SAND2011-3329P

Derivative-Free Hybrid Optimization Approaches to Simulation-Based Problems

Genetha Gray
Sandia National Labs, Livermore

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Agency under contract DE-AC04-94AL85000.

National Nuclear Security Administration

Sandia National Laboratories

Collaborators/Contributors

- Katie Fowler, Clarkson University
- Josh Griffin, SAS
- Thomas Hemker, TU-Darmstadt
- Matt Parno, MIT
- Herbie Lee, UC Santa Cruz
- Bobby Gramacy, Matt Taddy, University of Chicago
- Shawn Mattot, University of Waterloo
- Monica Martinez-Canales, Intel
- Paul Boggs, Patty Hough, Joe Ruthruff, SNL
- Cheryl Lam, Brian Owens, Chuck Hembree, SNL

Funding thanks to:

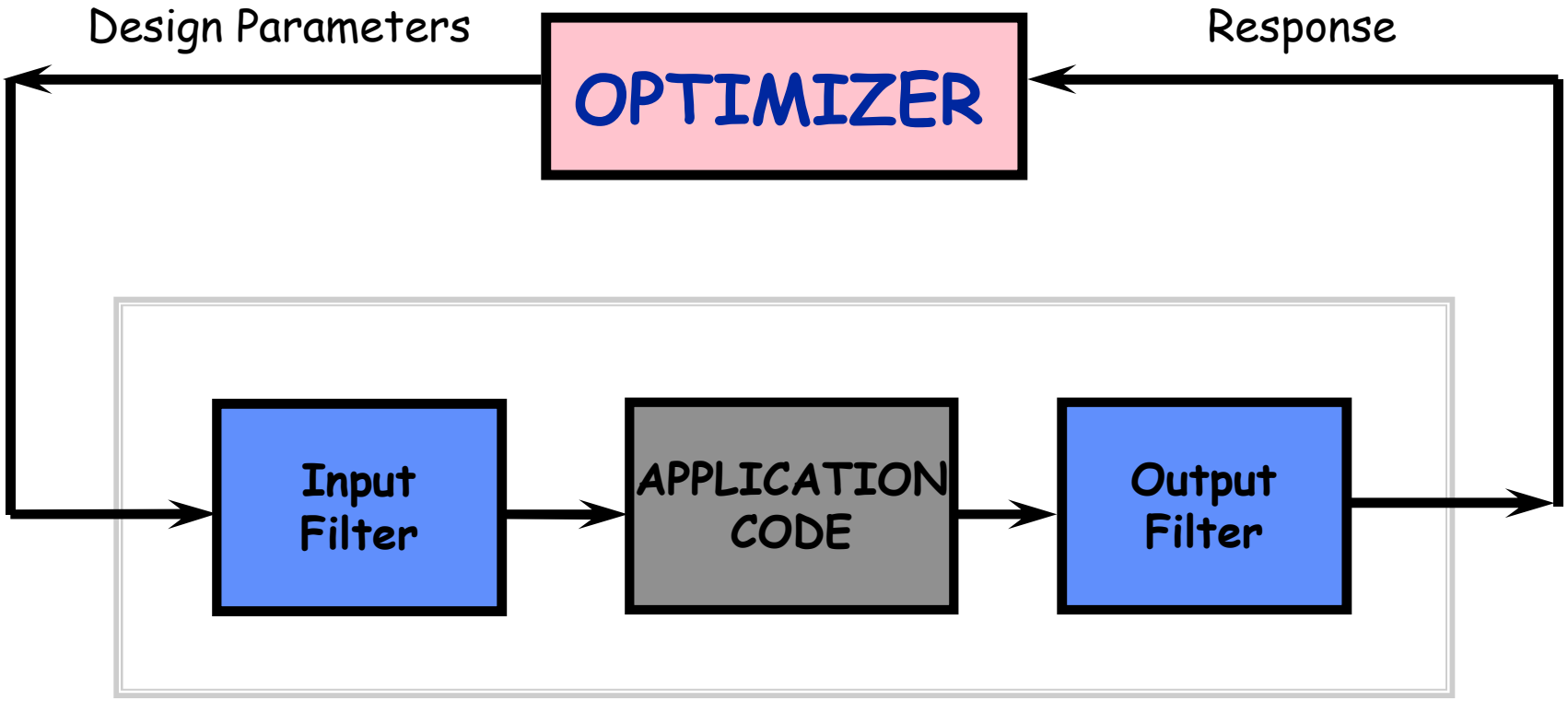
- American Institute of Mathematics (AIM)
- Sandia ASC & LDRD programs

Sandia National Laboratories

Outline

1. Two Motivating Examples
    1. Well Field Design (UNC6)
    2. Hydraulic Capture (HC)
2. Asynchronous Parallel Pattern Search (APPS)
3. Hybrid algorithms
    1. MFO
    2. APPS-TGP
    3. EAGLS
4. Results
5. Future Work

Simulation-Based Optimization



Evaluate $f(x), \quad g(x)$

Hydrological Applications

- **Objective Function: discontinuous, nonlinear, nonconvex, many local minima**
- **Evaluating the objective function requires the solution of an approximate numerical model**
  - Expensive
  - Low resolution of physical phenomena
  - Legacy code
- **Models of natural systems**
  - Not closed
  - Stochastic parameters
- **Physical aspects result in mathematical concerns**

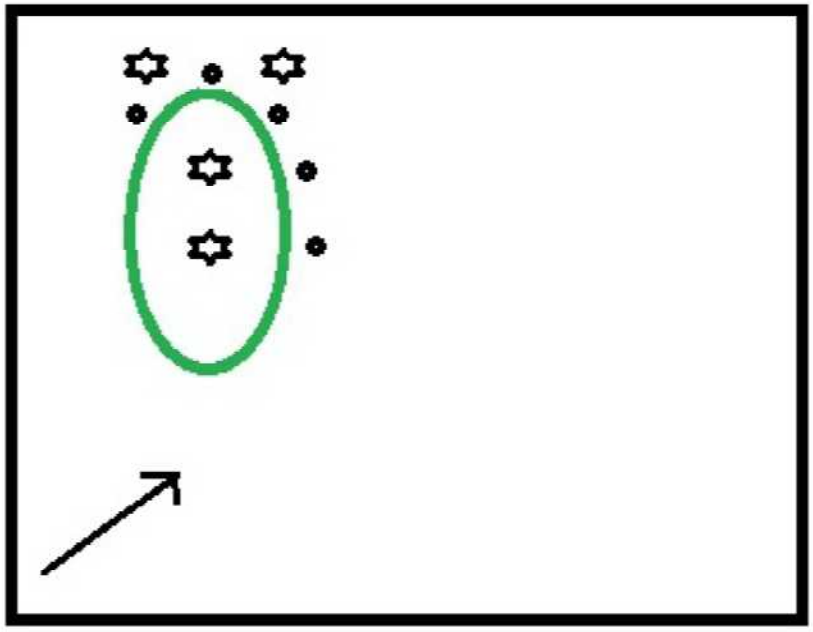
The Community Problems

- **Description**
  1. A suite of optimization problems and solutions where problems are implemented with standard simulators. (Mayer, Kelley, Miller)
  2. 4 Applications, 5 Domains, 2 Objective Functions, 6 Constraints, 30 Design Applications
- **Usefulness in testing of optimization methods**
  1. Use the complexities introduced by simulators to put the algorithm through its paces, uncover bugs, discover what “features” need to be added, etc.
  2. Hidden constraints make the problems challenging.
  3. Introduce a community to our algorithm by demonstrating it on a problem they care about or can relate to

Remediation

- **Hydraulic Capture (HC)**

  In a homogeneous, unconfined aquifer, place n ≤ 4 wells and determine their pumping rates to prevent a contaminant plume from spreading. Minimize the installation and operation costs.



Sandia National Laboratories

HC Objective Function

discontinuous

$$J(u) = \underbrace{\sum_{i=1}^{n} c_0 d_i^{b_0} + \sum_{i=1}^{n^e} c_1 \left| Q_i^m \right|^{b_1} (z_{gs} + h^{\min})^{b2}}_{\text{installation cost}} + \underbrace{\int_0^{t_f} \left(\sum_{i=1}^{n^e} c_2 Q_i (h_i - z_{gs}) + \sum_{i=n^e+1}^{n} c_3 Q_i \right) dt}_{\text{operational cost}}$$

Black-box

Evaluation of $J(u)$ requires $h_i \Rightarrow$ flow simulation

Sandia National Laboratories

HC Constraints

$$\text{Well capacity } -0.0064 \le Q_i \le 0.0064 \ m^3/s$$

$$\text{Net pumping rate } \sum_{i=1}^{n} Q_i \ge -0.0032 \ m^3/s$$

$$\text{Bounds on hydraulic heads } h^{\min} \le h_i \le h^{\max} m$$

Don't install a useless well:

$$|Q_i| < 10^{-6} \ m^3/s \Rightarrow \text{Remove well from design}$$

HC Formulations

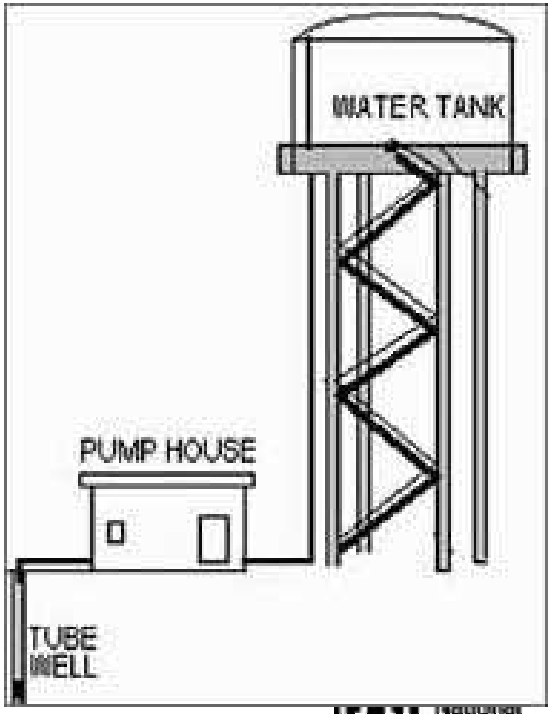
- **Continuous**
  - For each well, there are 3 associated variables (12 variables)
  - Two position; Pumping rate (12 continuous)
  - 2 linear constraints
  - Up to 8 nonlinear constraints (2 per well)
- **Mixed Integer**
  - For each well, there are 4 associated variables (16 variables)
  - 0,1: indicates well is on or off (4 discrete)
  - Two position variables, Pumping rate (12 continuous)
  - 2 linear constraints
  - Up to 8 nonlinear constraints (2 per well)

Optimal Well Field Design

- **Well Field Design**

  Position n ≤ N wells and determine their pumping rates in a homogeneous aquifer A so that the total cost of installing and operating the wells needed to supply a given amount of water is minimized. The pumping rates are constant over time and the wells have a fixed depth.

- **UNC6:** Let A be unconfined and N = 6



Formulation

- **Black Box (MODFLOW)**
- **Continuous**

  For each well, there are 3 associated variables & 2*n + 2 constraints
  - 2 positional & 1 pumping rate
  - 2 linear constraints
  - 2 nonlinear constraints per well
- **Mixed Integer**

  For each well, there are 4 associated variables & 2*n + 2 constraints
  - 0,1: indicates well is on or off (discrete)
  - 2 position variable & 1 pumping rate (continuous)
  - 2 linear constraints
  - 2 nonlinear constraints per well

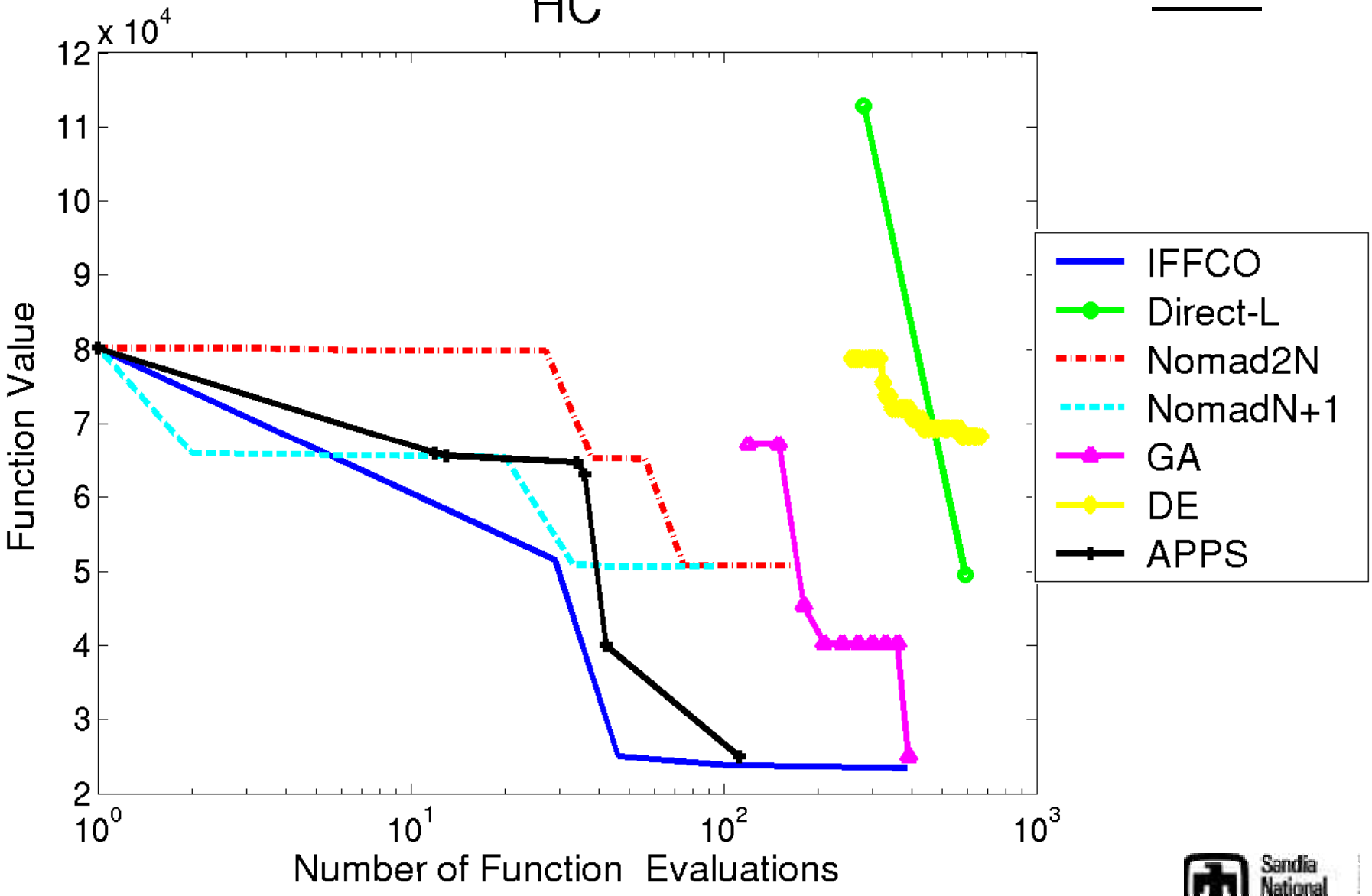
The Shootout

- Fowler et al, 2008
- Continuous Formulation
- Feasible initial iterate supplied
- 50 x 50 grid
- Each algorithm was applied by the algorithm creators
- Allowed a maximum of 3 changes to default parameters

