A Short Tour of Derivative-Free Optimization

Juan Meza
Applied Mathematics
UC Merced

Outline

- Motivation and Example Problems
- Taxonomy of Derivative Free Optimization
- Nelder-Mead Simplex
- Genetic Algorithms/Simulated Annealing
- Generating Set Search & Pattern Search
- Future directions

General Optimization Problem

$\min$	$f(x), x \in \mathbb{R}^n$	Objective function
	$h_i(x) = 0$	Equality constraints
	$g_j(x) \geq 0$	Inequality constraints

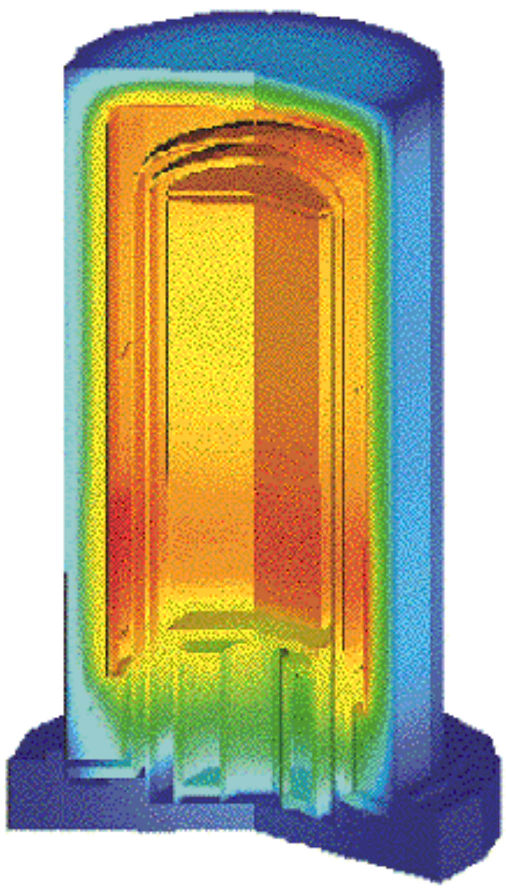
Some standard assumptions

- Objective function has infinite (machine) precision
- Objective function is smooth
  - First and second derivatives available
  - Both derivatives are also "nice"
- Constraints are linearly independent and smooth
- Objective and constraint functions cheap to evaluate

General Philosophy

1. Build an approximate model (usually quadratic) of the nonlinear objective function
2. Solve the model for its minimum
3. See how well you did and either accept the answer or throw it away
4. Repeat until you run out of time/money/patience

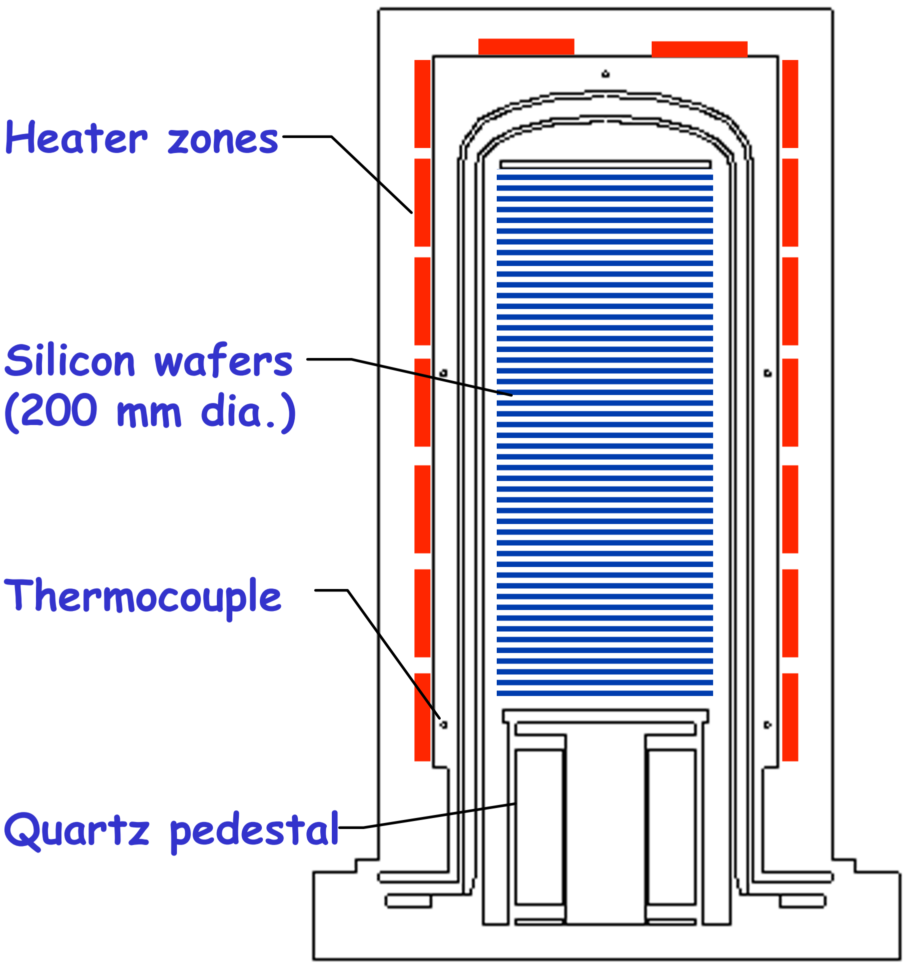
Optimizing the performance of a LPCVD furnace



Temperature fields in a vertical, stacked-wafer, low-pressure, chemical-vapor-deposition furnace

- The goal is to find heater powers that yield optimal uniform temperature
- Temperature uniformity is critical between wafers and across a wafer
- Computing temperatures involves solving a heat transfer problem with radiation
- Two-point boundary value problem solved by finite differences

Computing the temperature requires the solution of a nonlinear PDE



- Independently controlled heater zones regulate temperature
- Adjusting tolerances in the PDE solution trades off noise with CPU time
- Larger tolerances lead to less accurate PDE solutions; less time per function evaluation

The goal is to find heater powers that yield optimal uniform temperature

$$\min \quad F(p) = \sum_{i=1}^{N} (T_i(p) - T^*)^2,$$

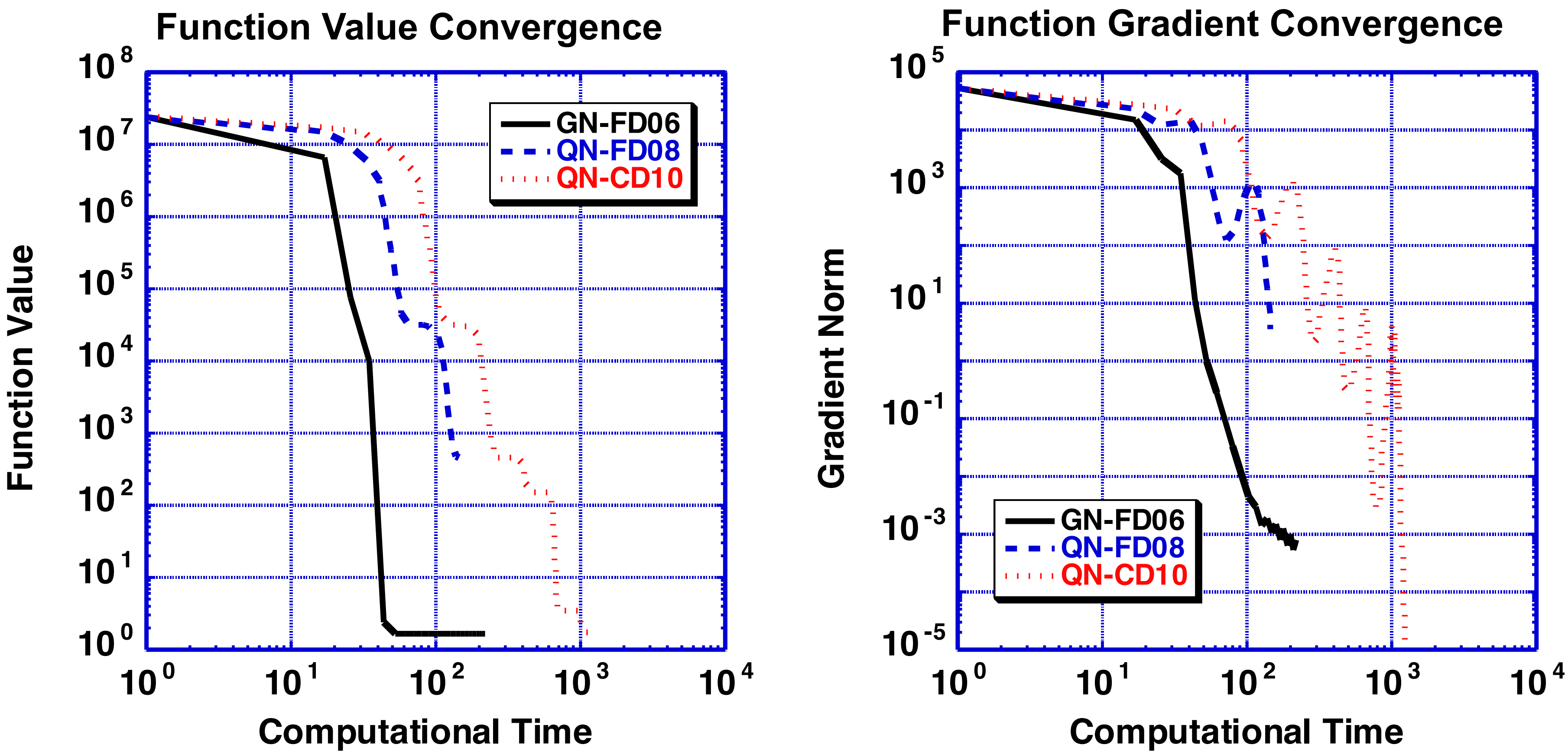
$$\begin{aligned} p \quad & \text{is the vector of heater powers,} \\ T_i(p) \quad & \text{is the temperature at discretization point } i \\ T^* \quad & \text{is the target temperature, and} \\ N \quad & \text{is the total number of discretization points} \end{aligned}$$

