# Statistical Challenges for Global Optimization Experiments

William Hart

Sandia National Labs

wehart@sandia.gov

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

# *Overview*

Goal: Discuss opportunities for applying statistical methods to analyze and improve algorithms for global optimization

Note:

- This presentation is not a review
- I am trying to highlight contexts where I think we could develop better statistical methods
- This need is motivated by the fact that statistical methods are rarely used to analyze global optimization methods!



Sandia National Laboratories

# The Global Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0 \\ & h(x) = 0 \\ & x \in D \end{aligned}$$

Considerations:

- $D$ may be discrete or continuous or a mixture of both
- f, g and h may have special structure

  Examples: convex, multimodal, differentiable
- There may be several notions of a "local neighborhood"

Goal: find a globally optimal point

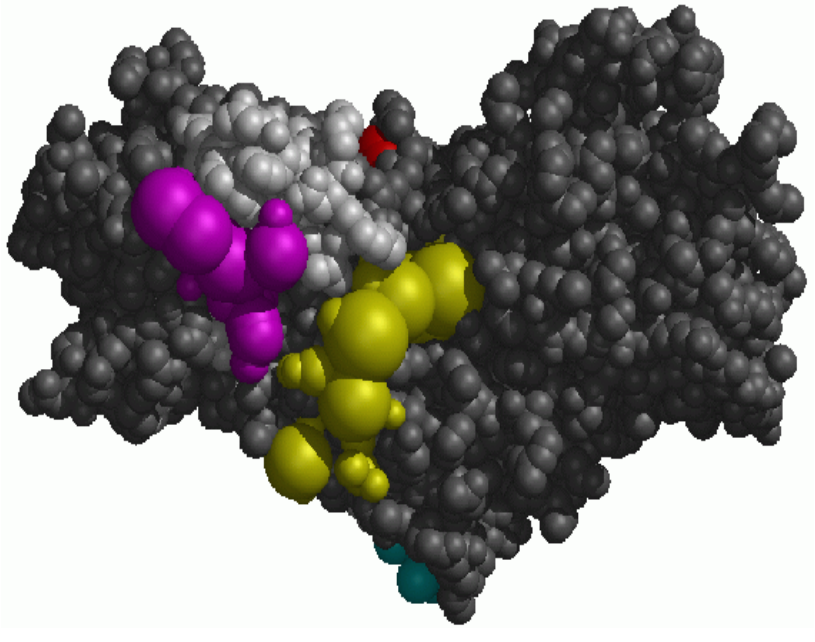
... in a robust, efficiently, straightforward, general manner!

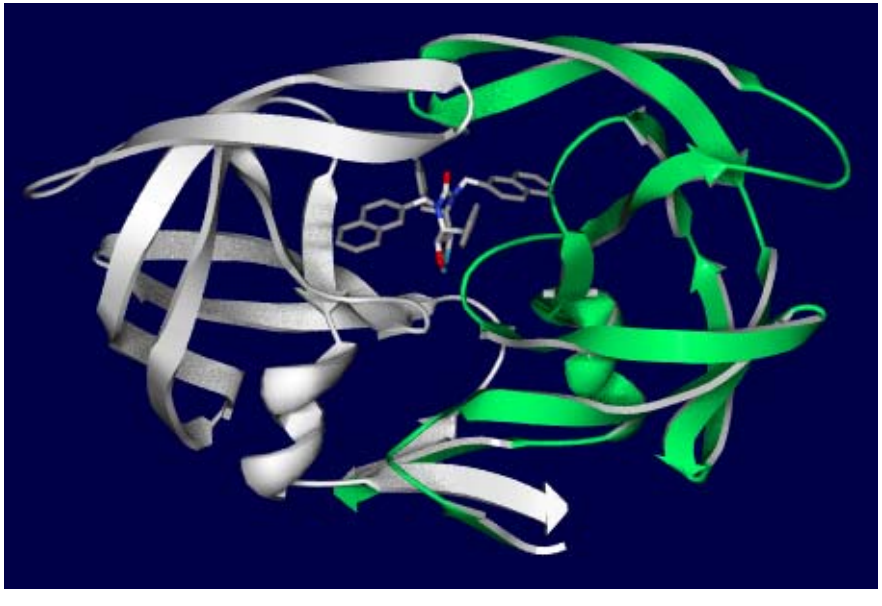


Sandia National Laboratories

# Example — Drug Docking

- Problem: find the optimal binding for a small ligand in a binding site





- Application: lead compound development

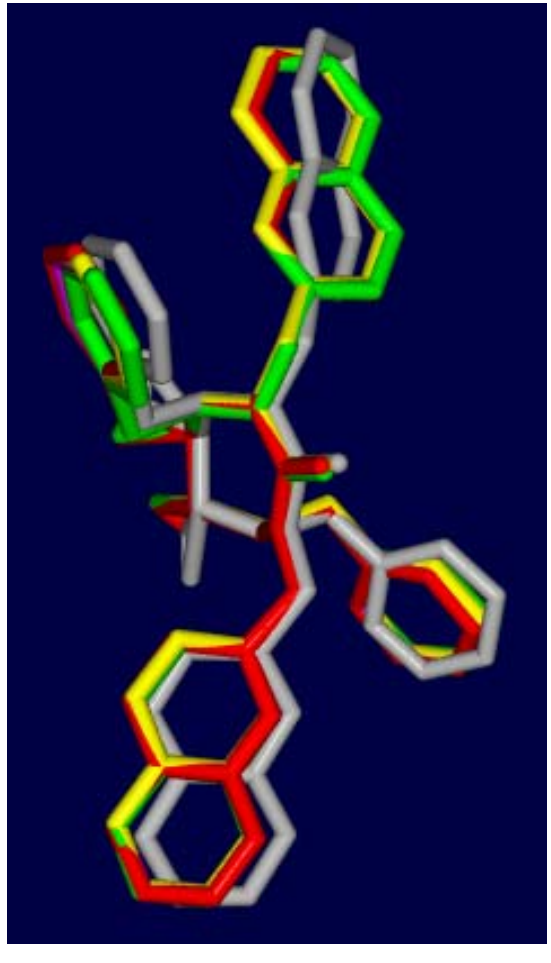
- Impact: limit laboratory experimentation required to develop new drugs



Sandia National Laboratories

# Flexible Drug Docking in AutoDock

Idea: dock a small flexible ligand into a static binding site



Search Dimensions:

- Cartesian Coordinates (3 dimensions)
- Quarternion (unit vector plus rotation angle)
- Torsion angles

Formulation:

- Nonsmooth energy potential
  - Standard energy model: electrostatic, van de Waals, Hydrogen bonds, etc
  - Trilinear interpolation of atomic interactions
- Simple bound constraints
- Unit-length quarternion constraint

Note: Binding constraints or ring constraints possible



Sandia National Laboratories

# Algorithmic Techniques Used for GO

Box decomposition
- Interval methods, MILP, DIRECT, Lipshitzian search

Population search
- EAs, scatter search, controlled random search, differential evolution

Response surface modeling
- Minimize surrogate functions, maximize expected improvement, etc

Global-local search
- Multistart-LS, memetic EAs, GRASP, clustering/topographical methods

Iterated local search
- Simulated annealing, tabu search, ant systems

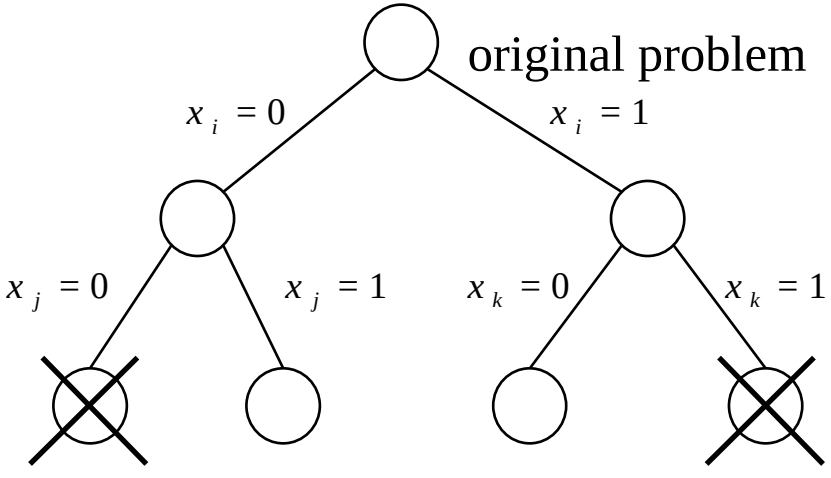


Sandia National Laboratories

# Exact Optimization: Branch and Bound

Idea:

- Subdivide the search domain into non-overlapping regions
- Ranked regions with estimates of their best value
- Subdivide regions in order of their rank




Practical Issues:

- Terminate with exact/approximate solution in finite time
- Too slow for large applications
- Ranking/bounding techniques are not available for complex applications



Sandia National Laboratories

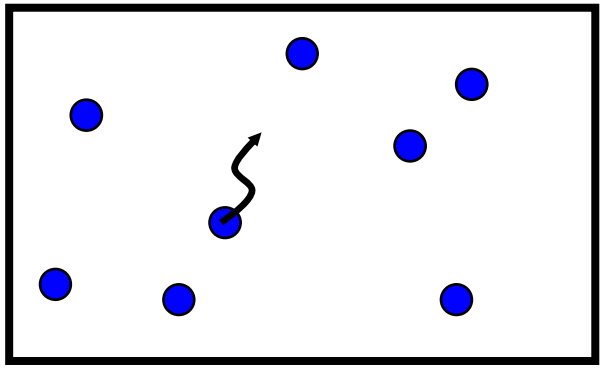
# Global-Local Search

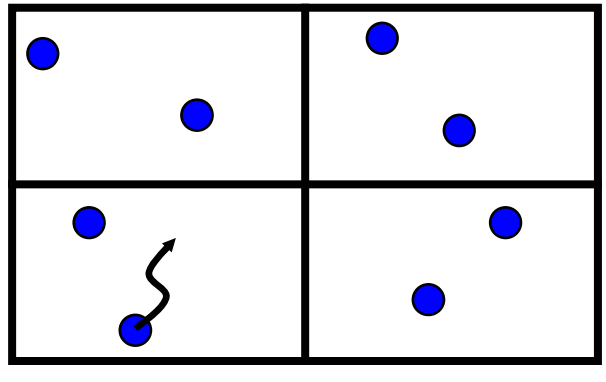
Idea:

- Generate initial points from a fixed distribution
- Apply domain-specific local search methods to refine solutions
- Asymptotically guarantees convergence w.p.1

Practical Issues:

- Weak stopping rules have been developed (e.g. Hart '99)
- This a standard benchmark method
- This can be improved by partitioning the random samples







Sandia National Laboratories

# Global-Local Search (cont'd)

GRASP:

- Multistart technique with semi-greedy initialization
- Semi-greedy approach
  - Greedy function: evaluates the value of augmenting the current partial solution
  - Semi-Greedy: rank augmentations, and pick randomly from among the best ♋ percent

Cluster/Topological Methods:

- Idea: limit the total number of local searches
- Identify local search seeds from random samples
  - Cluster: apply quick local search, cluster, seed LS with cluster center
  - Topological: apply local search to points that are 'downhill' from sufficiently many nearby points
- Note: clustering methods can provably require finitely many LSs!