Derivative-Free Methods

- **Asynchronous parallel pattern search (APPS)**
  - Direct search
  - Predetermined pattern of points
  - Reduces processor latency
  - APPSPACK (Kolda et al)
- **Dividing Rectangles (DIRECT)**
  - Divide space for deterministic sampling
  - Gablonsky, Kelley et al
- **NOMAD**
  - Generalized pattern search using Latin Hypercube
  - Two versions
  - Abramson, Audet, Dennis, et al
- **Genetic Algorithm (GA)**
  - Population based global search
  - Barrier or penalty approach to include constraints
  - NSGA-II (Deb & Agrawal)
- **Implicit Filtering**
  - Projected quasi-Newton method
  - Uses a large stencil to account for noise
  - IFFCO (Kelley)
- **Design Explore (DE)**
  - Generalized pattern search on model problem
  - Booker et al (Boeing)

HC Shootout Results



HC Results

Optimizer	Cost ($)	Solution (# of wells)
IFFCO	$23,421	1
DIRECT	$49,549	2
Nomad(2N)	$50,797	2
Nomad(N+1)	$50,574	2
APPS	$25,018	1
APPS (MFO)	$22,428	1
GA (integer)	$24,934	1
GA (real)	$54,973	2
Surrogate (MINLP)	$23,491	1

- Starting point
  - 4 active wells
  - Cost of $78,587
- FBHC formulation
- Continuous formulation
- MFO: both FBHC & TBCC
- References:
  - Fowler et al, *Adv. Water Res.*, 31(5), 2008
  - Hemker et al, Opt & Eng, 2007

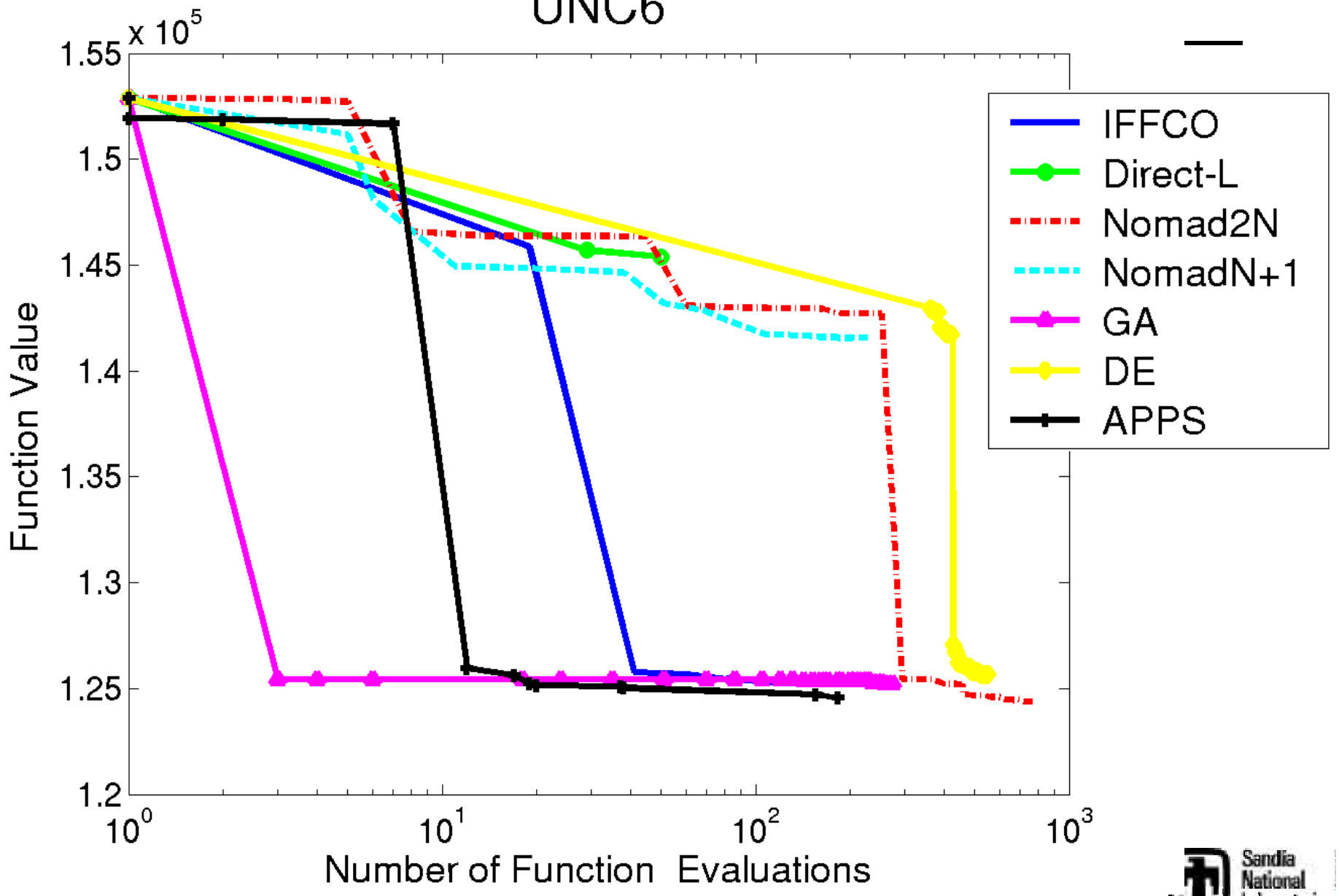
Sandia National Laboratories

HC Results: Initial Iterates

- Space-filling design of 100 points
- More than 1/3 infeasible → 62 feasible starting points
- Remaining points used as starting points for two of the successful local methods
- For all feasible starting points in this test, algorithms got "stuck"
- Continuous formulation

	APPS	IFFCO
Stuck at starting point	11	5
Stuck at a local min	51	57

UNC6 Results



Issues

- **What if an initial point is not supplied?**
  - **All methods that did not rely upon or were not supplied with an initial iterate failed.**
  - **Investigation of 100 other initial iterates: APPS, IFFCO, and NOMAD failed.**

  CONCLUSION/QUESTION: Find a method that is efficient despite the lack of initial iterate.

- **Can we use the mixed integer formulation?**
  - **GA is the only method that can handle integer variables.**
  - **GA seeded with a "good" initial iterate works well.**

  CONCLUSION/QUESTION: Improve the GA to eliminate need for initial iterates.

Hybrid Optimization Algorithms

- Take advantage of the benefits of more than one approach
- Overcome method-specific shortcomings
- Goal: simultaneously perform global and local searches (cheap and accurate!)
- Categorization scheme (Radl '06)
  1. Class of algorithms used to form hybrids,
  2. Level of hybridization- tightly or loosely coupled
  3. Order of execution- sequential or parallel
  4. Control strategy- integrative or collaborative
- Tailoring hybrids to address the characteristics of the problem

Considerations for Creating Hybrids

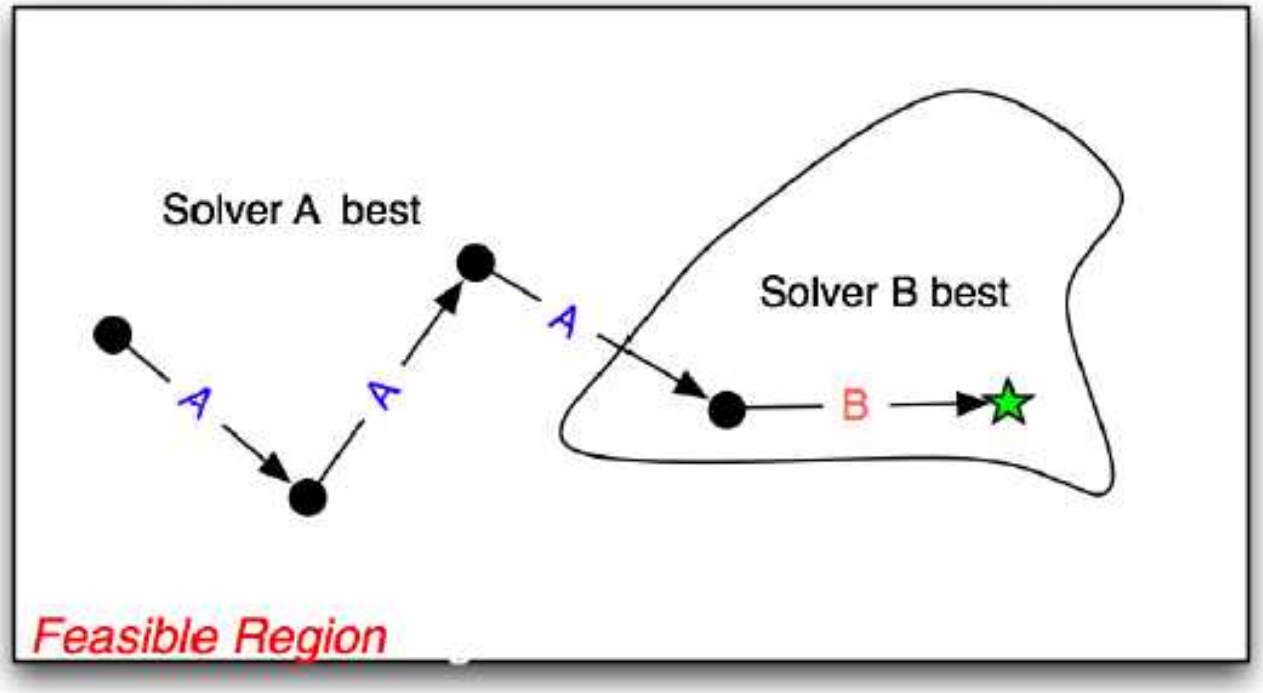
- What should we combine and why?
- How do we combine two or more methods to best exploit the advantages of each?
- What are the theoretical implications?
  - Convergence
  - Implementation
- What else can hybridization buy?
  - Less “tuning” time
  - Multiple perspectives of the domain
  - Additional computational data points
  - Robustness
  - Inclusion of uncertainty