Some observations

- Nice simple objective function – quadratic
- Box constraints
- Easy problem to set up
- Analytic gradients are not available but the number of heaters is small so we can use finite-difference gradients

What could possibly go wrong?

Finite-Difference Gradients



- 7 Zone furnace configuration
- Quasi-Newton method exhibits “stair-stepping”

Lessons learned

- Objective function didn't have infinite machine precision
  - Sometimes true, but many simulation-based optimization problems can behave as if function was noisy
- Objective function is smooth
  - Probably differentiable, but how do you prove it?
  - What do you do if you're not given derivative information
- Constraints are linearly independent and smooth
  - Users can sometimes over specify or incorrectly guess constraints
- Objective function not cheap (PDE solve)

Derivative-Free Optimization

- Realization that simulation-based optimization problems require other methods
- Long history of these types of methods
- Fell out of favor with rise of Newton-type methods and computers

One Taxonomy

- Estimation
  - Finite Difference
  - Implicit Filtering (Mifflin, Kelley)
- Model-Based
  - Powell
  - Conn, Scheinberg, Toint
- Physics/Bio Inspired
  - Simulated Annealing
  - Genetic Algorithms
- Direct Search
  - Nelder-Mead Simplex Method
  - Pattern Search
    - Box
    - Hooke & Jeeves
    - Torczon
  - Similar Methods
    - Wenci
    - Lucidi & Sciandrone
    - García-Palomares and Rodríguez

One Taxonomy

- Estimation
  - Finite Difference
  - Implicit Filtering (Mifflin, Kelley)
- Model-Based
  - Powell
  - Conn, Scheinberg, Toint
- Physics/Bio Inspired
  - Simulated Annealing
  - Genetic Algorithms
- Direct Search
  - Nelder-Mead Simplex Method
  - Pattern Search
    - Box
    - Hooke & Jeeves
    - Torczon
  - Similar Methods
    - Wenci
    - Lucidi & Sciandrone
    - García-Palomares and Rodríguez

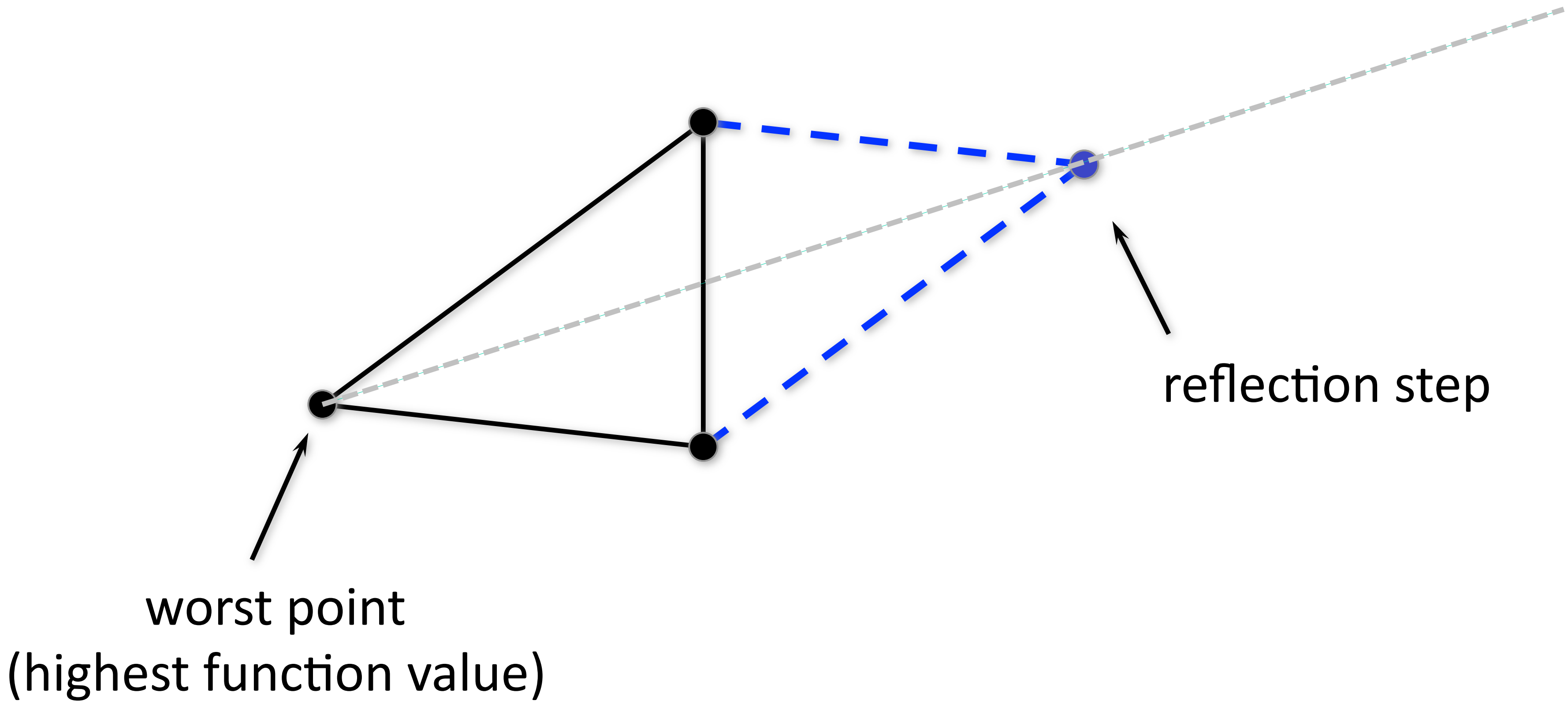
Early History (1960's)

- Nelder-Mead Simplex (1965)
- One of earliest examples of a direct search method
- Huge success, especially in engineering
- Possibly the most highly cited optimization method