Sandia National Laboratories

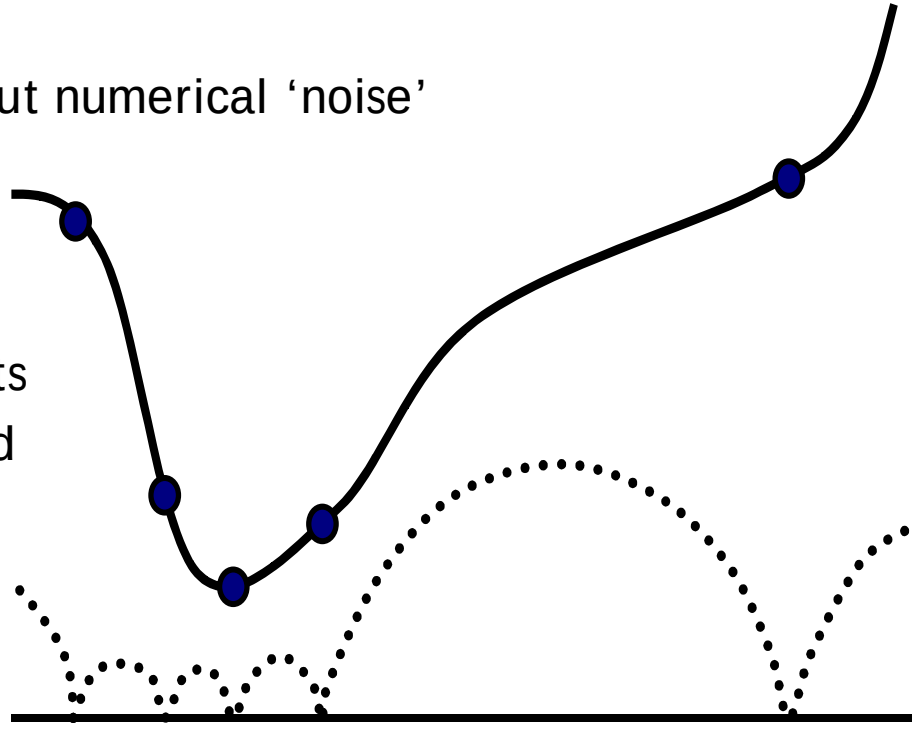
# Response Surface Modeling

Idea:

- Model the objective function and/or constraints using response surfaces
- Analyze/optimize the response surface to identify trial points
- Evaluate these trial points and update the response surface model
- Note: very effective for applications with expensive objective/constraint functions
- Note: also effective at smoothing out numerical 'noise'

Bayesian Methods:

- Stochastic model of possible functions that match the data points
- Analyze both the function value and model uncertainty
- Select a point to maximize the probability of improvement over the best point seen so far



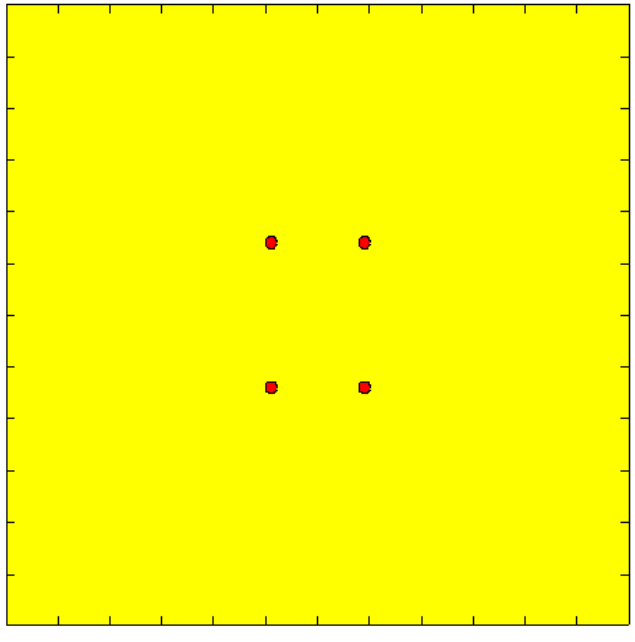


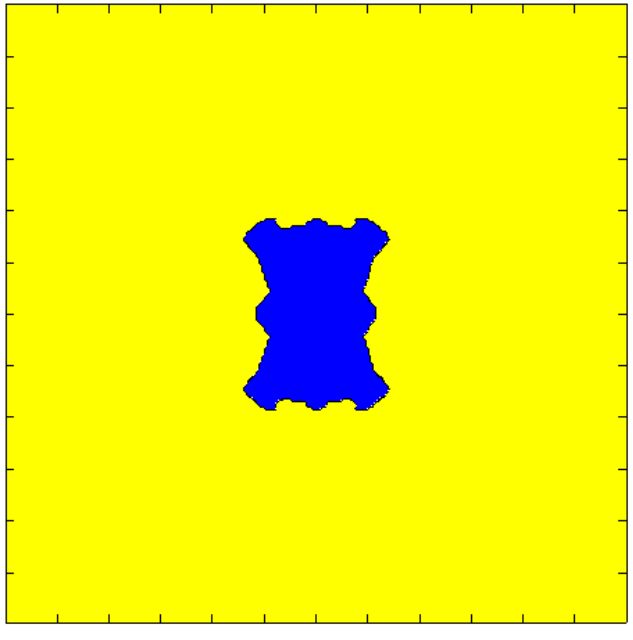
Sandia National Laboratories

# Response Surface Modeling (cont'd)

Example: Explosives Placement

- Goal: maximize the size of a cavity created by explosives
- Constraints: cost, placement time
- Very noisy, expensive simulation required to evaluate each trial point





Application: demolition, mining



Sandia National Laboratories

# Iterated Local Search

Idea:

- Algorithmic frameworks for local search
  - Define neighborhood search operator
  - Define restart mechanism
  - Define solution similarity metric

Note:

- Easily customized to new problem domains
- Often find better local minima than canonical local search methods
- Often randomized
  - e.g. adaptive/biased sampling methods
- Often inspired by natural processes



Sandia National Laboratories

# Iterated Local Search (cont'd)

Simulated Annealing:

- A probabilistic local search procedure inspired by physical cooling processes
  - A trial point is sampled from a local neighborhood
  - The point is accepted or rejected using a probabilistic criterion
  - The acceptance criterion is updated so that accepting a worse point is less probable in the next round.
- If the probability of rejection is 'cooled' sufficiently slowly then some variants of simulated annealing provably converge in probability.

Ant Colony Optimization:

- Inspired from collective behaviors in ant colonies
- Ant reinforcement strategies can be tailored to solve optimization problems
  - Both local and global reinforcement
- Note: well-suited for a variety of graph-based combinatorial problems



Sandia National Laboratories

# Population Search

Idea:
- Maintain a set/population of points in each iteration
- Generate new points with
  - Local steps about points
  - Recombination steps using two or more ‘parents’
- Keep the ‘best’ points seen so far

Genetic Algorithms:
- A general population-based meta-heuristic
- Exploits interplay between exploration and exploitation
  - Global sampling: cross-over between diverse solutions
  - Local sampling: mutation around solutions
  - Various randomized selection strategies to subselection solutions in each iteration (generation)
- Can use local search



Sandia National Laboratories

# MetaHeuristics?

Idea: a framework for defining optimization

Examples:

- Genetic algorithms
- Tabu search
- GRASP
- Simulated Annealing

Note: you could include others here as well

- Branch-and-bound
- Sequential quadratic programming



Sandia National Laboratories

# Analysis of Global Optimization Methods

Convergence theories look like:

- Let X_t be the stochastic process defined by the best point found by the global optimizer in each iteration
- For any suitably chosen function f(), f(X_t) -> f* with probability one, where f*

OR

- In a finite number of iterations, $|f^* - f(x_t)| < \varepsilon$

These are asymptotic results that have little practical relevance.