Basic Sequential

1. Apply method A.
2. Use the solution of method A as a starting point for meth
3. App

- Mos
- Hyb

d A

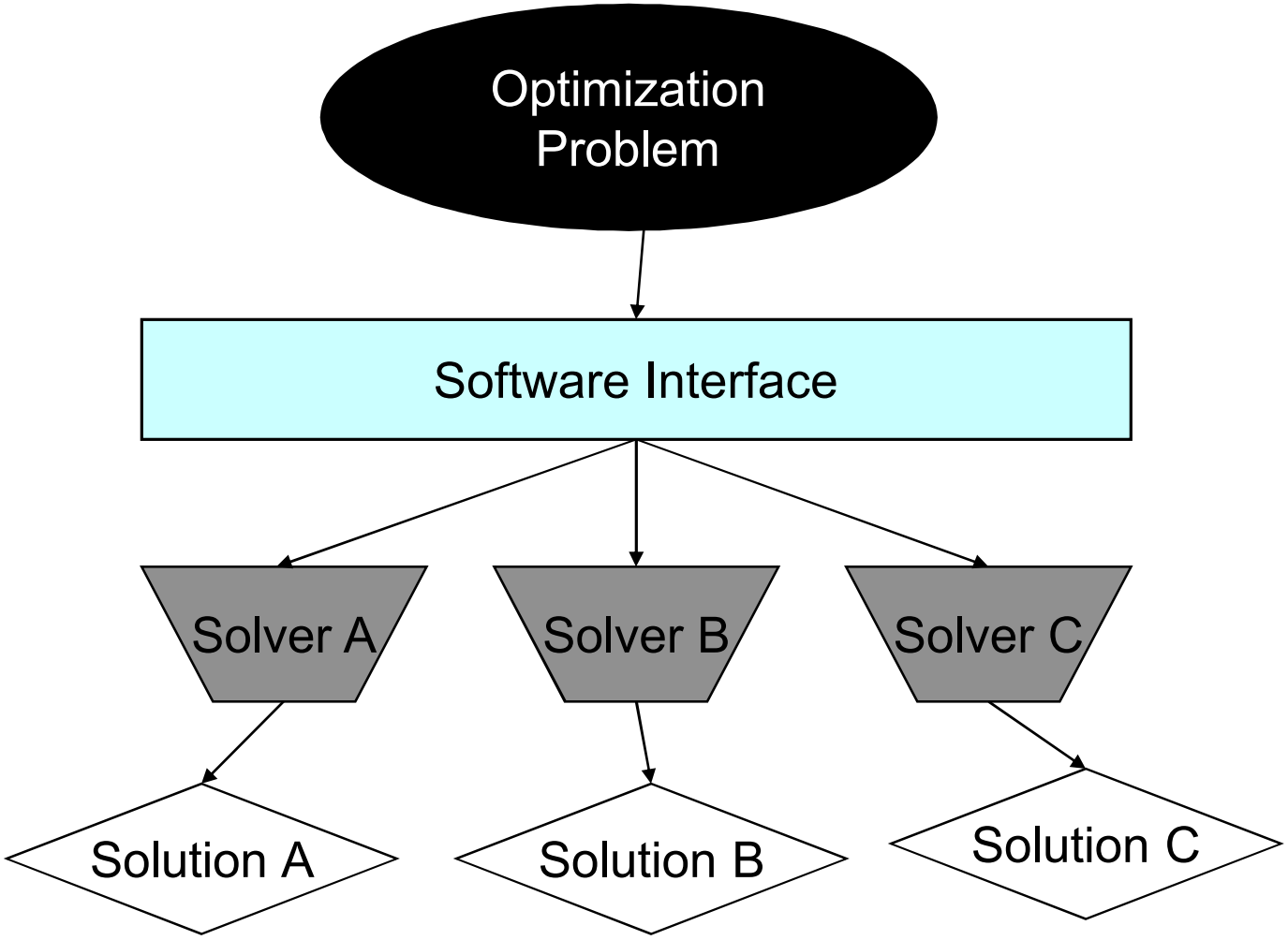


Multi-starts

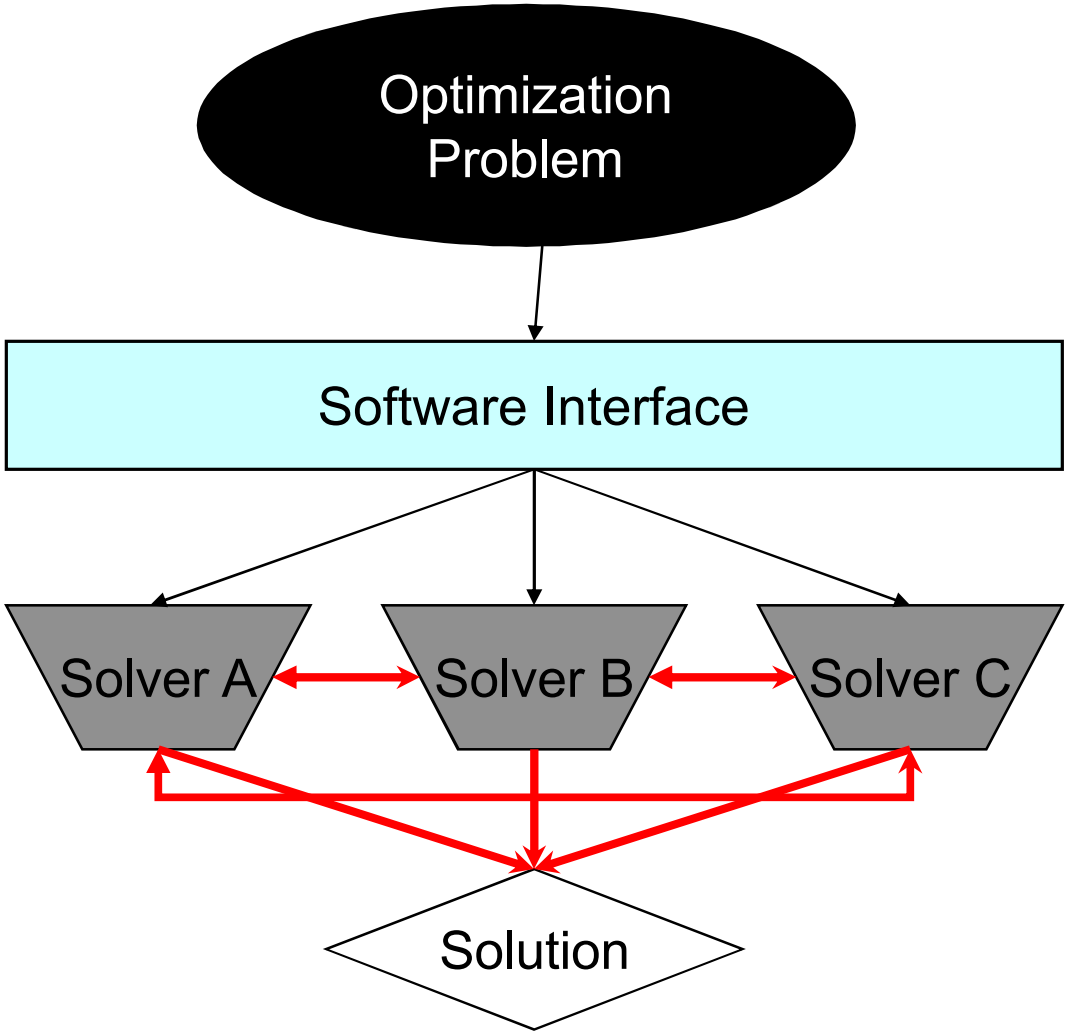
1. Select a set of starting points
2. In parallel, run your optimization method of choice on each algorithm
3. Compare results

- Comparing results:
  - Loosely coupled: Share results at predefined times
  - Cache sharing
  - Hybrid component: Decision making about which point to use