Nelder-Mead Simplex

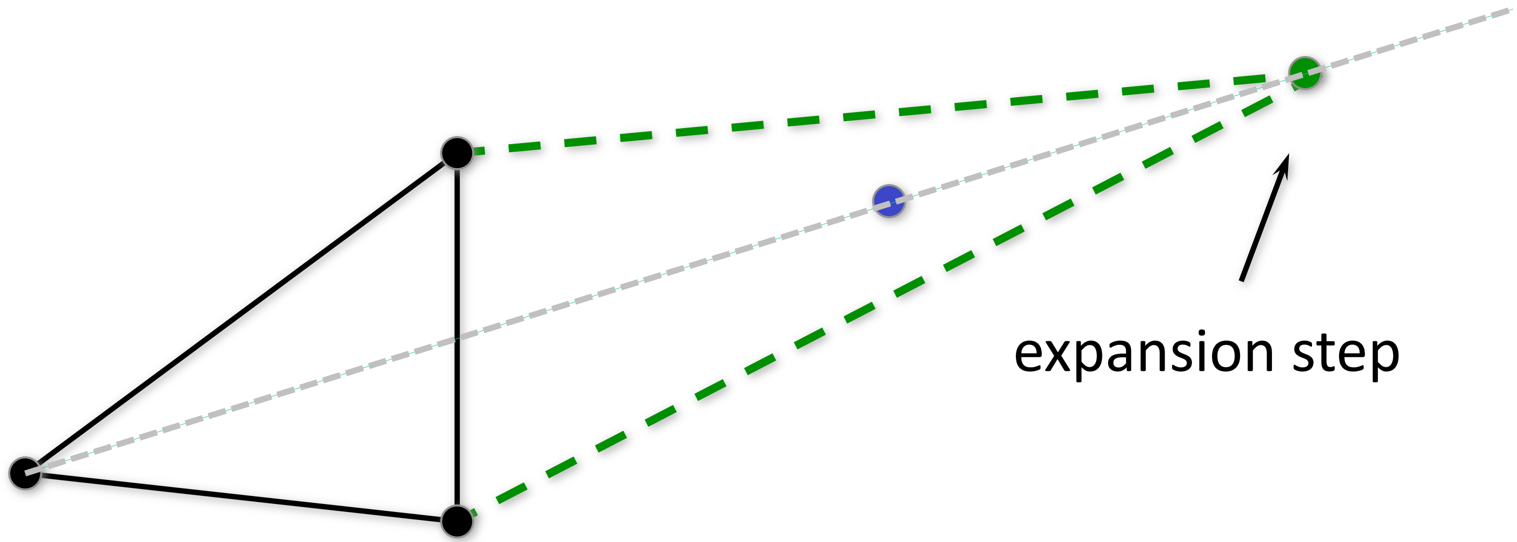
- Start with a simplex (polytope in n+1 dimensions)
- Compute function value at vertices and order the function values
- 4 Basic Steps
  1. Reflect about the centroid
  2. Expansion of simplex if really good
  3. Contract if reflection didn't work
  4. Shrink the simplex if all else fails

Nelder-Mead Reflection Step



Nelder-Mead Expansion Step

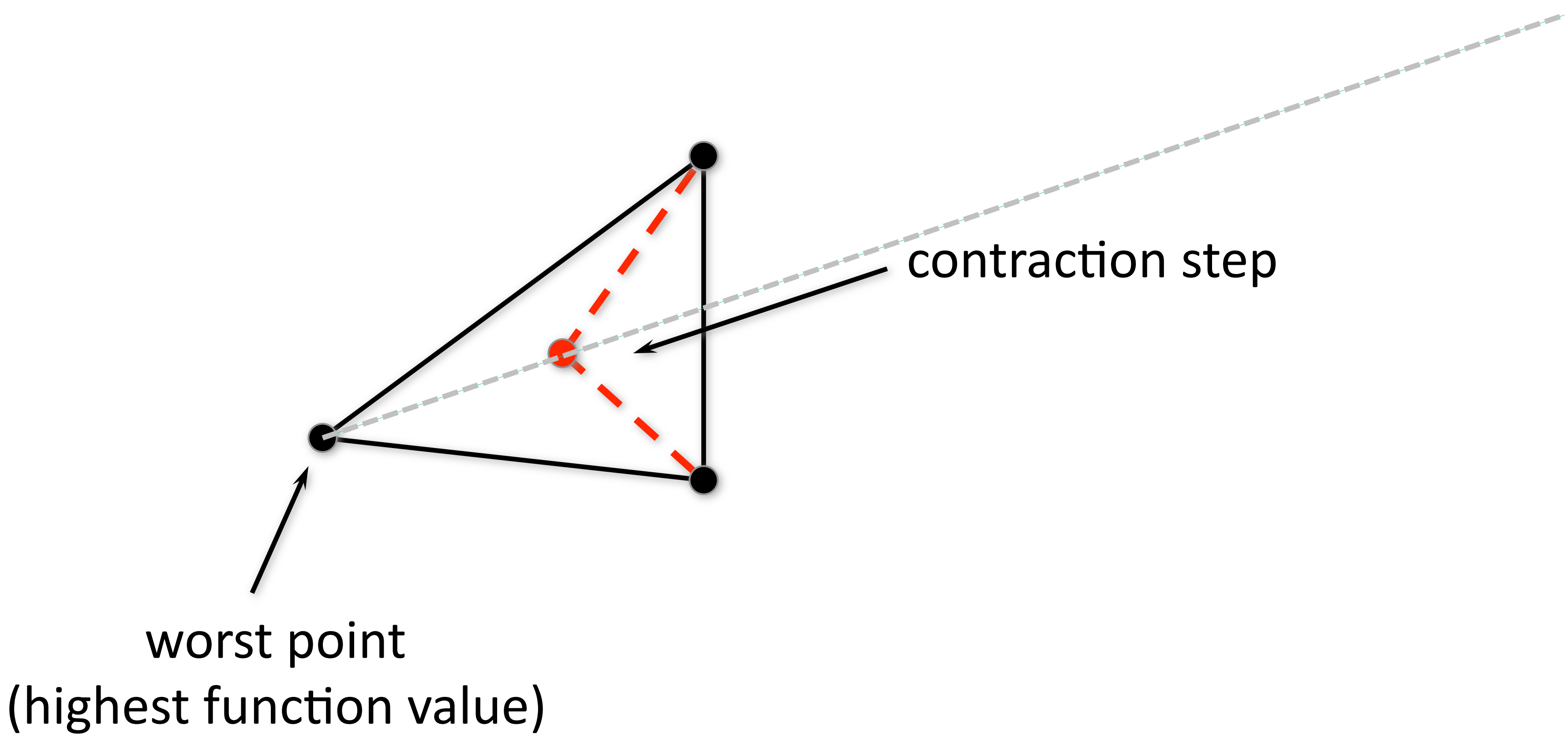
- If reflection resulted in best (lowest) value so far, try an expansion