Sandia National Laboratories

# Analysis and Comparison of Global Optimizers

Experimental comparisons are the standard approach to analyzing global optimizers

- Complex algorithms
- Little distinction between software and algorithm

Q: which code works better for my application?

Typical analysis:

- Run two or more optimizers on a finite set of test problems
- Terminate optimizers after a fixed time limit
- Compare values of best solutions
  - Worst, Mean, Median, Max



Sandia National Laboratories

# Overview of Statistical Challenges

Statistical Methods for Optimization

Estimating Solution Time

Accounting for Optimizer Variability

Assessing Optimizer Scalability

Large-Scale Performance Comparison



Sandia National Laboratories

# *Statistical Methods for Optimization*

The most common comparison is between the mean performance of two or more optimizers

- Typically a comparison of mean solution values after a given level of effort (runtime or function values)

Problem: Hypothesis testing is still not widespread

- Statistical methods are rarely used in optimization articles

Possible explanations:

- Many optimization researchers come from disciplines that are not strongly experimental
- The complexity of statistical methods is an issue for many users
- There are a variety of gaps between classic experimental methods and methods needed for computer experiments



Sandia National Laboratories

# Idea: A Process For Model Validation

Observation: Statistical text books and tutorials typically focus on methods (hypothesis tests, regression, etc)

Problem: model validation is an important issue for computational experiments

- It can be difficult to predict the distribution of data
- Bimodal distributions are not uncommon when characterizing optimizer behavior

Challenge: develop experimental best-practices

- This is more than just developing techniques
- This is about developing *processes* that are easy to follow



Sandia National Laboratories

# Example: An experimental sequential analysis

The steps in this analysis have been described by many authors...

... but this process has not been socialized within the optimization research community

Text books and papers on statistical analysis focus on methods

A key challenge is to guide users when a textbook analysis is not sufficient

Ridge and Kudenko 2006

1. Check blocking
2. Check correlation
3. Note Best Responses
4. Find important effects
5. Rank Important Effects
6. ANOVA test
7. Model Diagnosis: normality, constant variance, etc
8. Model power
9. Model significance
10. Model Reduction
11. Model Fit
12. Examine Aliasing



Sandia National Laboratories

# Possible Next Steps

- Better tools for computational experimentation
  - Idea: implement this sort of statistical analysis for typical experimental contexts
    - Explanatory descriptions of results
  - Idea: tools to automate the evaluation of statistical models
    - Example: the use of data transformations
    - It does not really matter if these are computationally expensive

- Books and tutorials that focus on the process of statistical analysis
  - Idea: assume reader is familiar will statistical methods
  - Goal: describe the process of statistical analysis
  - Example: The Little LISPer
    - A clear, simple, interesting, and complete book that succinctly teaches the reader how to use LISP



Sandia National Laboratories

# *Estimating Solution Time*

Goal: find a globally optimal (or near-optimal) solution

Problem: it is difficult to predict how long a global optimizer will need to solve a problem

- Complex applications may require "too many" optimization iterations
- Randomized optimizers have runtime distributions with fat tails
- Heuristic optimizers typically have weak stopping criteria

Impact: optimization data is often right censored

- Optimizers are often terminated after a specified time limit
- The final solution value can be assessed
- Cannot assess the time needed to find a near-optimal solution for all experimental trials



Sandia National Laboratories

# Idea: Leverage Survival Analysis

Can we apply survival analysis techniques developed for the biological and medical sciences?

How can survival functions, hazard functions and mean residual life functions inform our selection of optimizers?

Can we expect that these will be similar for different types of optimizers?

Can we compare multiple algorithms with these methods?



Sandia National Laboratories

# Restarting Optimizers

Observation:

- The search in heuristic optimizers can stagnate and fail to search globally
- Restarting heuristic optimizers with new initial conditions can be a more effective search strategy in many contexts

Q: can survival analysis inform the design of restart methods?

Example: Hoos and Stutzle, 2000

- Empirical analysis of iterated local search for 3SAT
- Demonstrate that the runtime distribution to find a solution with a given value is exponential
- Thus, restart methods are not useful for these optimizers



Sandia National Laboratories

# Idea: Adaptively Estimate Median Runtime

Problem: optimizer runtime distributions often have thick tails

- Estimating mean runtime can be expensive
- Estimates of mean with right censored data are suspect

Idea: estimate *median* runtime in an iterative manner

- Iteratively estimate bounds on the estimate
- Truncate optimization runs beyond these bounds to minimize compute time

Goal: estimate median performance without performing lengthy optimizations



Sandia National Laboratories

# Algorithmic Issues

Bounds:

- What type of bounds should be used?
- Are these tight enough to make this practical?
- Should we bound the median, or a larger percentile
  - We can get tighter bounds on the 75$^{th}$ percentile, but that might cost more

Restarts:

- Should we restart truncated optimization trials, or pick new trials?

Analysis:

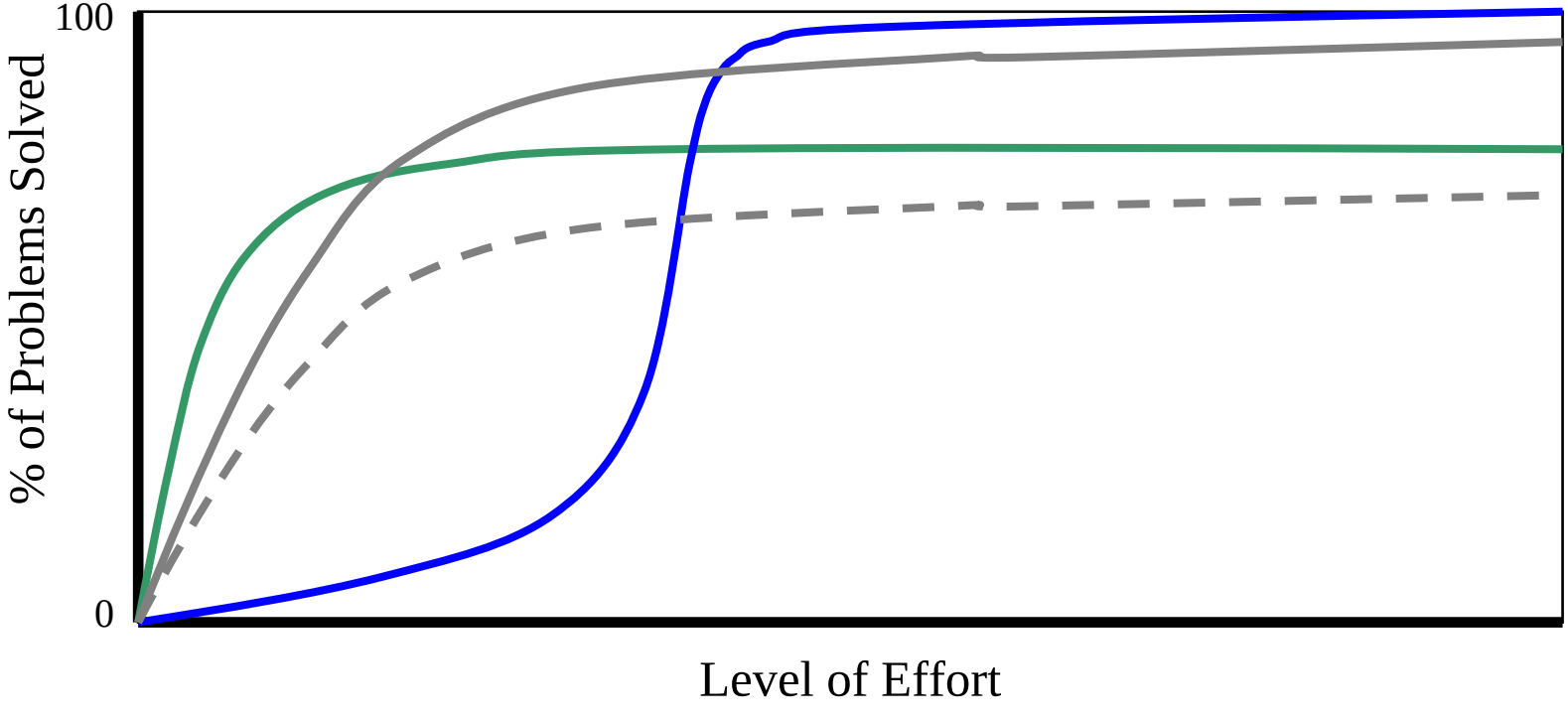
- Can we predict the expected cost of this process?
- CS methods can be used, but they are asymptotic in the number of samples…



Sandia National Laboratories

# Accounting for Optimizer Variability

Optimization studies are increasingly comparing runtime differences with cumulative "solvability" plots





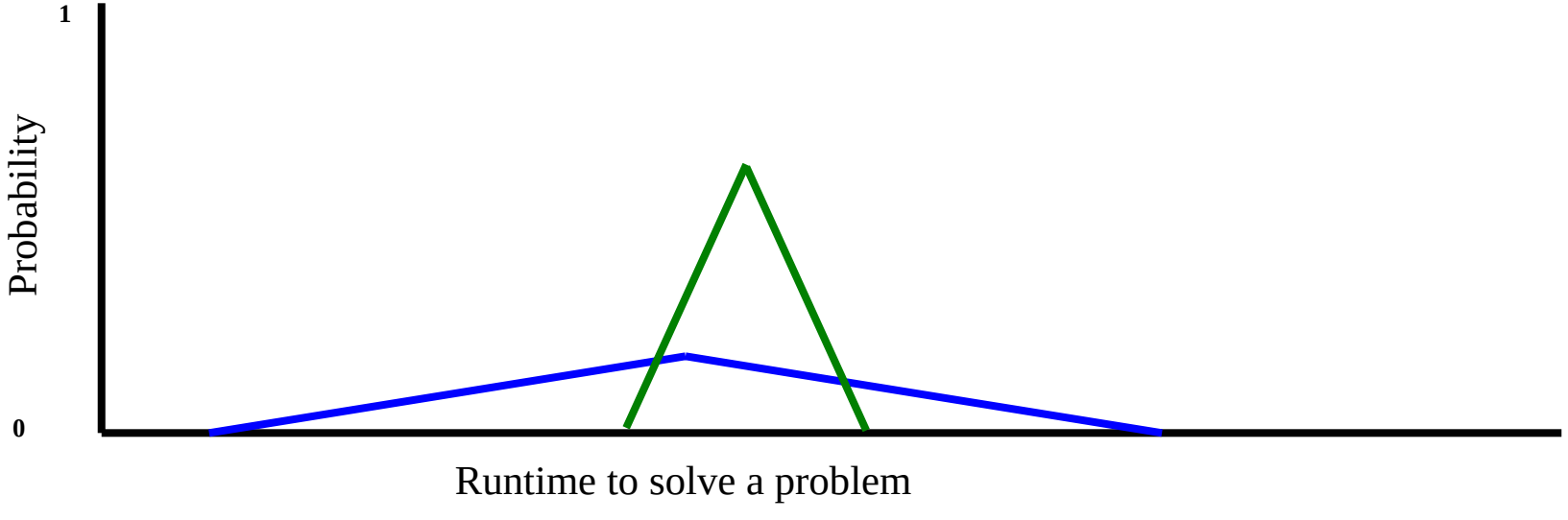

Sandia National Laboratories

# Evaluating Optimizers

Problem: these plots do not adequately describe the variability of optimizer runtimes

We do not simply want the optimizer with the best mean performance

- Potentially long runtimes are very undesirable to users






Sandia National Laboratories

# Idea: Robust Performance Assessment

Challenge: account for solver variability as a separate design criteria

Idea: treat algorithmic comparison as a bicriteria assessment

- Characterize mode and variability
- Would the goal be to identify optimizers with nondominated statistics?
- How could we leverage existing statistical techniques?

Idea: perform statistical comparison with robust statistics

- Example: compare extreme statistics
- Do robust statistical methods exist for these measures?