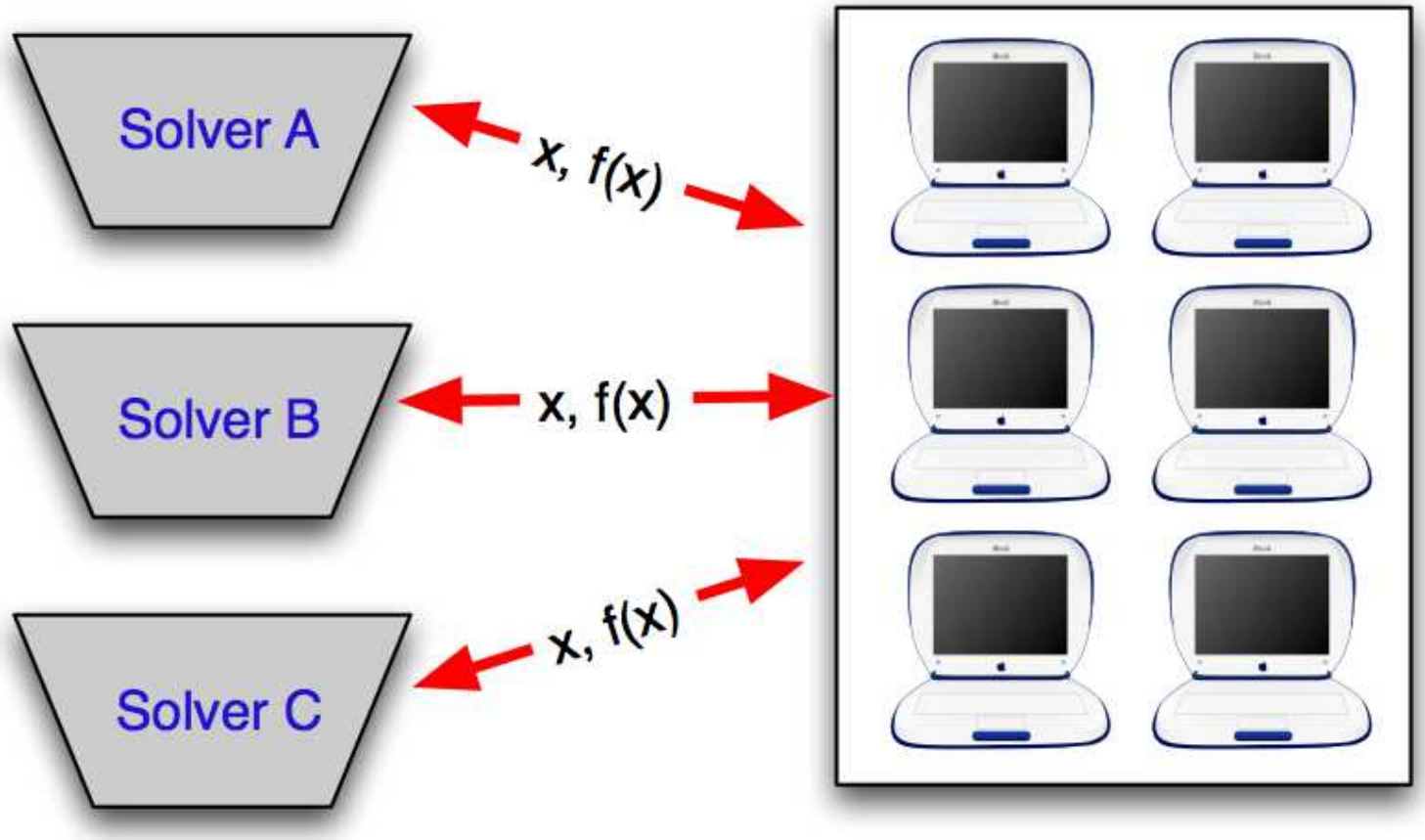
Maintain Solver Integrity



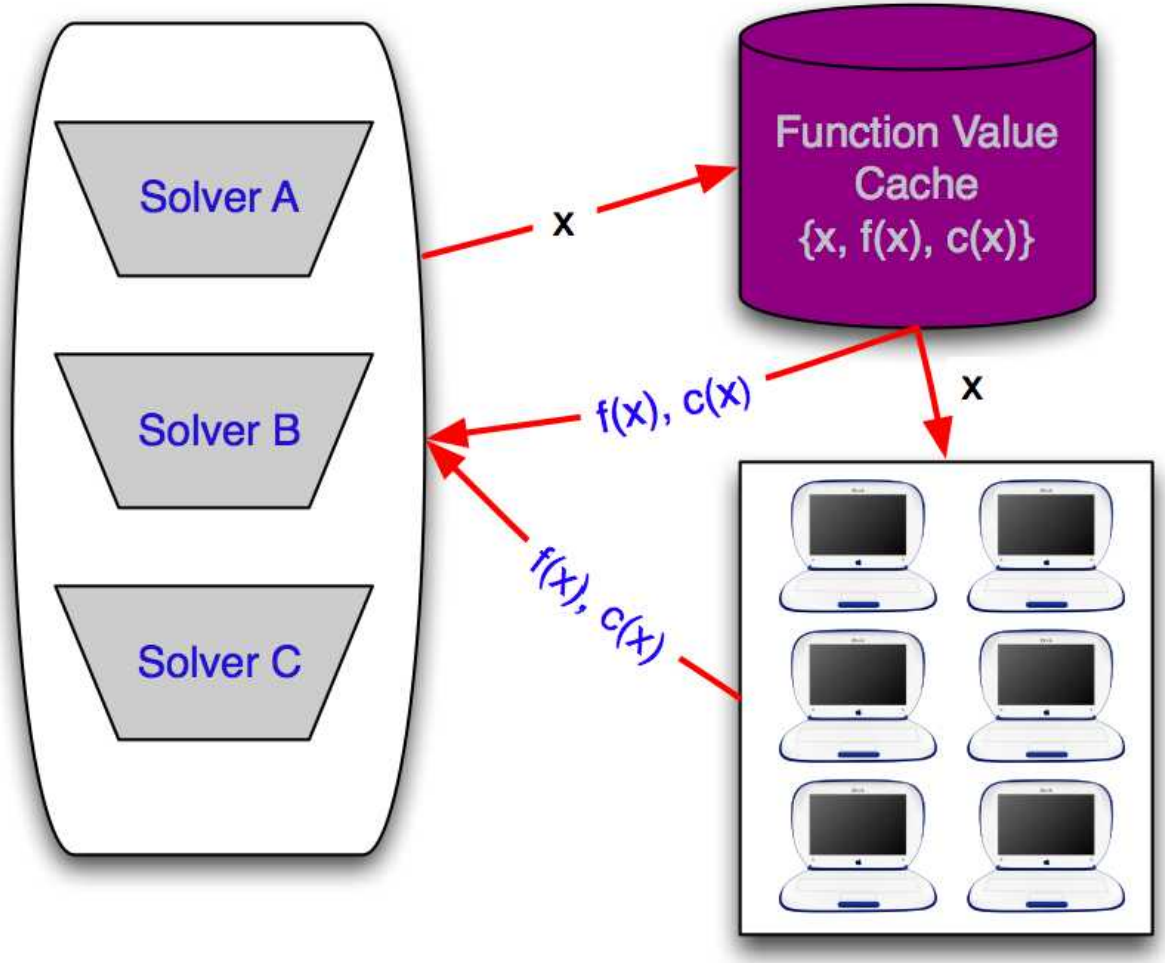
Allow Solver Interaction



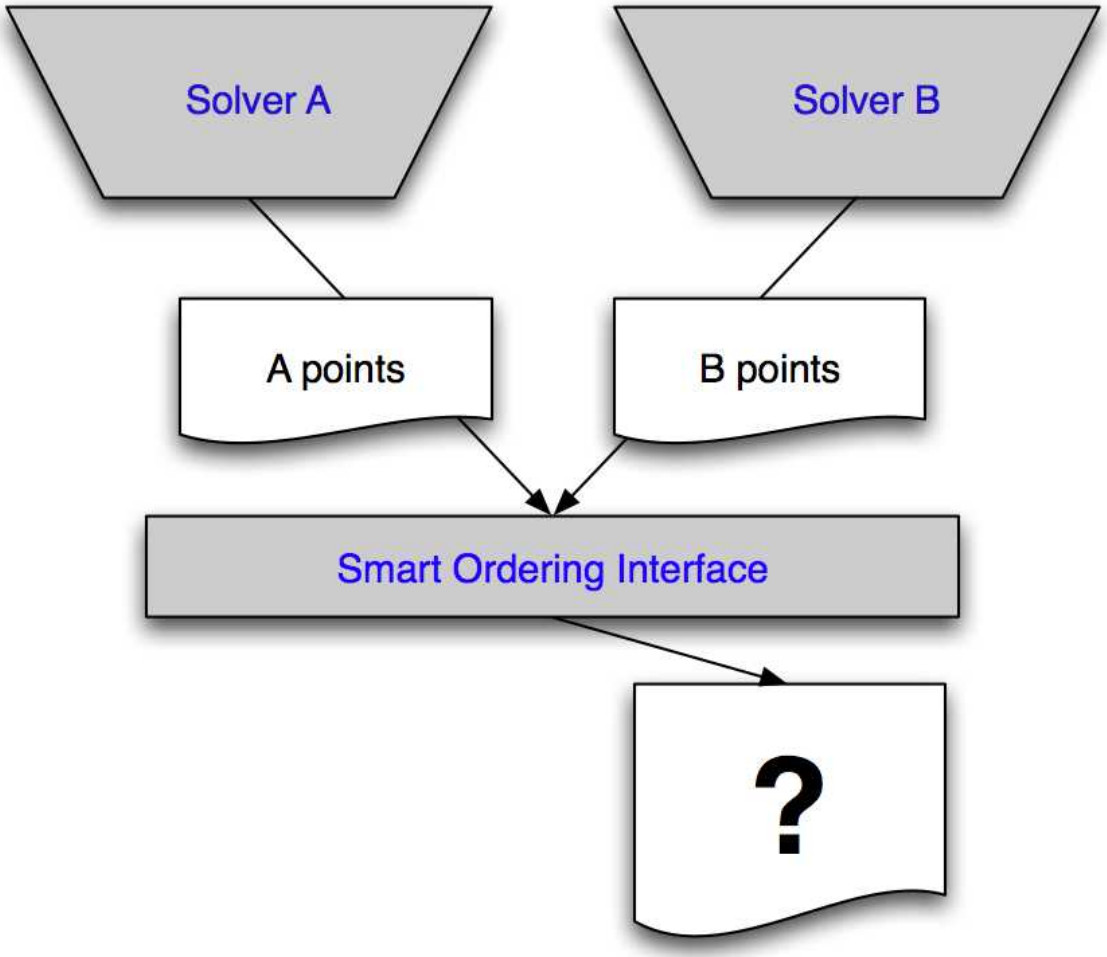
Share Evaluation Processors



Share Function Evaluation Results



Order Points

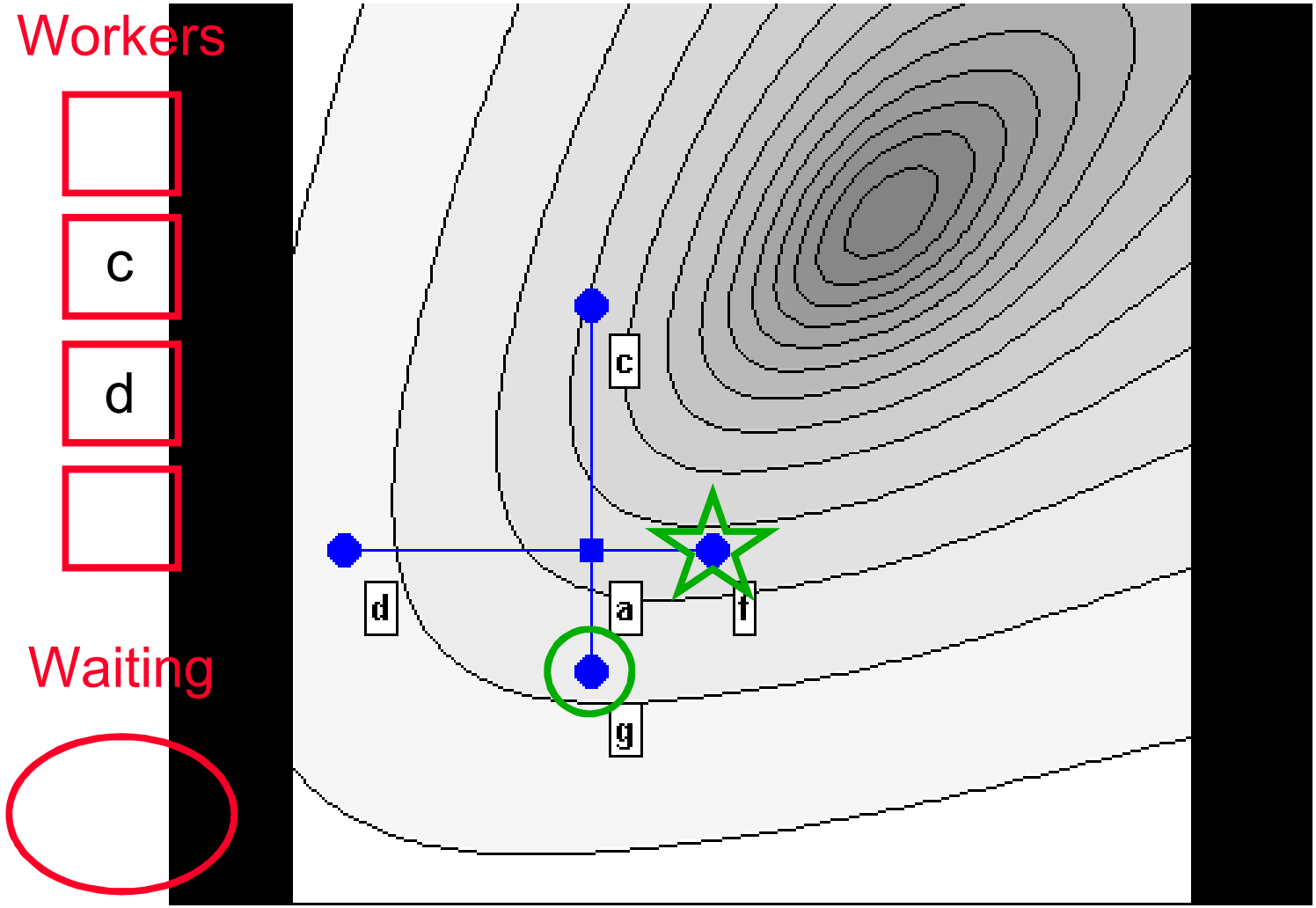


APPSPACK

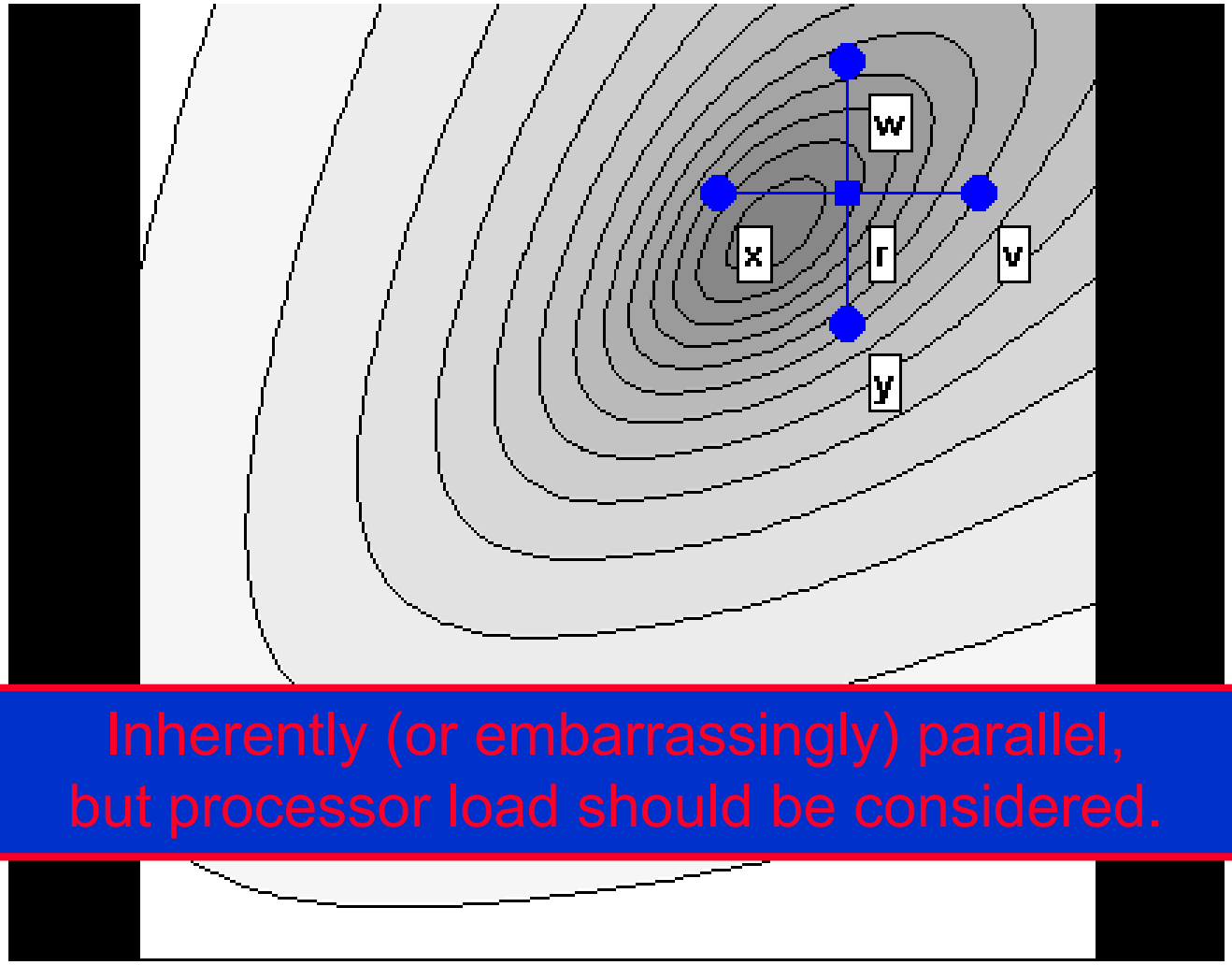
http://software.sandia.gov/appspack/

- Asynchronous parallel pattern search (can be synchronous)
- C++ implementation
  - Object-oriented design
  - User customizable interfaces
- MPI for message passing
  - Generic C++ wrapper
  - Manager-worker paradigm
- Software Tools
  - CVS repository
  - Autotools for configuration
  - Bugzilla for bug-tracking
- Freely available via GNU L-GPL Licensing

APPSPACK Example



Synchronous Pattern Search

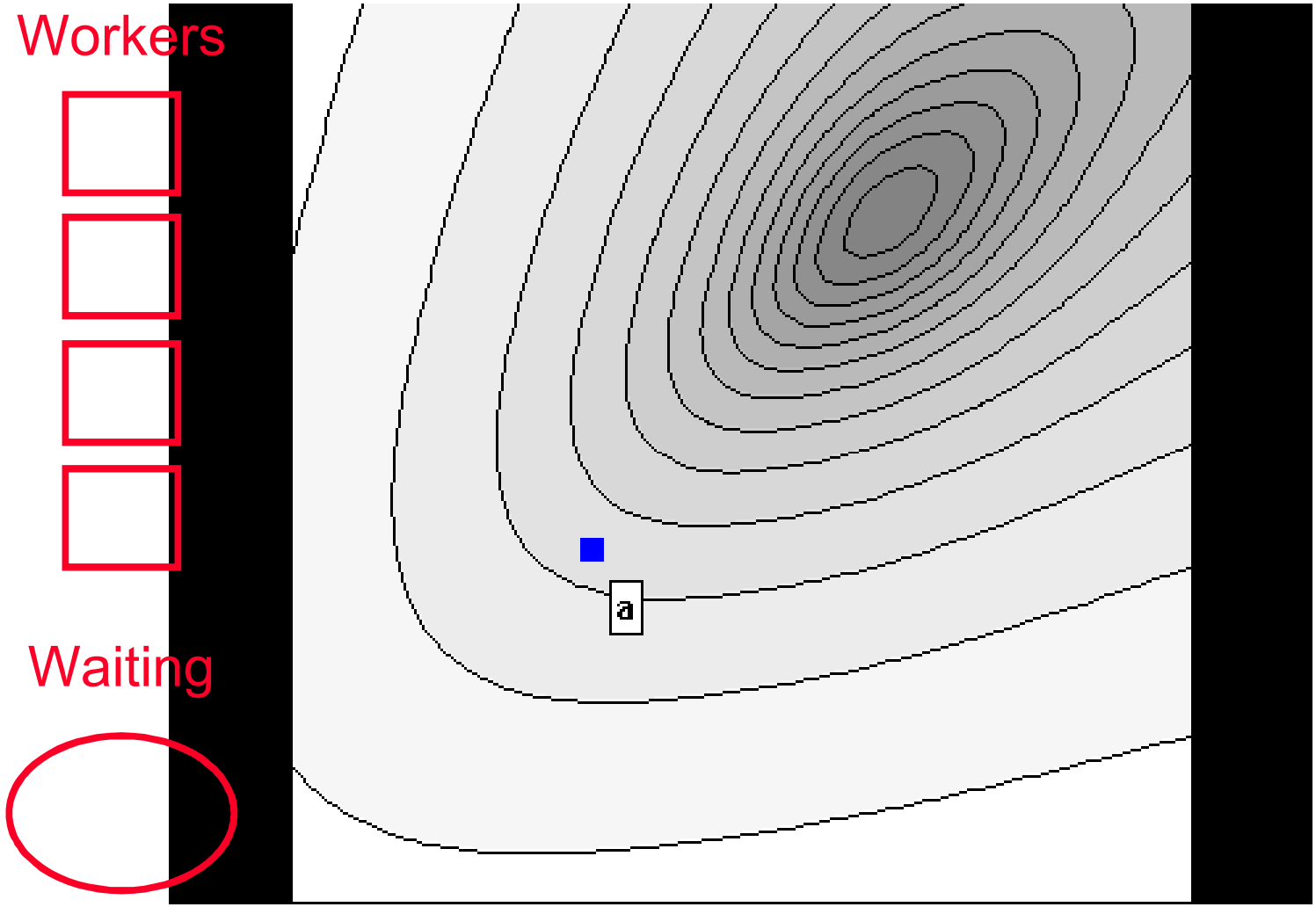


Inherently (or embarrassingly) parallel,
but processor load should be considered.