- else, if reflection helped at all, keep it

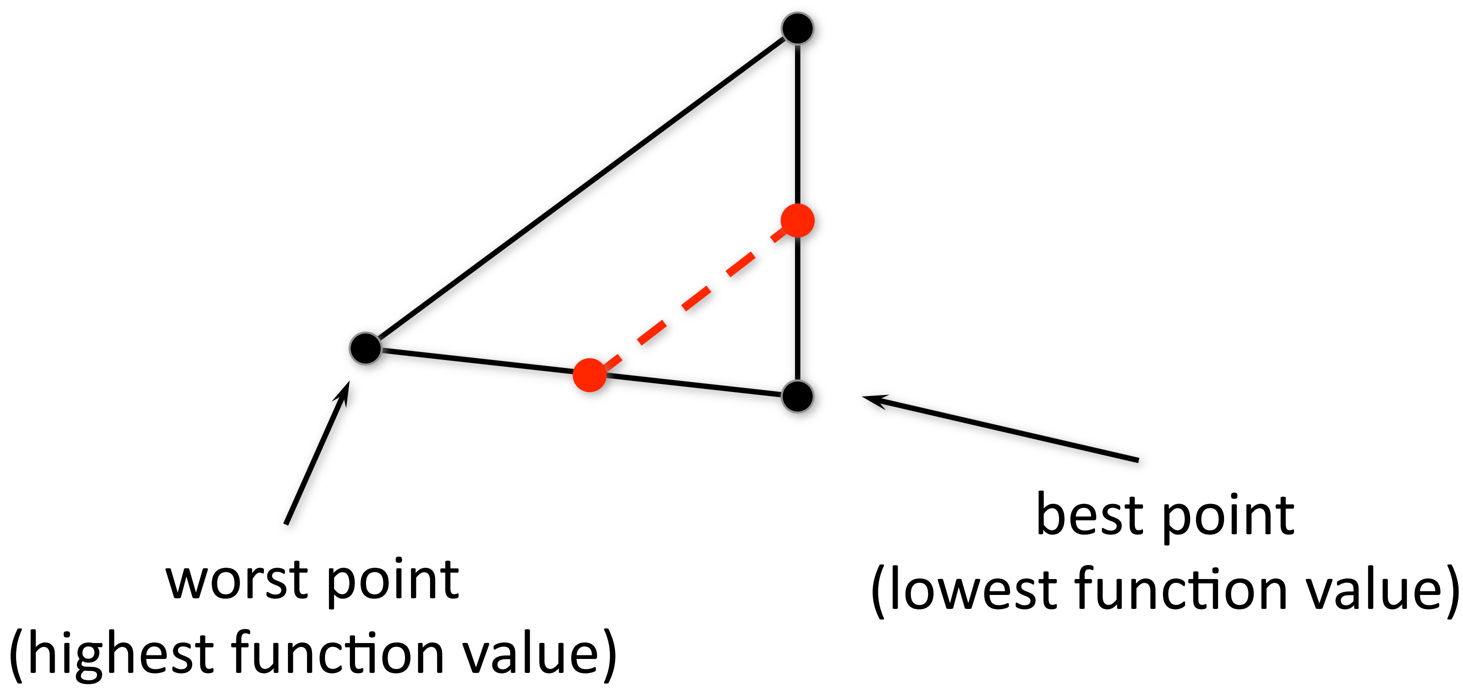
Nelder-Mead Contraction Step

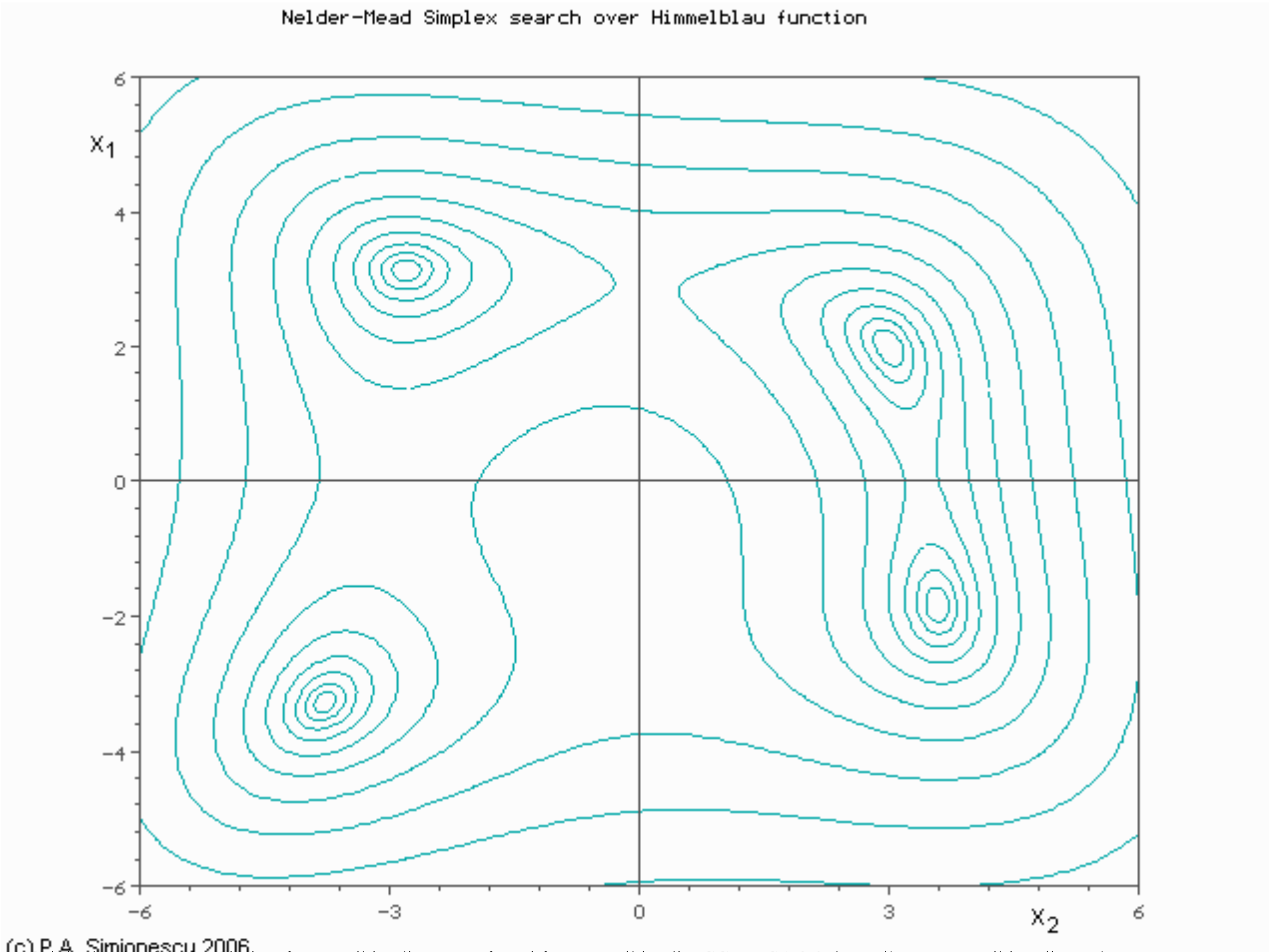
- If reflection didn't help try a contraction



Nelder-Mead Shrink Step

- If all else fails shrink the simplex around the best point




Nelder-Mead Simplex search over Himmelblau function
X1
X2
6
4
2
0
−2
−4
−6
−6
−3
0
3
6
(c) P.A. Simionescu 2006

Genetic Algorithms

- Based on evolutionary principles
- Can be used for discrete variable problems
- Random element to search
- Claim to be good for global optimization problems

Genetic Algorithms

1. **Encode** the “problem” in a binary string
2. Randomly generate a “population”
3. Calculate fitness of each member of the population
4. **Select** pairs of parent strings based on fitness
5. Apply **crossover** and **mutation** to generate a new population of offspring
6. Repeat step 3 to 5 until optimal

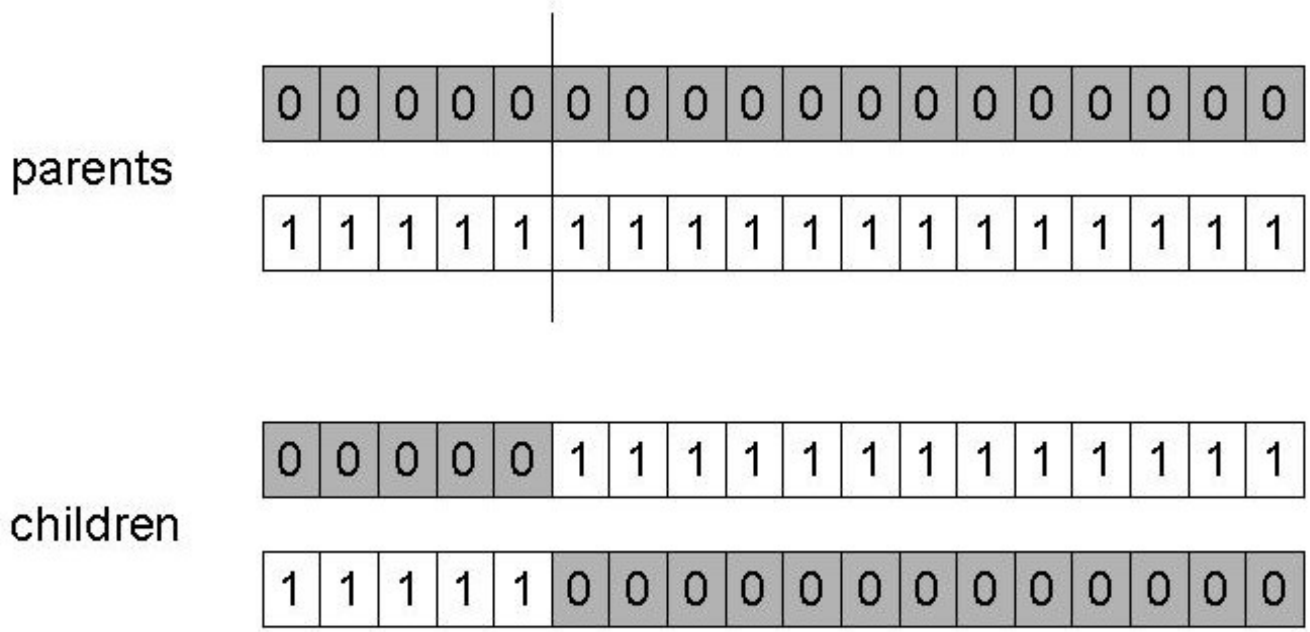
Encoding Methods

- Binary Encoding – Most common method of encoding. Chromosomes are strings of 1s and 0s and each position in the chromosome represents a particular characteristic of the problem.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Crossover