Sandia National Laboratories

# *Assessing Optimizer Scalability*

Observation: large, real-world problems are often quite different from test problems

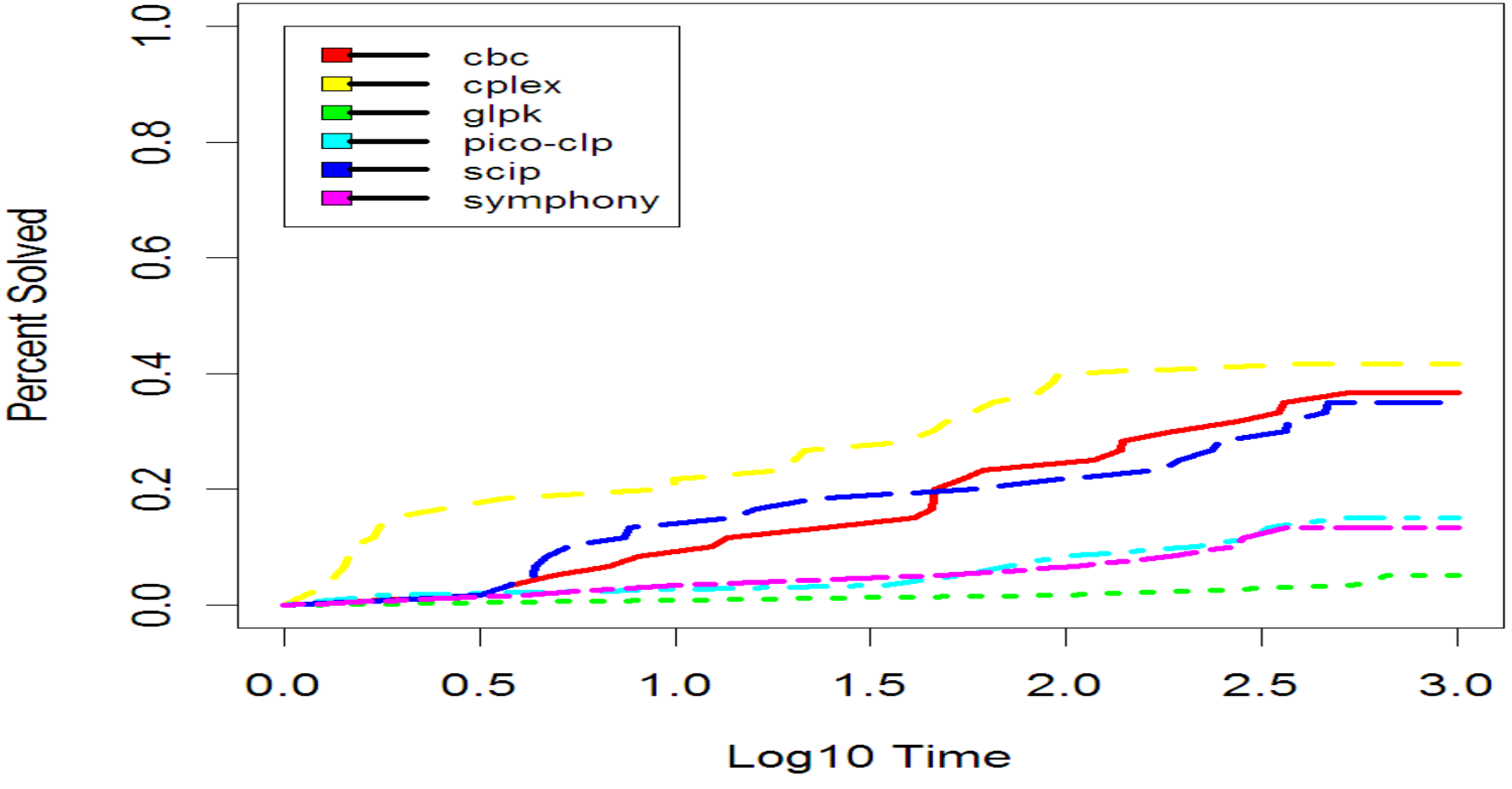
Challenge: efficiently compare the scalability of optimizers



Sandia National Laboratories

# Example: Fraction of Problems Solved – 10 minutes

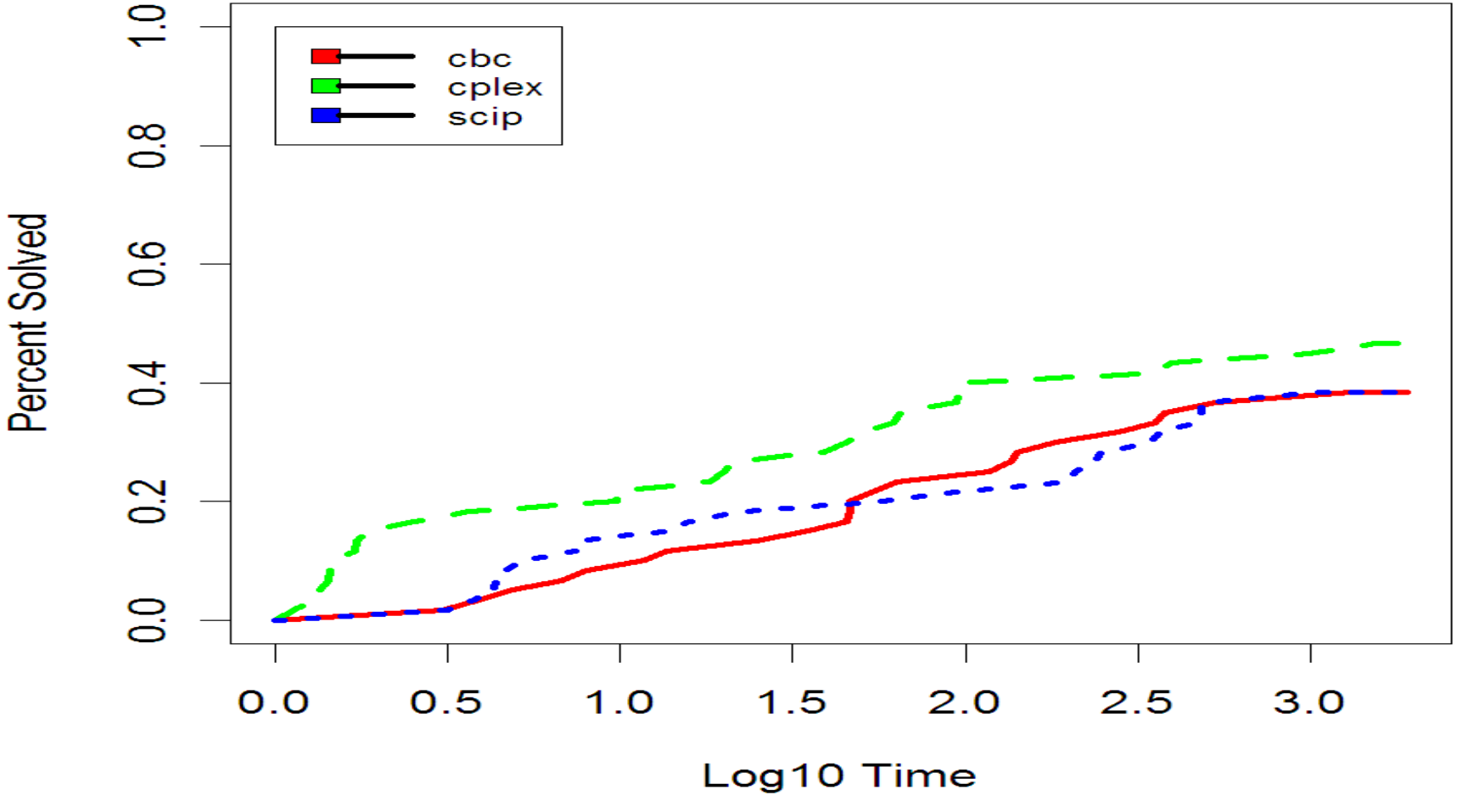
Note: the experimental runtime is dominated by trials in which the optimal solution is not found!