Processor Load Balance Considerations

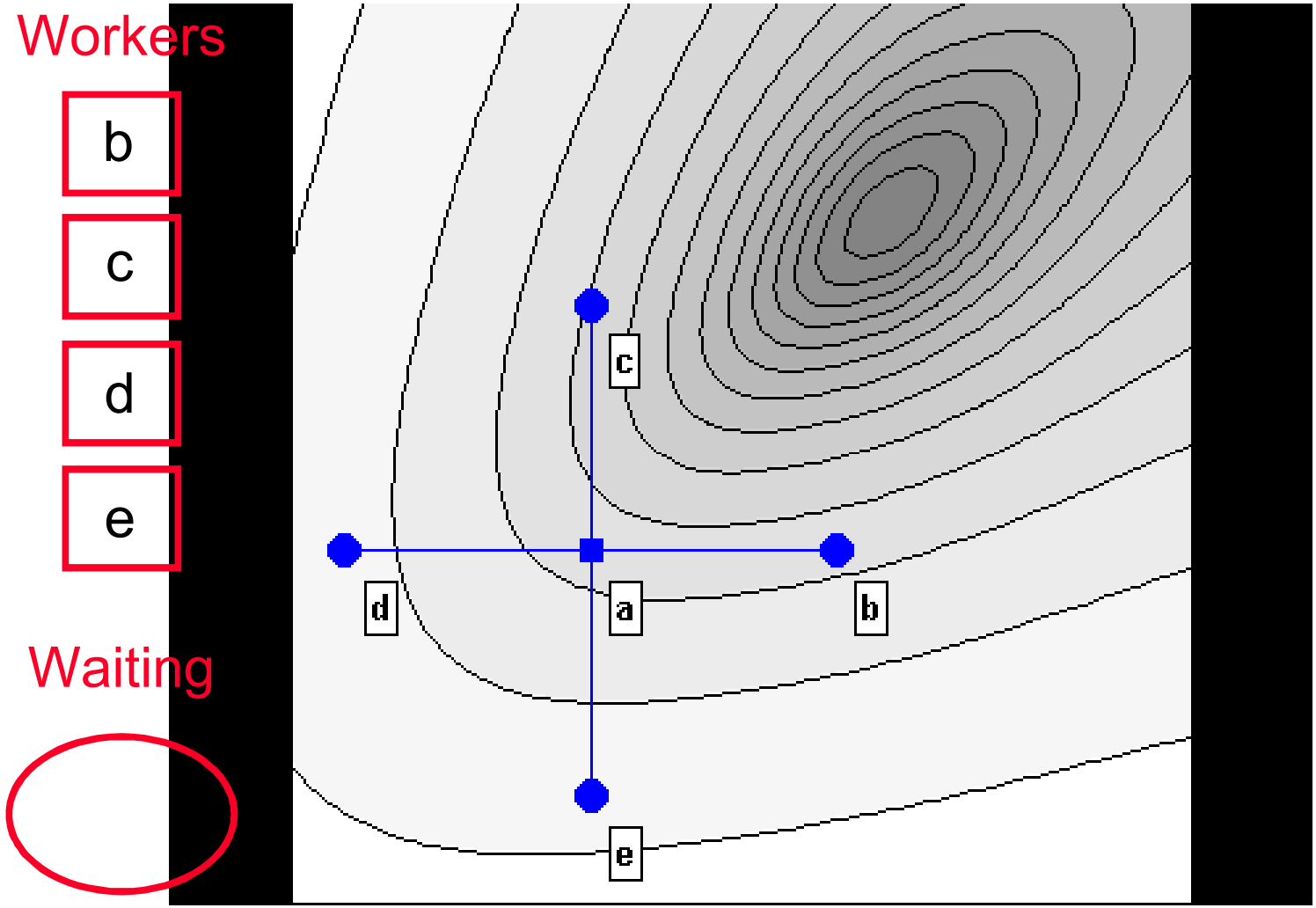
- The number of trial points can vary at each iteration.
  - Cached function values
  - Search patterns change
  - Constraints (infeasible trial points are not evaluated)
- Evaluation times can vary for each trial point.
  - Different processor characteristics
  - Effect of input on function
  - Function evaluation faults