- Choose a random point on the two parents
- Split parents at this crossover point
- Create children by exchanging tails



www.cs.vu.nl/.../Genetic_Algorithms.ppt

Mutation

- Alter each gene independently with a probability p_m
- p_m is called the mutation rate
  - Typically between 1/pop_size and 1/ chromosome_length

parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

www.cs.vu.nl/.../Genetic_Algorithms.ppt

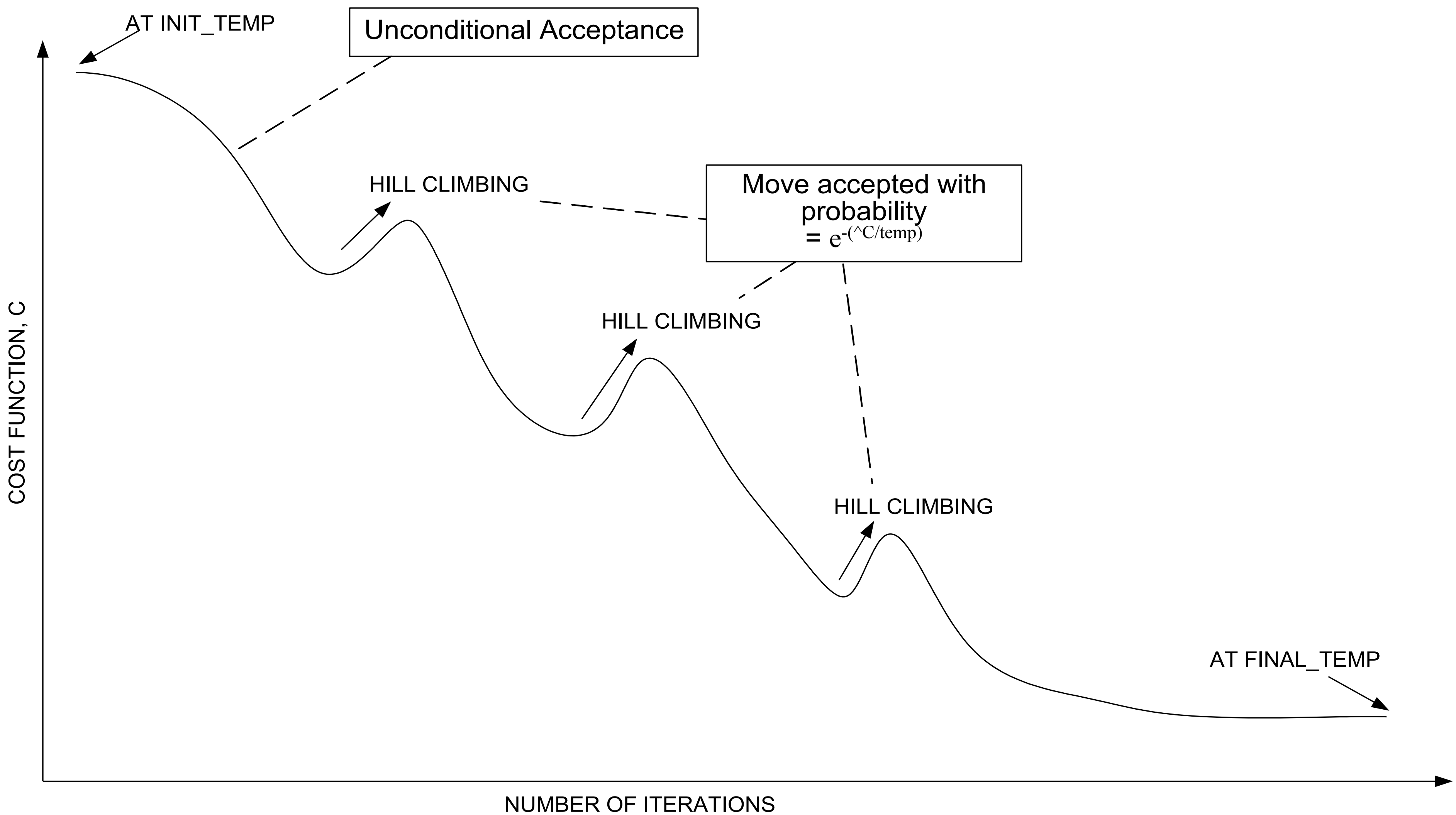
Simulated Annealing

- Based on physical concept known as "annealing" – cooling of a liquid to a solid
- Heat the solid state metal to a high temperature then cool it down slowly according to a specific schedule
- Distinctive feature is that it allows uphill directions
- Random element to search
- Claim to find global minimum in asymptotic sense

Simulated Annealing

1. Start with a random initial guess and (high) temperature
2. Perturb the current point through a defined move
3. Calculate the change in the function value due to the move made
4. If function decreases, then accept the new point
5. If function increases accept the move with a probability that depends on the current "temperature"
6. Update the temperature value by lowering the temperature
7. Repeat 2-6 until "Freezing Point" is reached

Convergence of simulated annealing



Courtesy of P. Akella, www.ecs.umass.edu/ece/labs/vlsicad/ece665/.../SimulatedAnnealing.ppt

Direct Search Methods

- Methods that "in their heart" do not use gradient information, e.g. Nelder-Mead
- Main operation is function comparisons
- Useful whenever the derivative of the objective function is not available or is too expensive to compute
- Strictly monotonic

Generating Set Search Methods

- Subset of Direct Search Methods
- Includes Pattern Search methods as well as some other variations
- Main idea is to generate a set of search directions that guarantee descent
- Main differences are in how new step is chosen and how to choose directions

Generating Set Search

procedure GSS
    Pick a set of search directions D_k
    for each $d \in D_k$ **do**,
        evaluate $f(x_k + \Delta_k d)$
    end for

Decrease Condition

    if $\exists d_k \in D_k$ s.t. $\boxed{f(x_k + \Delta_k d_k) < f(x_k) - \rho(\Delta_k)}$ **then**
        $x_{k+1} = x_k + \Delta_k d_k$
        $\Delta_{k+1} = \phi_k \Delta_k$, where $\phi_k \geq 1$

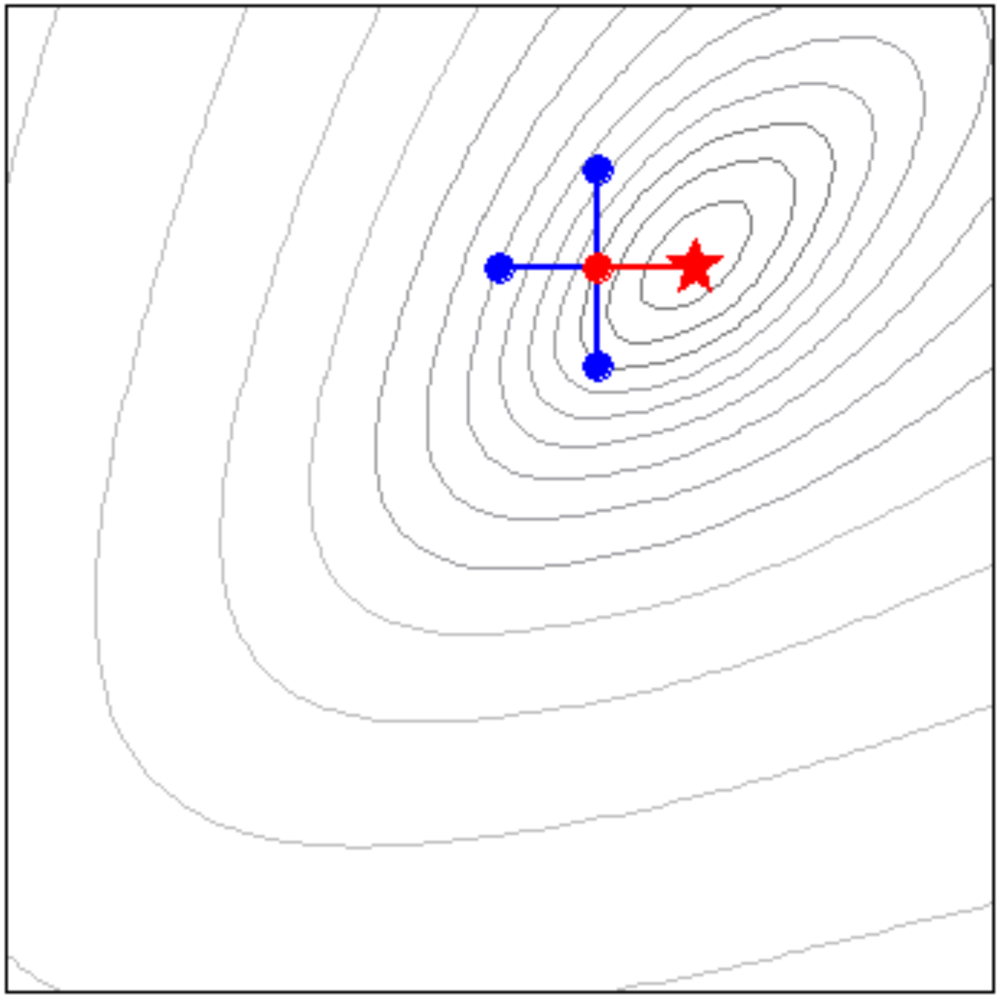
Successful Iteration

    else
        $x_{k+1} = x_k$
        $\Delta_{k+1} = \theta_k \Delta_k$, where $\theta_k \in (0, 1)$