Sandia National Laboratories

# Example: Fraction of Problems Solved – 30 minutes






Sandia National Laboratories

# Idea: DOE for Scalability

Idea: Treat scalability as a factor

- Can we design experiments that treat runtime as a separate control variable and then analyze this dimension?
- Can we perform "paired" experiments and adaptively truncate them at different final solution values?

Idea: Adaptive experiments that terminate optimizers when a near-optimal solution is found

- Can we treat this as a method for generating right censored data?
  - Idea: run different experiments with different thresholds
- Can we leverage this to assess scalability questions?



Sandia National Laboratories

# Scalability of Exact Solvers

For exact solvers, we can usually only solve a fraction of the problems to optimality

Exact methods can determine that an optimal solution is found

Thus, we have data on the runtime to solve problems, and right censored data with final solution values

How do we integrate this data into a quantitative comparison?

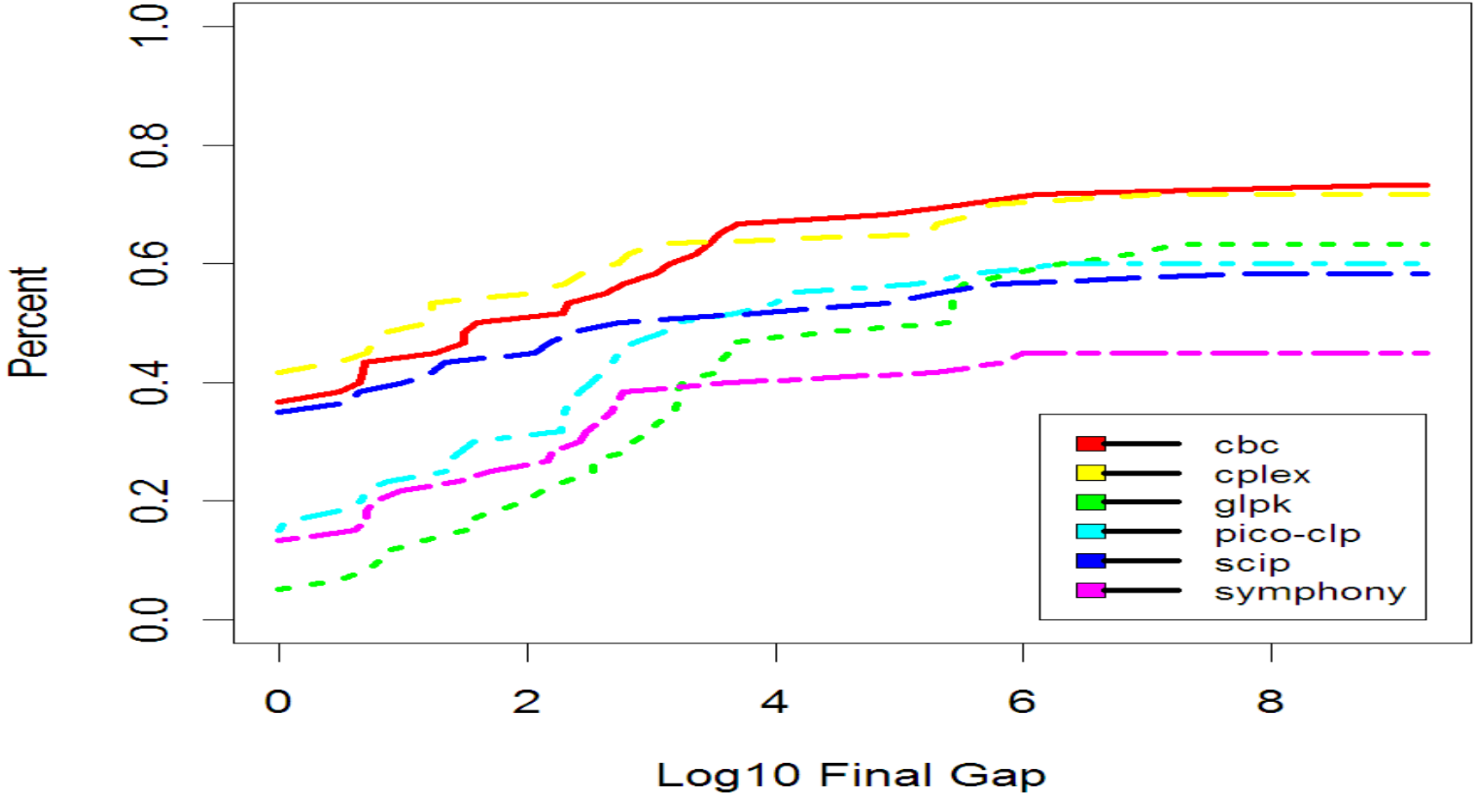
Are there better summary methods than the cumulative performance graphs?