APPSPACK Example

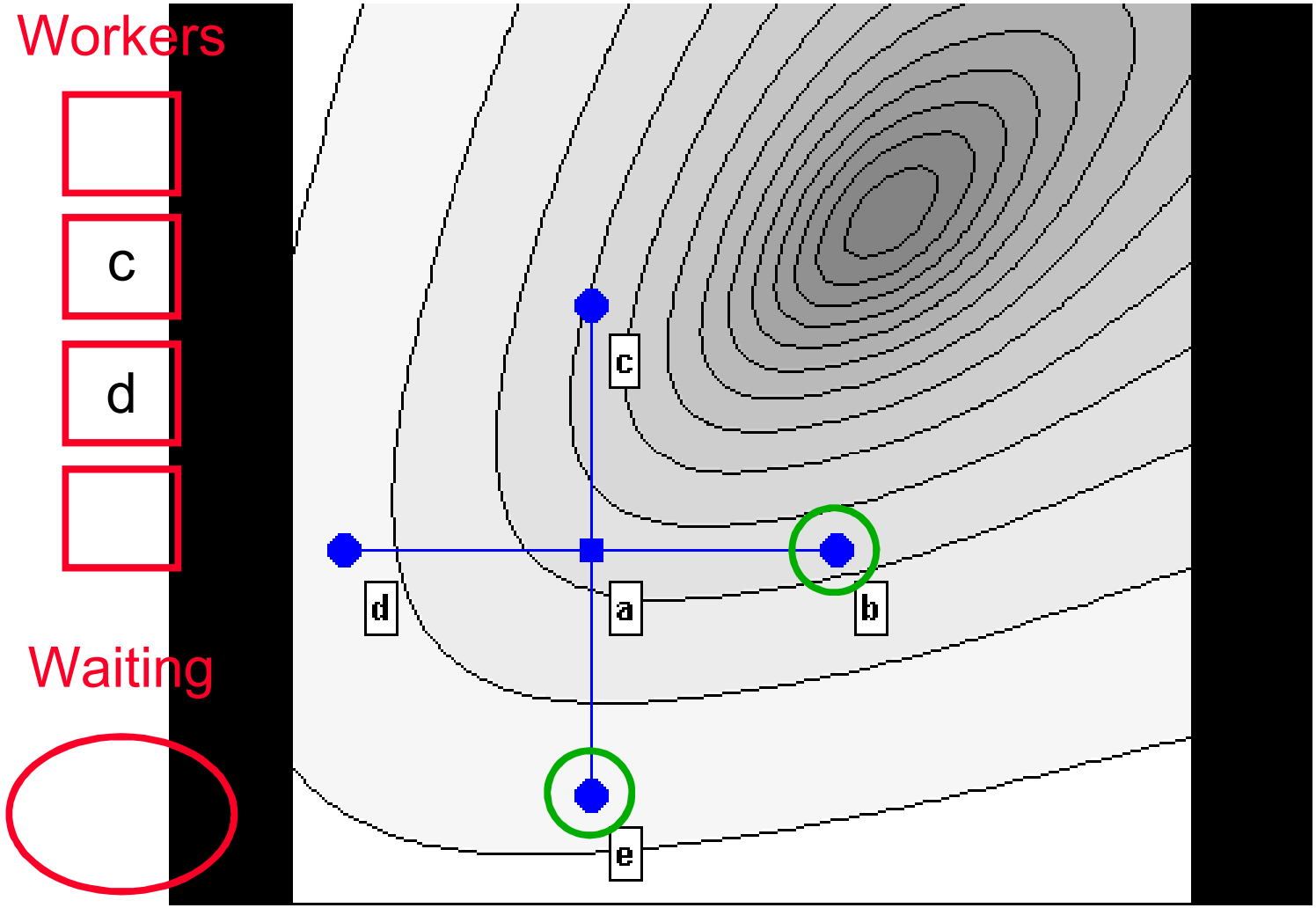


Sandia National Laboratories

APPSPACK Example

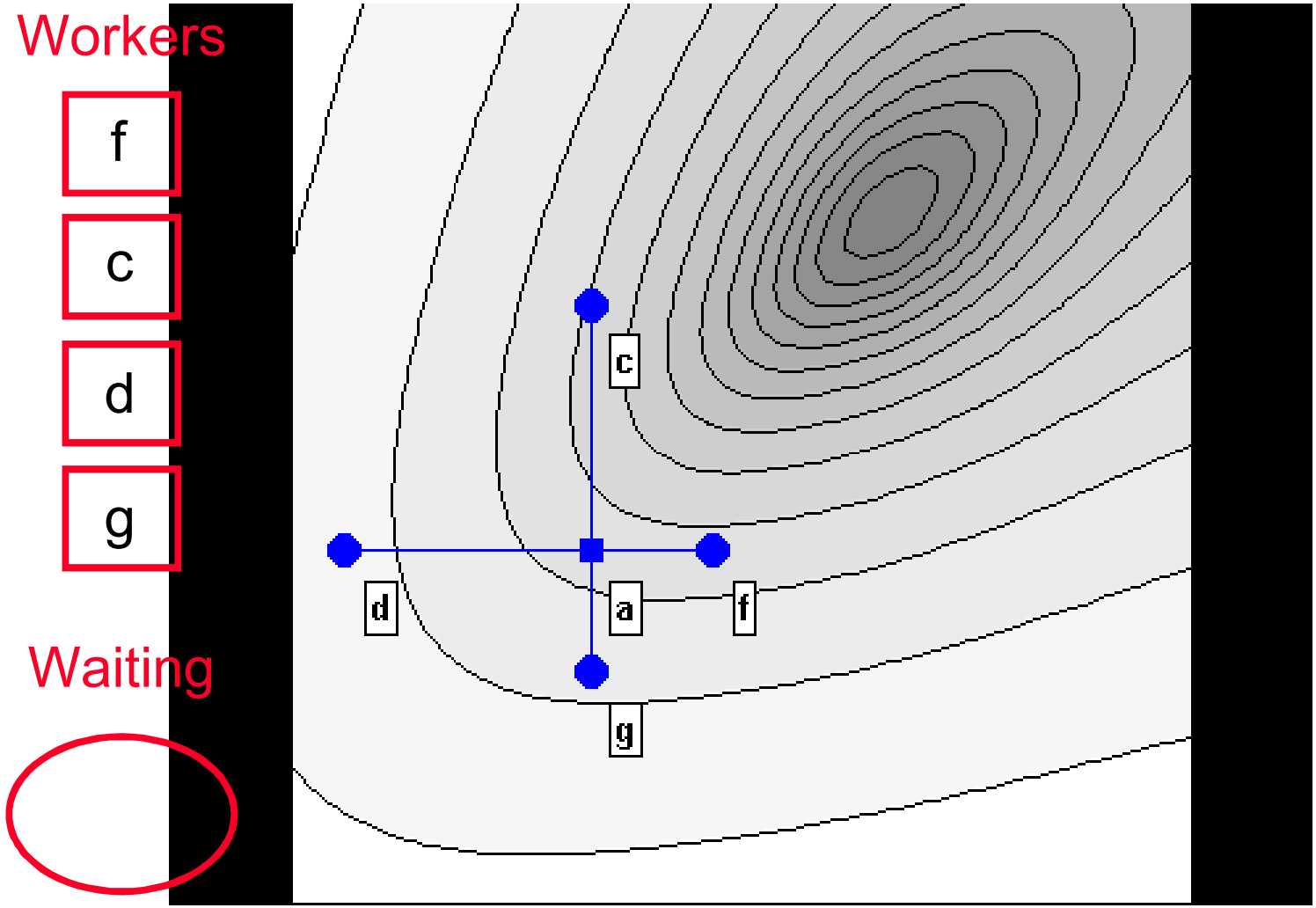


APPSPACK Example

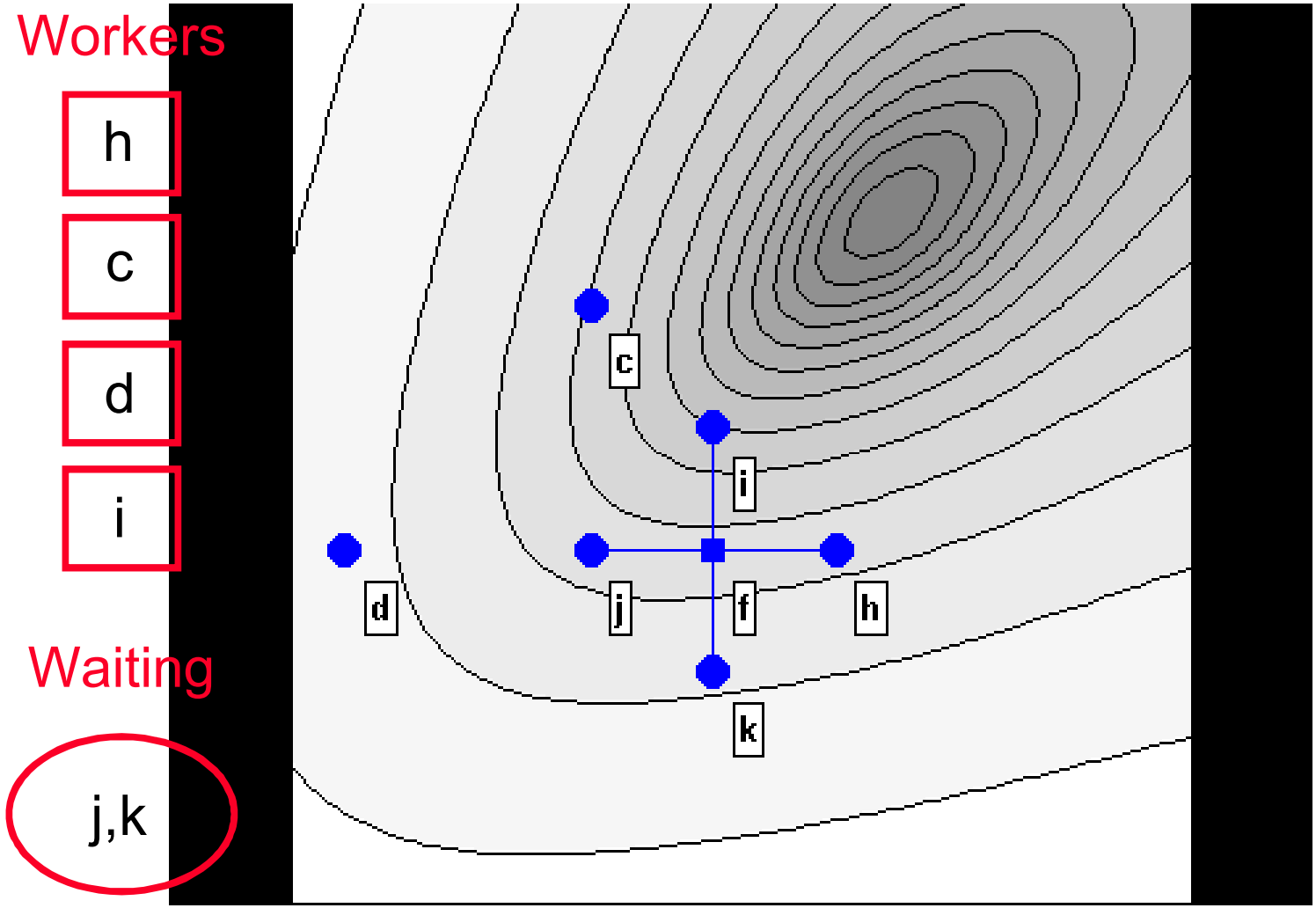


Sandia National Laboratories

APPSPACK Example

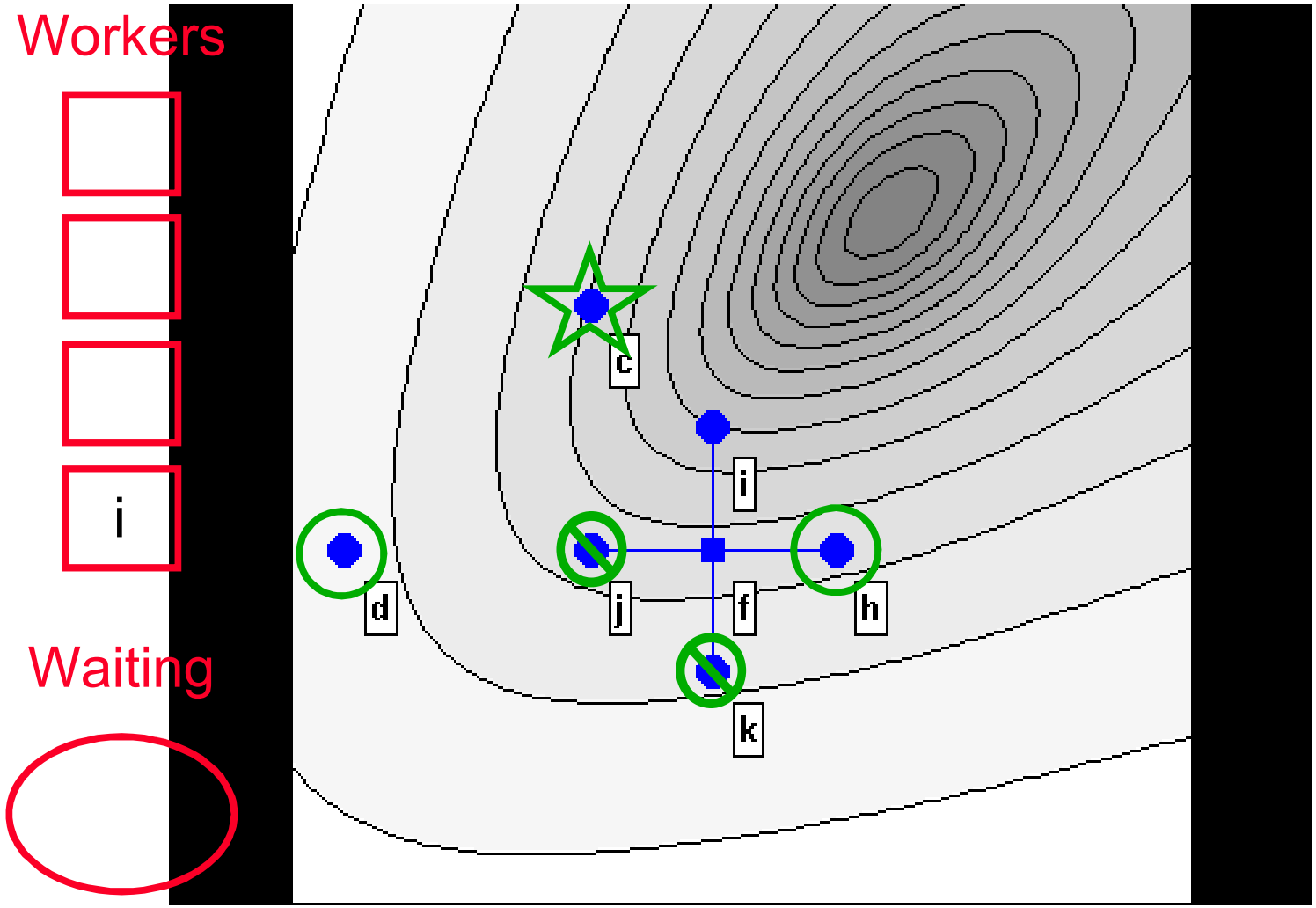


APPSPACK Example



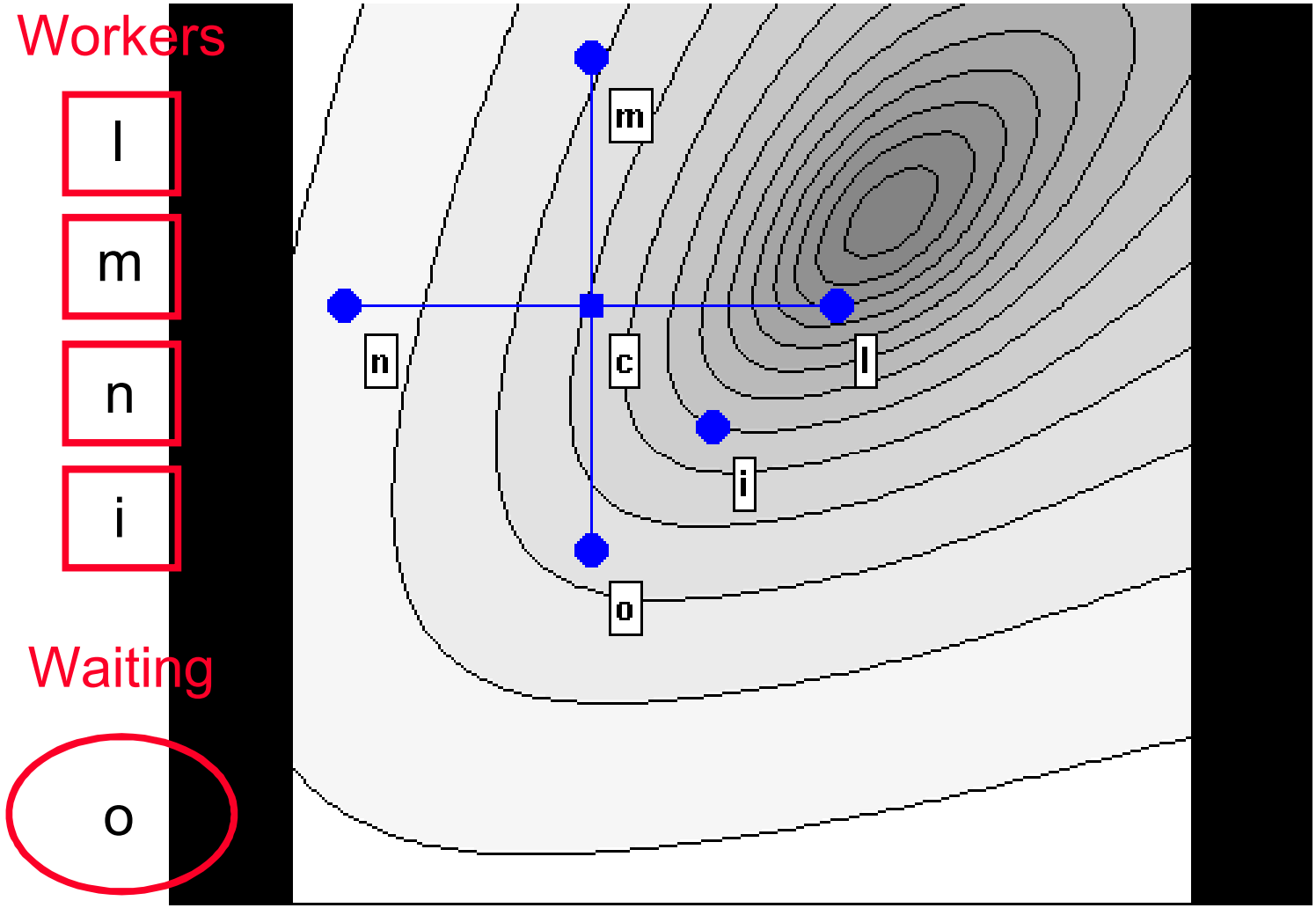
Sandia National Laboratories

APPSPACK Example

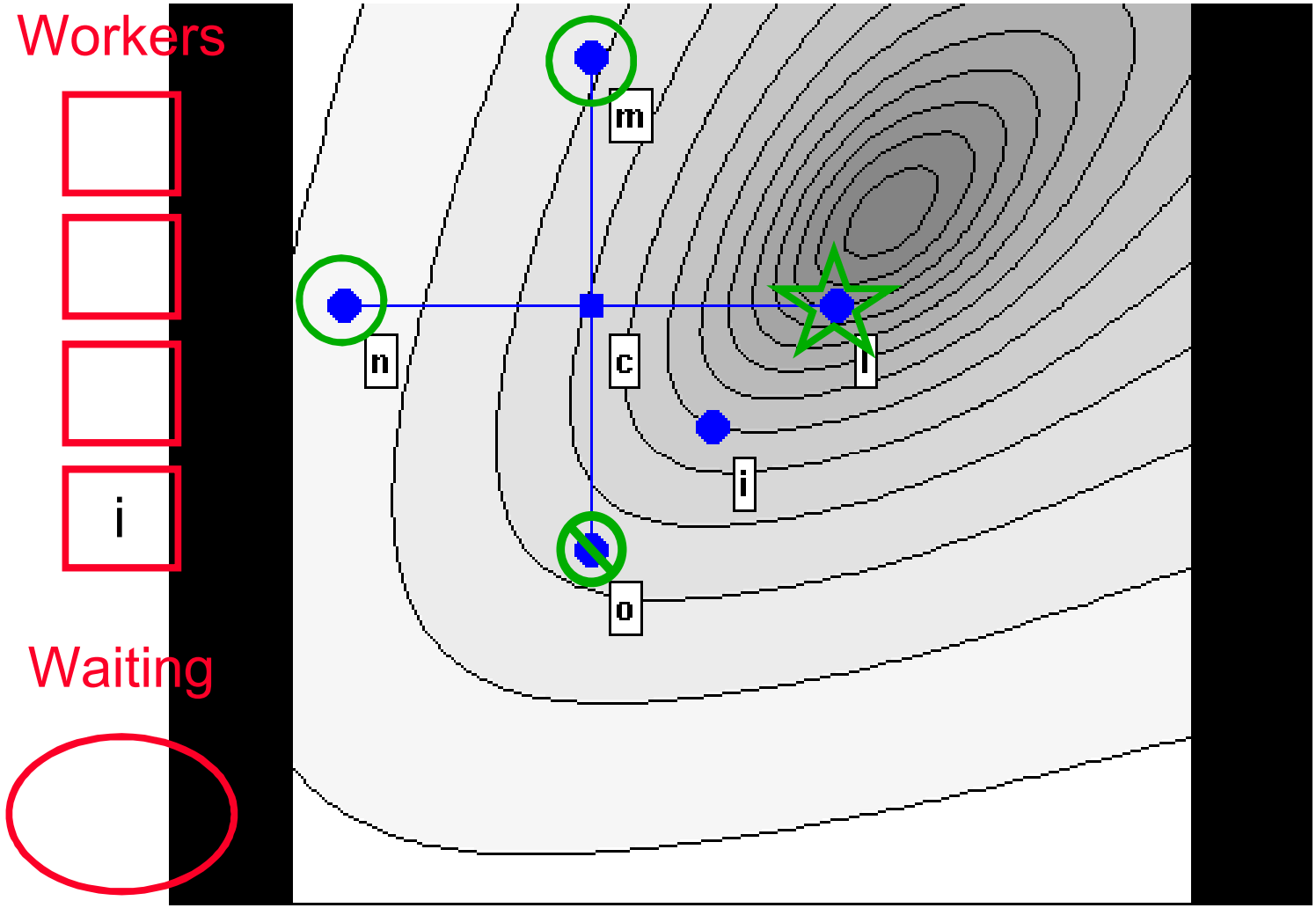


Sandia National Laboratories

APPSPACK Example

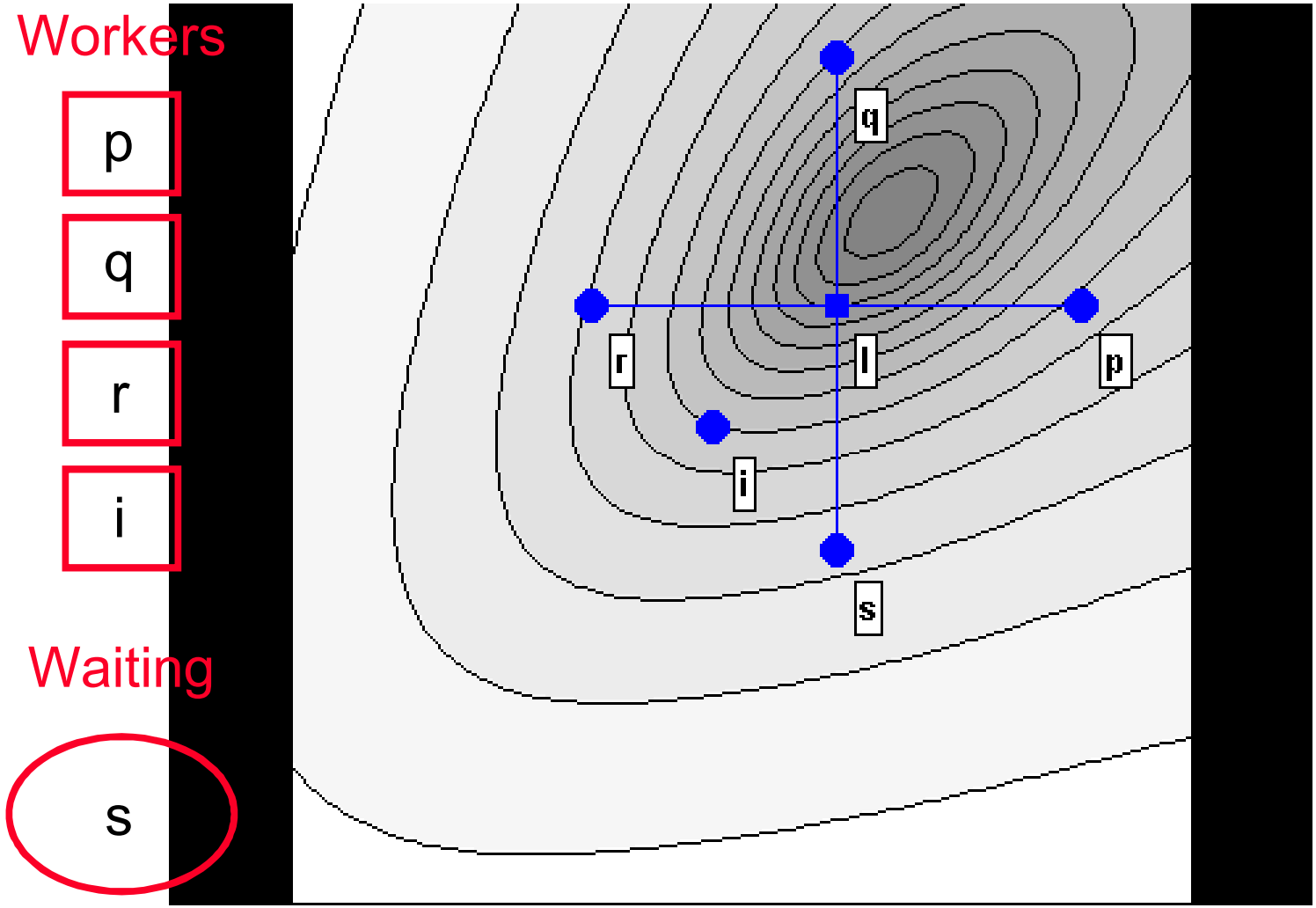