Unsuccessful Iteration

    end if
end procedure

Pattern Search $D_k = \{\pm e_1, \pm e_2\}$



Special thanks to Tammy Kolda for this slide

Theoretical Properties of GSS

If $f(x)$ is suitably smooth …

- Guaranteed "good" descent directions
- For unsuccessful iterations, $\| \nabla f(x_k) \|$ is bounded as a function of the step length Δ_k
- Can also show: lim inf Δ_k = 0
- Conclusion:

lim inf $\| \nabla f(x_k) \|$ = 0,
i.e. Weak Global Convergence

Some observations

- GSS methods can use simple or sufficient decrease
- GSS uses multiple search directions in such a way as to ensure at least one is a descent direction
- Never uses gradient, but theory does require gradient is Lipschitz continuous

Summary

- Practical problems in science and engineering often exhibit characteristics that make standard methods difficult/impossible to use
- Many good ideas and a lot of work on derivative-free optimization
- Can often be competitive with standard methods

Future Directions

- Parallel Optimization
- Surrogate Models for expensive functions
- Optimization under uncertainty
- Non-smooth optimization
- ...

Questions?

Parallel Optimization

Schnabel (1995) identified three levels for introducing parallelism into optimization

1. Parallelize evaluation of functions, gradients, and or constraints

2. Parallelize linear algebra

3. Parallelize optimization algorithm at a high level

Parallelism is easily introduced when finite-difference gradients are used

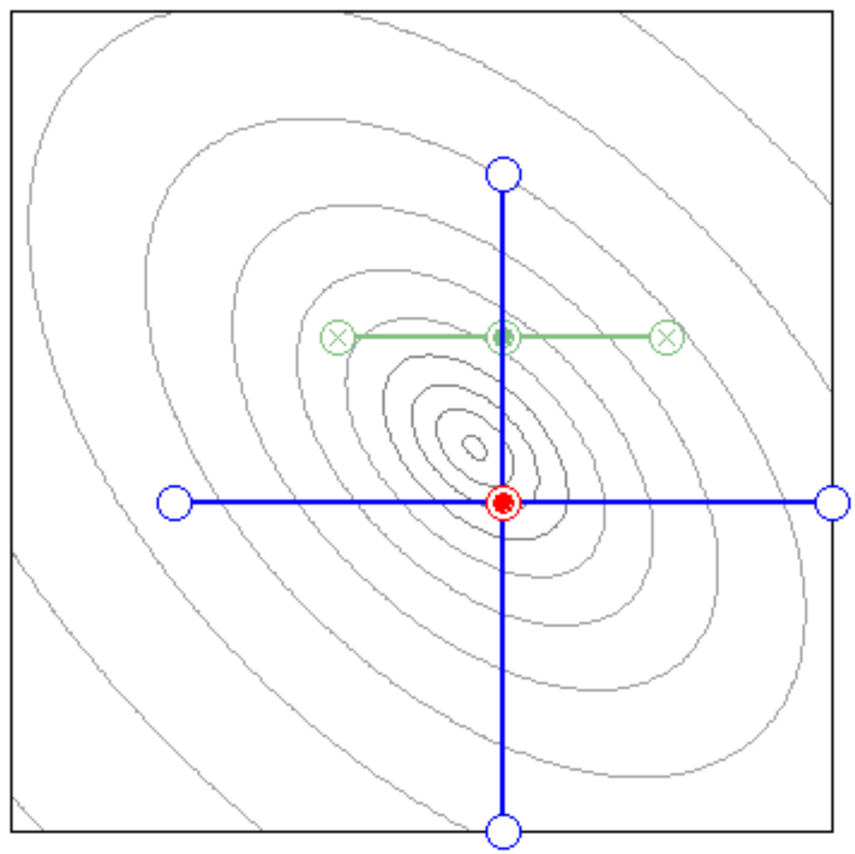
- Option 1 in Schnabel's taxonomy

- Components of the gradient can be computed independently on separate processors
- Components of the gradient can be computed speculatively (Byrd, Schnabel, Shultz, 1988)
  - trial point is accepted 60-80% of the time
  - compute components of the gradient

Parallelize the linear algebra

- Much research in this area
- Outstanding progress in recent years
- BUT, this is really only useful for large-scale optimization problems
  - If the function evaluation dominates the computational time, then this option will not prove effective

Basic Parallel Pattern Search



Special thanks to Tammy Kolda for this slide

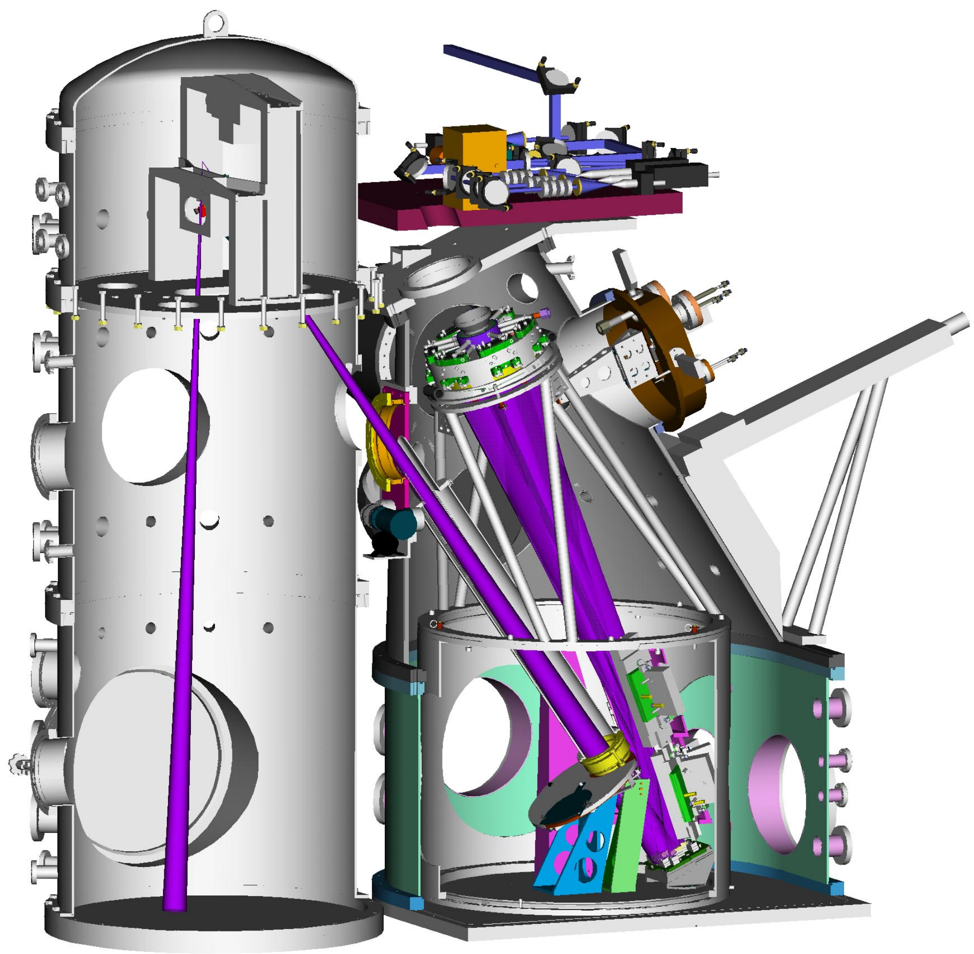
Optimization References

- Dennis and Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983
- Gill, Murray, Wright, *Practical Optimization*, Academic Press, 1981
- Stephen J. Wright, *Primal-Dual Interior-Point Methods*, SIAM 1997
- El-Bakry, Tapia, Tsuchiya, Zhang, *On the Formulation and Theory of the Newton Interior-Point Method for Nonlinear Programming*, JOTA, Vol. 89, No.3, pp.507-541, 1996
- M.J.D. Powell, *Direct search algorithms for optimization calculations*, Acta Numerica 1998
- M.H. Wright, *Direct search methods: once scorned, now respectable*, Numerical Analysis, 1995
- P. D. Hough, T. G. Kolda, and V. J. Torczon. *Asynchronous Parallel Pattern Search for Nonlinear Optimization.* SIAM J. Scientific Computing, 23(1):134-156, June 2001