- e.g. methods like decision trees might better summarize randomized optimizers



Sandia National Laboratories

# Example: Assess Gap with Known Global Minimum






Sandia National Laboratories

# *Large-Scale Performance Comparison*

Challenge:

- Optimizers can have lots of tunable parameters
- Researchers often want to run them on many test problems
- Randomization is a common feature
- Some optimizers are influenced by environmental factors
  - E.g. parallel methods can be intensity nondeterministic
- Performing global optimization on a single problem can be expensive

Goals:

- Identifying key algorithmic parameters
- Optimizing algorithmic parameters
- Design and analysis of optimization experiments



Sandia National Laboratories

# 1. Identifying Key Algorithmic Parameters

Observation: optimizers can have lots of tunable parameters

- Search options: control how search is performed
- Algorithmic parameters: control behavior of search options

Example: hybrid genetic algorithms

- Crossover type: uniform, two-point, none
  - Crossover rate
- Mutation type: normal, Cauchy, uniform
  - Mutation rate, scale
- Local search type: pattern search, quasi-Newton
  - Step scale, step tolerance



Sandia National Laboratories

# Idea: Experimental Design

What are best practices for experiment designs to identify key parameters?

- This is not an uncommon practice
- Not often a focus of research

Research Challenges:

- Methods that can design experiments with mixed-levels
- Methods that can address nesting of algorithmic parameters
- Methods that work well for 10s-100s of parameters

Example: Xu, 2002

- Heuristic for constructing nearly-orthogonal arrays
- Generated arrays with mixed-levels and few runs



Sandia National Laboratories

# 2. Tuning Optimizer Parameters

Observation: researchers typically tune their algorithmic parameters by hand (or not at all)

Challenge: tune parameters to maximize performance while not over-fitting to the problem test set

- Need to account for optimizer variability
- Need to account for generalization
- May need to account for multiple performance criteria
  - Runtime, final value, etc
- May need to handle both discrete and continuous parameters



Sandia National Laboratories

# Idea: Experimental Design

Use fractional factorial designs to sample intelligently

Statistical analyses of variance and parameter interactions



Sandia National Laboratories

# Fractional Factorial Designs For Tuning

Idea: treat tuning as an experimental analysis
(Enda and Kudenko, 2007)

- Discretization of tuning parameters
  - Hi/Low values
- Experimental design on few parameters
  - $2_{IV}^{12-5}$ fractional factorial design
  - 8 replicates and 24 center points
  - Randomly generated test problems
    - Problem variability was a factor
- Analysis with ANOVA

Note: the response surface exhibits curvature, so predictions from this linear model need to be made with caution



Sandia National Laboratories

# Idea: Stochastic Optimization

These methods use

- Fractional factorial designs help sample intelligently
- Model-based analyses help minimize the number of experiments

But this is fundamentally a stochastic optimization problem

- Need to find interesting parameters
- Need to validate that it has better performance

Challenges:

- How model both discrete and continuous parameters?
- How can we account for generalization beyond the test set?
- How can we tune with multiple performance criteria?



Sandia National Laboratories

# CALIBRA

Idea: use Taguchi's fractional factorial experimental designs coupled with local search (Adenso-Diaz and Laguna, 2006)

1. Perform initial experiments to determine best/worst values for each parameter
2. Perform local search
   - Select initial values from initial experiments
   - Perform Taguchi $L_9(3^4)$ experiment with these values
   - Update values and iterate
   - Terminate when no improving values are found
3. Possibly perform multiple local searches

Note: implementation of CALIBRA appears limited to few parameters, but it could be generalized



Sandia National Laboratories

# Sequential Parameter Optimization

Idea: minimize cost of tuning by using DACE techniques
(Bartz–Beielstein et. al. 2005)

1. Select initial points with LHS
   - Round to generate integer values
   - Use repeated trials
2. Fit a second-order polynomial model with trial data
3. Select new points using this model
   - Use LHS again to generate many points
   - Select points with the highest likelihood of being better than the best
4. Iterate
   - Increase number of random trials

Note: this approach is probably not well-suited for binary parameters



Sandia National Laboratories

# F-race

Idea: iteratively focus comparisons on statistically promising parameters (Birattari et.al. 2004)

1. Select a fixed set of parameter configurations with a full-factorial design
2. Iteratively
   - Run configurations on new test problems
   - Eliminate a configuration if it is statistically inferior to at least one other configuration
     - Friedman two-way analysis of variance by ranks

Idea: an iterative version of F-race has been developed to avoid the initial full-factorial analysis

Note: F-race requires discretization of parameters



Sandia National Laboratories

# Tuning – Final Thoughts

- Authors of these methods fail to make comparisons
  - Is there a clear metric for evaluating a tuner?
  - These look like quick-and-dirty stochastic optimizers…
- These methods do not explicitly consider different performance objectives: time to solution and reliability of solver
- Solving mixed-continuous tuning remains a challenge
- It's not clear how to avoid overfiting a test-set with these methods
- Nested parameter choices remains a key challenge
  - This is directly related to experimental algorithmic design



Sandia National Laboratories

# 3. Optimization Experiments

Suppose the we have

- selected a set of key parameters and
- tuned parameters for our optimizer…

… then, we are still left with the challenge of designing an experiment to compare this with other optimizers

Challenge:

- There can be lots of test problems
- Pseudo-random and random effects need to be controlled



Sandia National Laboratories

# Problem Factors

A fixed set of test problems

- We want optimizer to work well on all possible problems
- Test problems represent a sample from a distribution of possible optimization problems

Note: can treat test problems as a *random effect*

- This does not appear to be widely recognized

How does this impact the design and analysis of experiments?

- Blocking of problem factors is common
- This is typically done by treating problems as a nuisance factor
- Can we use design of experiments with incomplete blocking?



Sandia National Laboratories

# Idea: Analysis with Random Effects

Goal: use random and mixed-effects models to leverage the fact that problem factors are random

What additional modeling assumptions need to be verified?

Do we need to validate the randomness of this factor?

What additional information can be inferred from these models?

- E.g. understand how variances change with problem size



Sandia National Laboratories

# A Motivating Example?

I tried to use a mixed-effects model, but the data was significantly non-normal



Sandia National Laboratories

# *Related Statistics Problems*

- DOE for software testing
  - Goal: explore many algorithmic combinations to test a large fraction of the optimization software
  - Note: cannot reduce size of experimental design a priori

- Multivariate analysis
  - Goal: compare optimizers that have multiple performance criteria
  - Example: explore runtime/performance tradeoffs in heuristic optimizers with stopping rules

- Comparison of multi-criteria optimizers
  - Goal: compare the Pareto sets generated by multicriteria optimizers
  - Note: methods for comparing Pareto sets is an open research area



Sandia National Laboratories




Sandia National Laboratories