APPSPACK Example

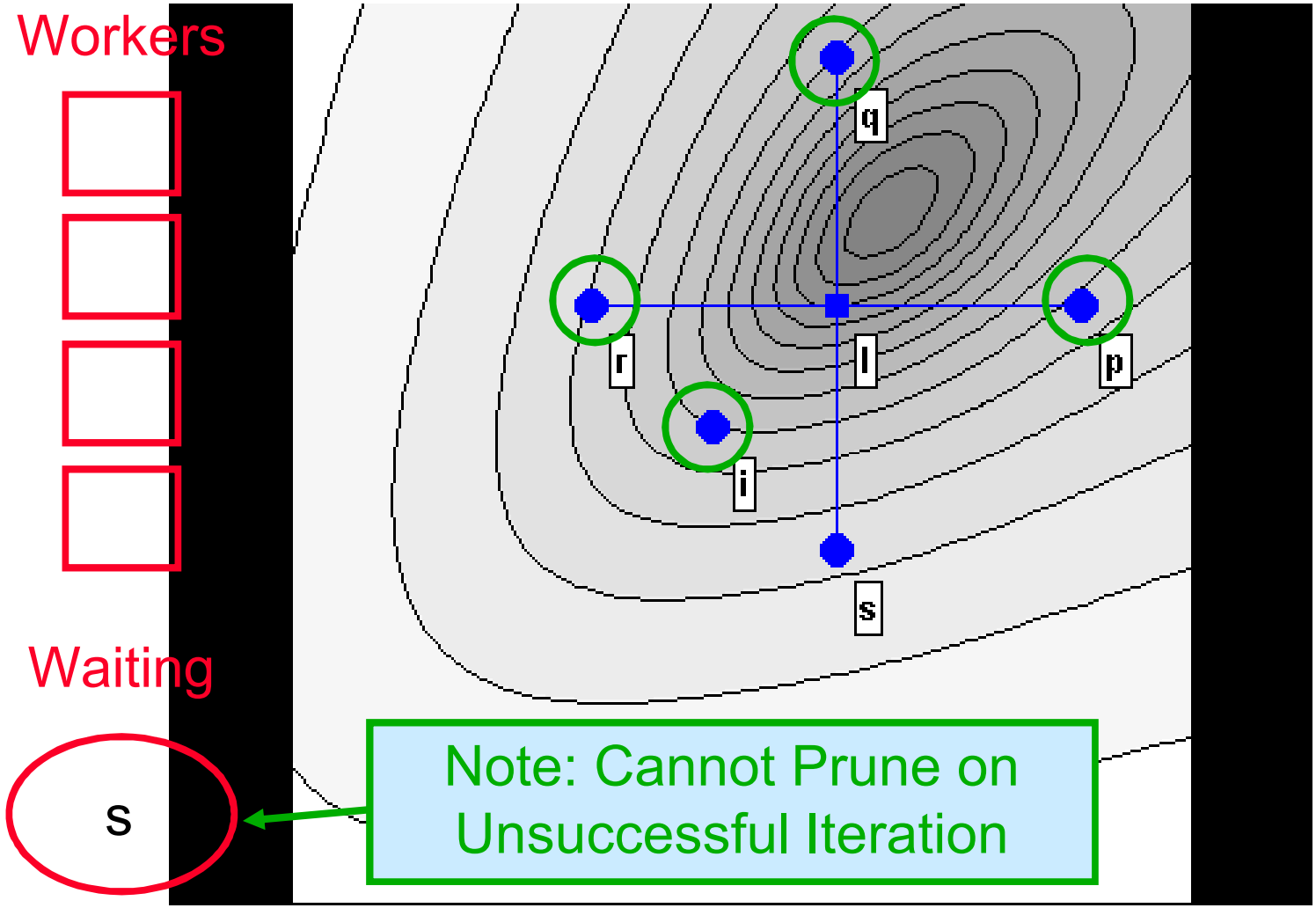


Sandia National Laboratories

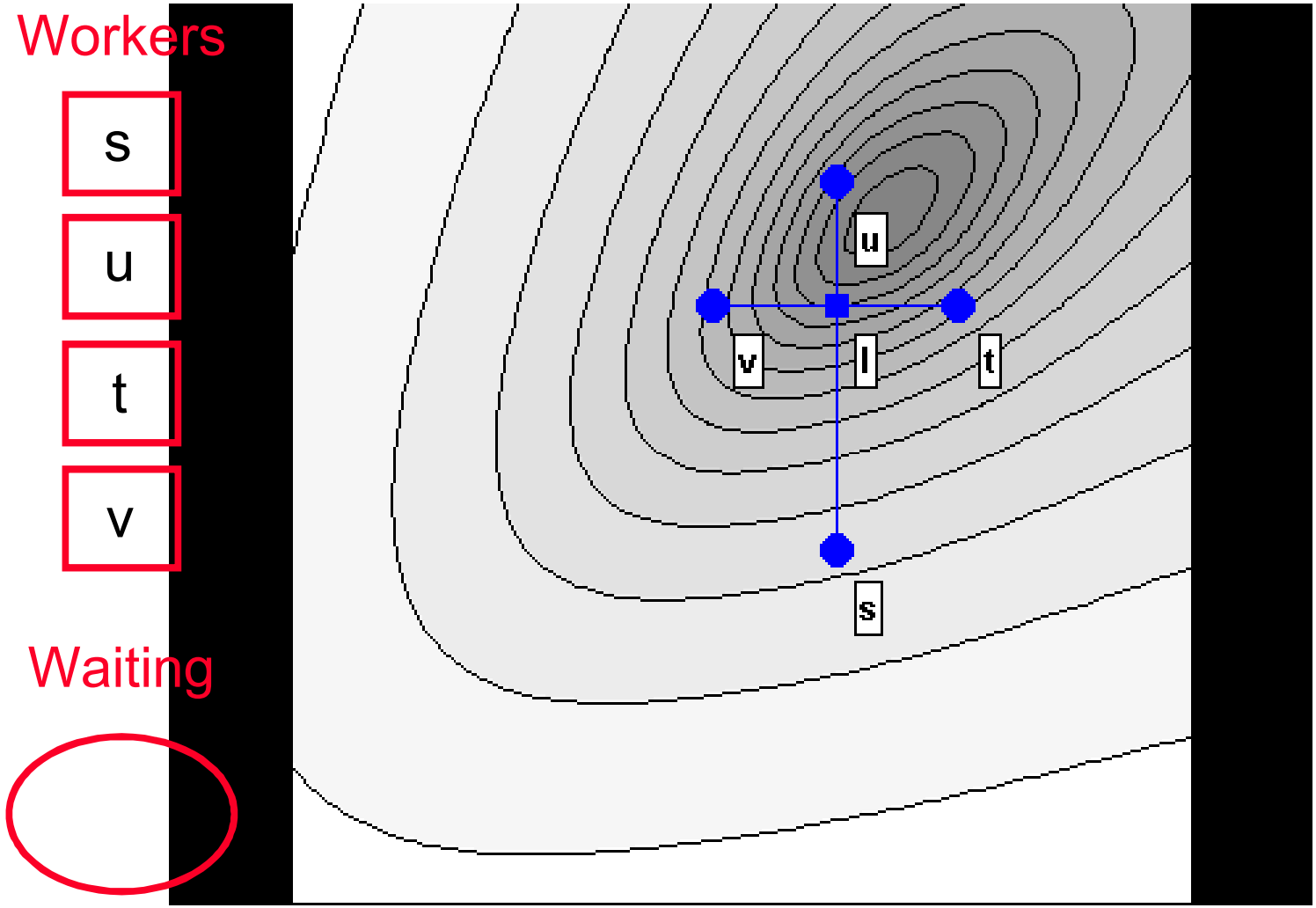
APPSPACK Example



APPSPACK Example



APPSPACK Example



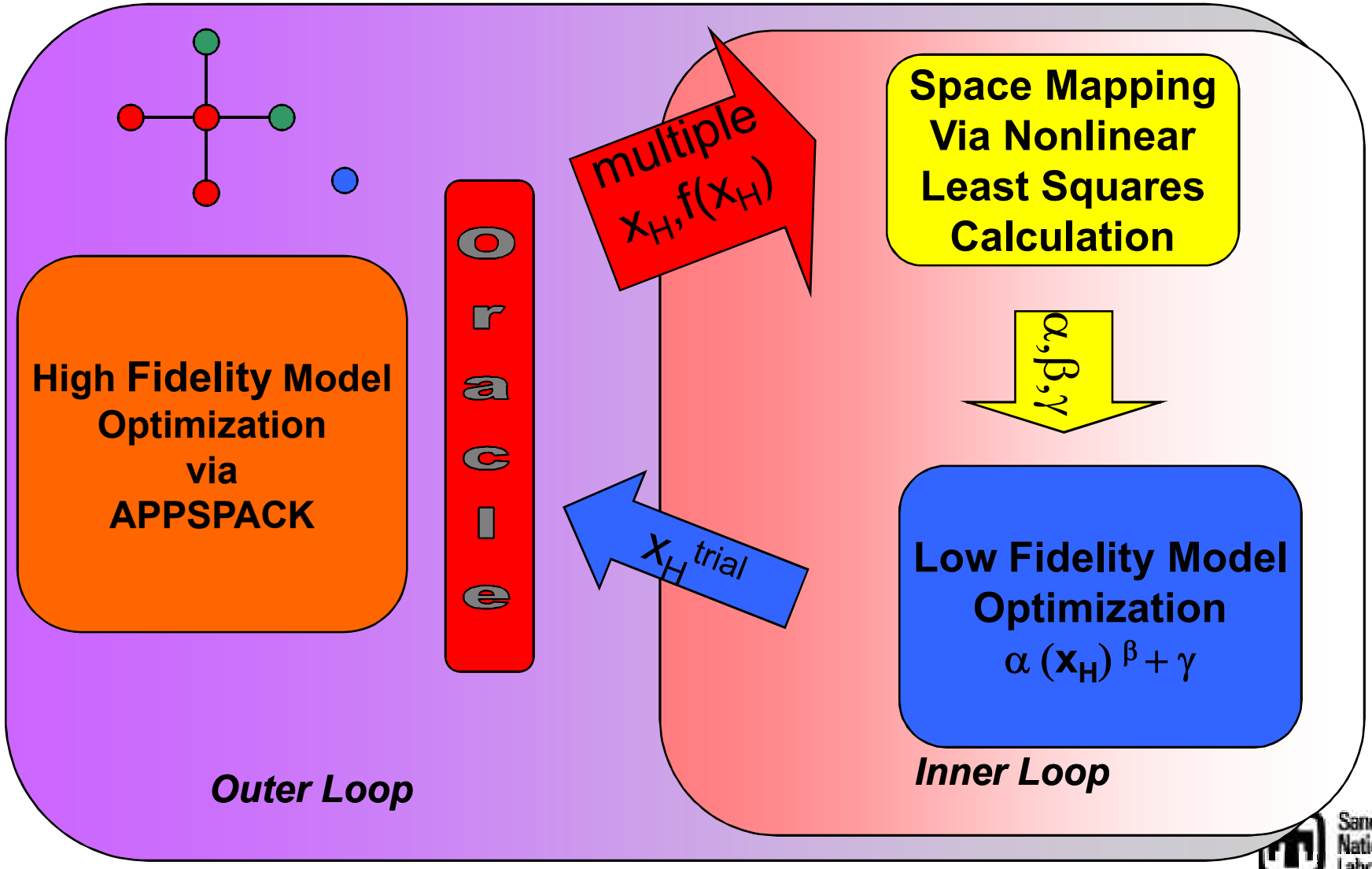
Oracle

- An oracle predicts points at which a decrease in the objective function might be observed.
- Analytically, an oracle can choose points by any finite process.
- Oracle points are used in addition to the points defined by the search pattern.
- APPSPACK convergence is not adversely affected