Software References

- DOE ACTS Collection
  - http://acts.nersc.gov/
- APPSPACK
  - http://csmr.ca.sandia.gov/projects/apps.html
- NEOS – Network Enabled Optimization Software
  - http://www-neos.mcs.anl.gov/neos
- General Software
  - http://sal.kachinatech.com/B/3/index.shtml

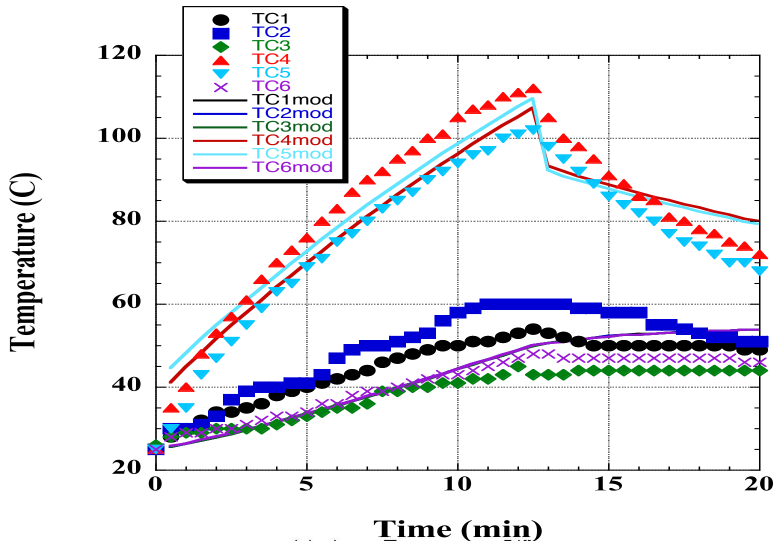
Parameter identification example

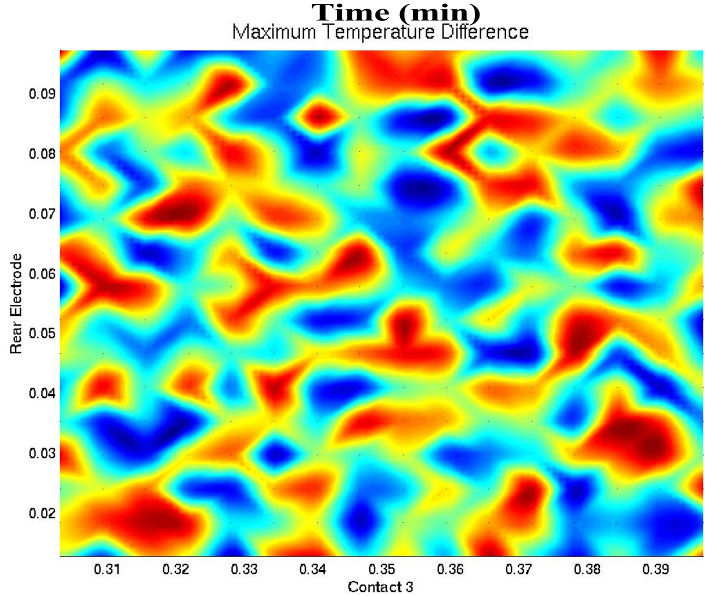


- Find model parameters, satisfying some bounds, for which the simulation matches the observed temperature profiles
- Computing objective function requires running thermal analysis code

$$\min_{x} \quad \sum_{i=1}^{N} (T_i(x) - T_i^*)^2$$
$$\text{s. t.} \quad 0 \le x \le u$$

Data Fitting Example





- Objective function consists of computing the max temperature difference over 5 curves
- Each simulation requires approximately 7 hours on 1 processor
- Uncertainty in both the measurements and the model parameters

Derivation of Newton equations

- Build quadratic model

$$q(x_k + s) = f(x_k) + g(x_k)^T s + \tfrac{1}{2} s^T H(x_k) s$$

- Find the minimizer of the quadratic

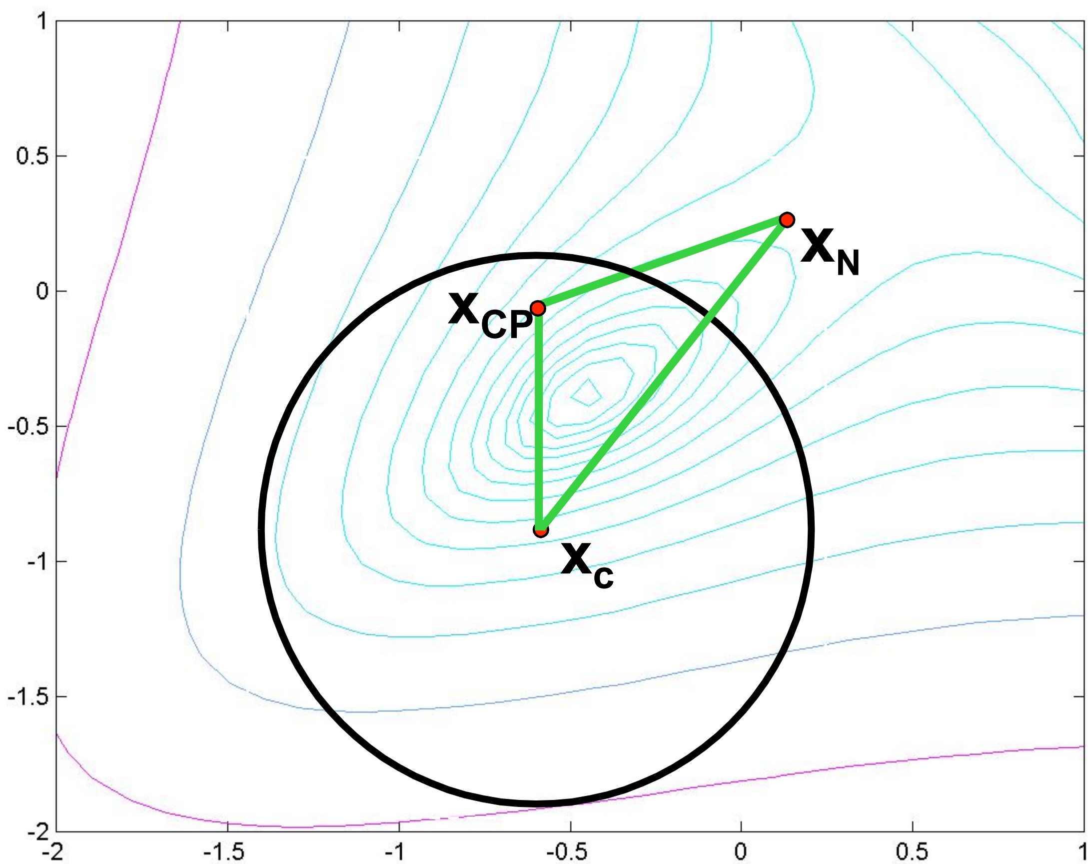
$$\min q(x) \quad \Leftrightarrow \quad \nabla q(x) = g + Hs = 0$$

$$\begin{array}{ll} Solve & Hs_k = -g \\ Set & x_{k+1} = x_k + \alpha \cdot s_k \end{array}$$

