours), optimum not as good as Genetic Algorithm
- Genetic Algorithm [Climb time: 308 s]
  – No issues with any of the challenges of this problem.
  – No convergence guarantee and SLOW! Run-time ~24 hours.
  – But, best result.

Summary

- You have a large algorithm toolbox.
- You can often tell by inspection what algorithm might work well.
- Always take advantage of aspects of your problem that will speed convergence.

ESD.77 / 16.888 Multidisciplinary System Design Optimization
Spring 2010