- Check how well you did, i.e. is

$$f(x_{k+1}) < f(x_k)$$

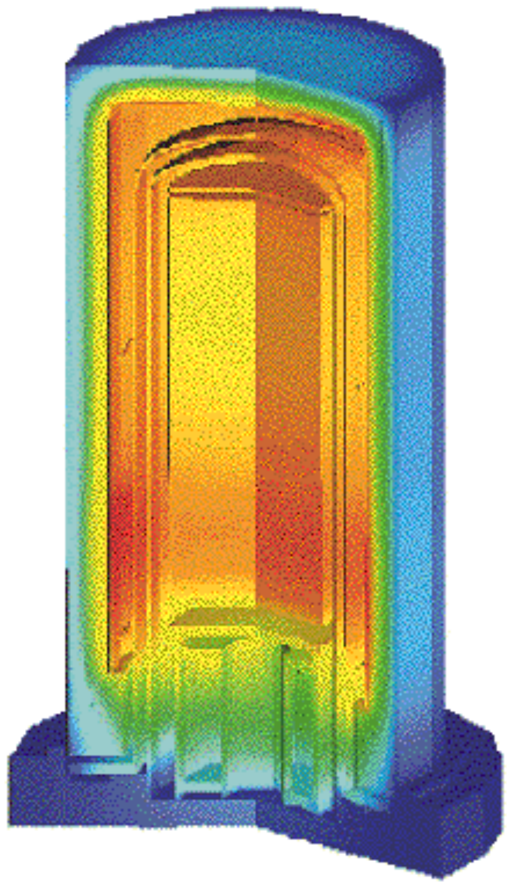
Newton Methods



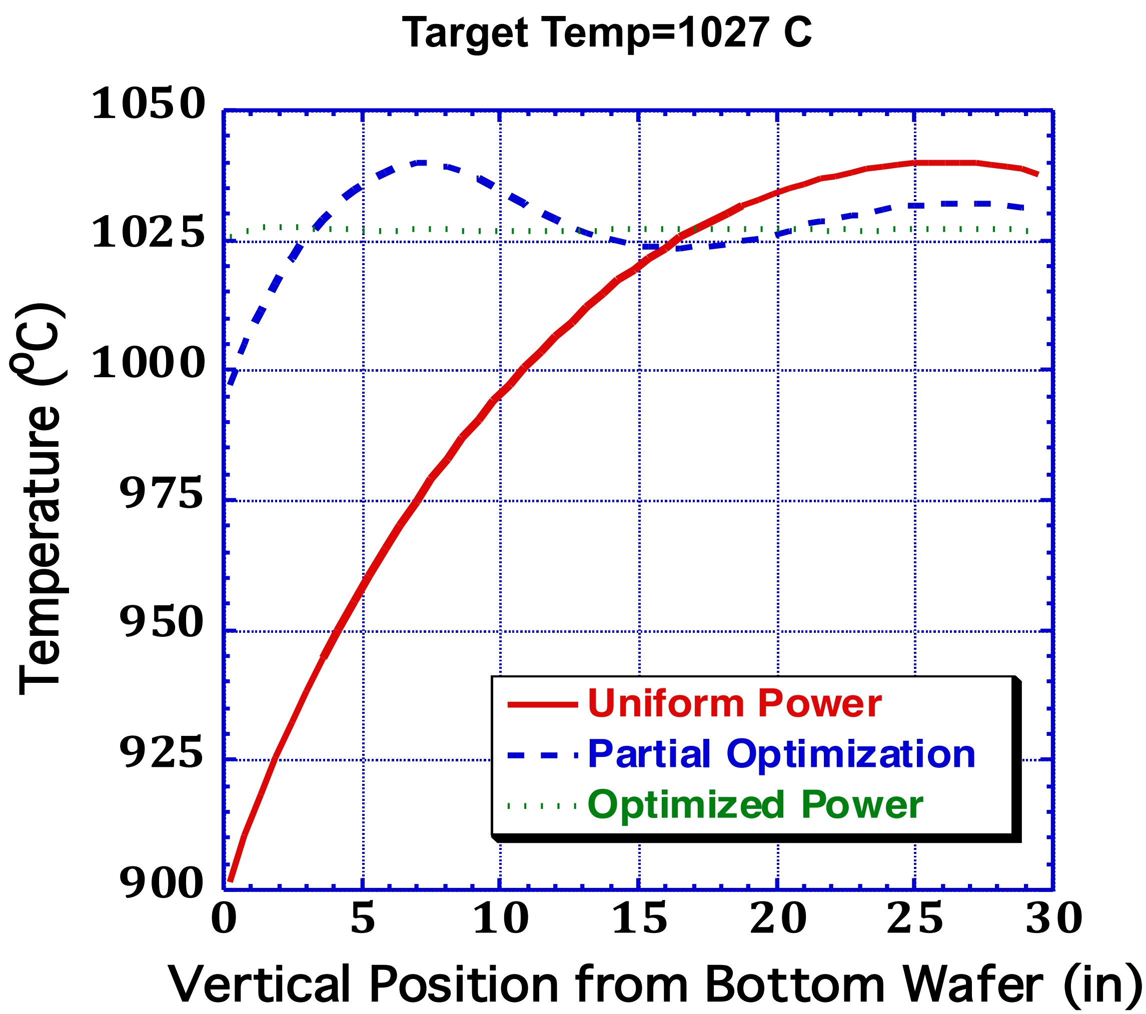
- Fast convergence properties
- Good global convergence properties
- Quasi-Newton approximations work well in practice

- Inherently serial
- Difficulties with noisy functions

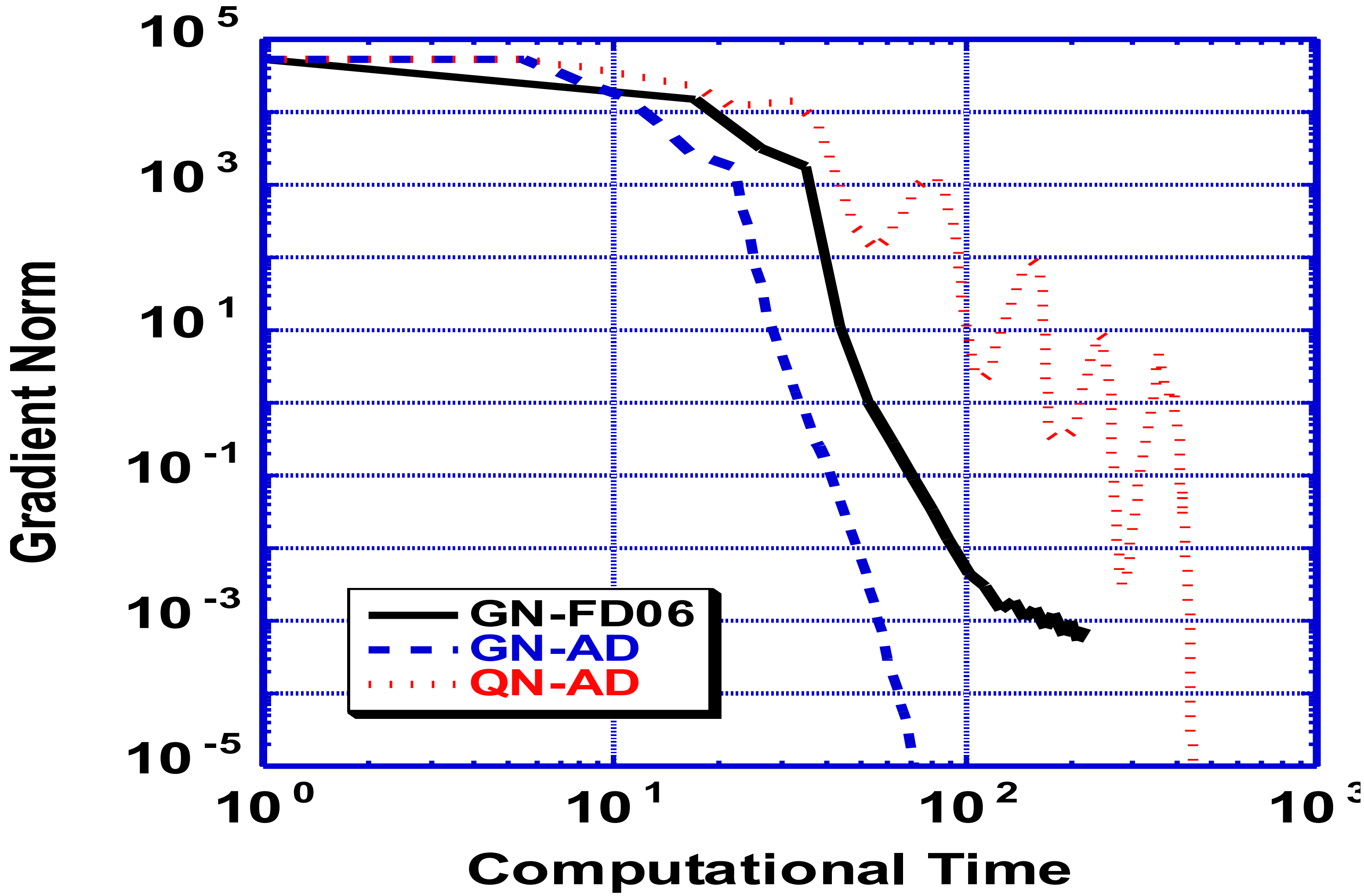
Optimized Power Distribution



Temperature fields in a vertical, stacked-wafer, low-pressure, chemical-vapor-deposition furnace



Analytic Gradients vs. Finite-Differences



General observations

- Many optimization problems have expensive objective functions
  - Objective function requires solution to a large-scale PDE or similar type of simulation
  - One function evaluation can take several CPU hours even on a parallel processor
- Adding more processors to the function evaluation is not always efficient or productive
  - Many applications do not scale well
- May not even be able to parallelize the objective function
  - Black-box functions