Oracle 1: Space Mapping

- Often referred to as:
  - Multifidelity Optimization (MFO)
  - Surrogate-Based Optimization (SBO)
- The low-fidelity surrogate model retains many of the important features of the high-fidelity "truth" model, but is simplified in some way.
  - Decreased physical resolution
  - Decreased FE resolution
  - Simplified physics
  - Fewer design parameters
- Want models such that low-fidelity trends match high-fidelity trends
- Space mapping acts as a conduit between the high and low fidelity models (Bandler et al.)

MFO for HC



Sandia National Laboratories

MFO Results for HC

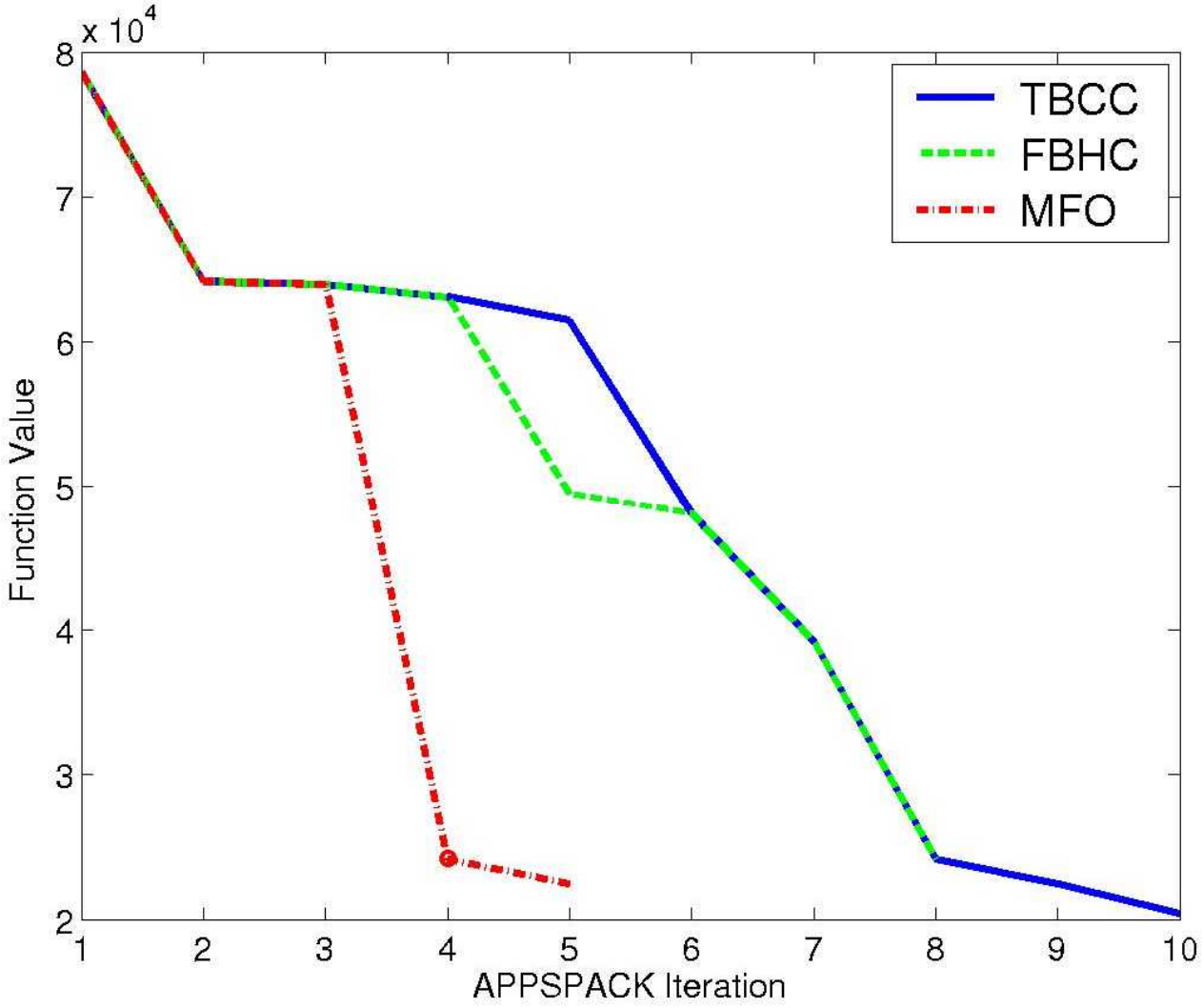
Initial cost: $78,586

Method	Cost	% decrease	# Fn evals
FBHC	$24,175	69.2%	117 mf2k 0 mt3d
TBCC	$20,362	74.1%	188 mf2k 160 mt3d
MFO	$22,428	71.5%	152 mf2k 86 mt3d

MODFLOW (mf2k): ~2 seconds
mt3d: ~50 seconds

Sandia National Laboratories

MFO Results for HC



Sandia National Laboratories

Oracle 2: tgp

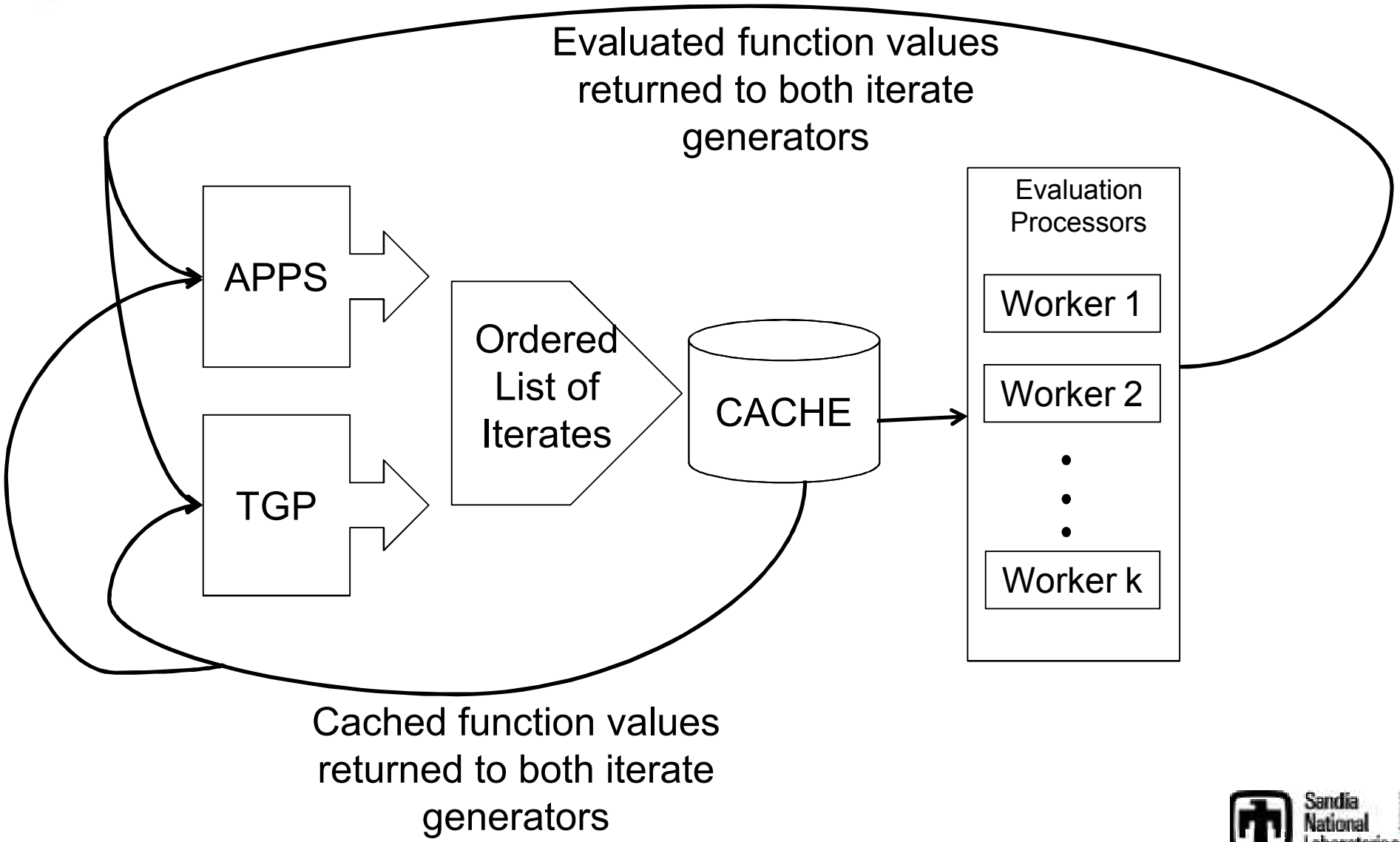
- Gaussian Process: stochastic process that quantifies the uncertainty about future evaluations conditional on previously evaluated points
- Adapted to complex computer simulations using treed partitioning
- tgp: R code implementation (Gramacy, Taddy, Lee)
  - Available from the CRAN
  - Stats award for Gramacy thesis

Implementation of APPS-tgp

1. Discretize decision space and include a dense LHS of extra points around current "best" point
2. The stats model (MCMC fit) produces a "posterior" distribution for the response at each point
3. The resulting model gives a full distribution of points for improvement

- NOTES:
  - Recursive algorithm ranks the list of points to be evaluated
  - Alternative: Pre-seed tgp so that it can start working immediately

APPS-TGP

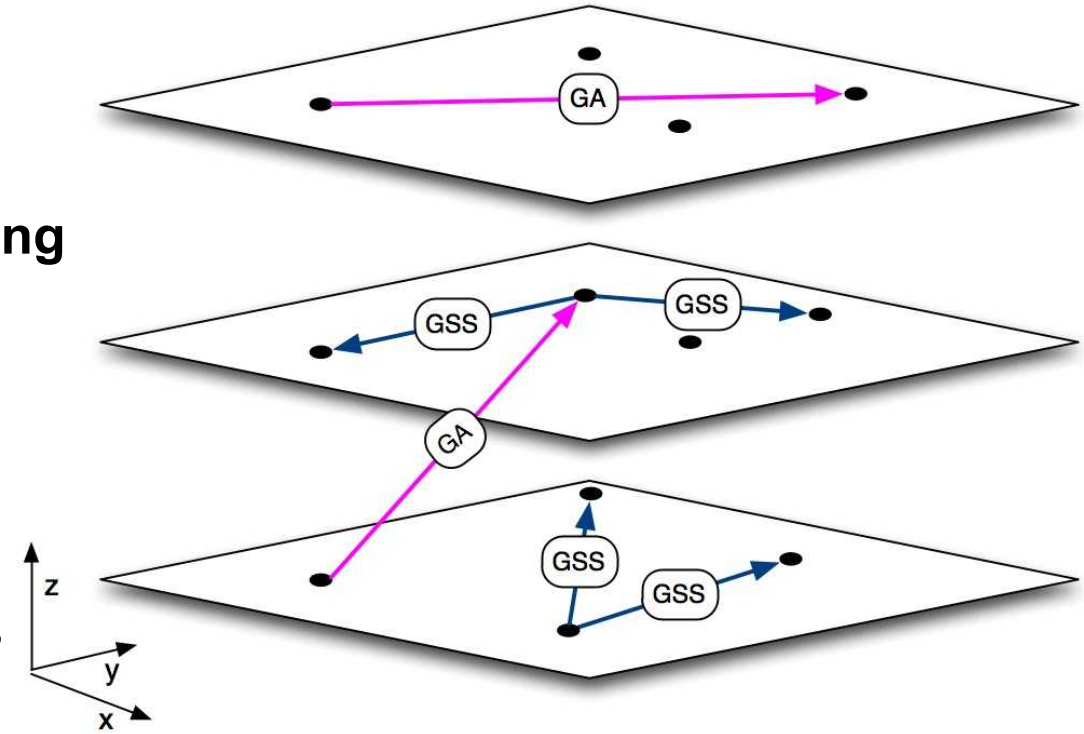


HC Results with APPS-tgp

- No initial point
- Solutions: 1 well with cost of
  - FBHC Formulation: $23,796
  - TBCC Formulation: $22,182
- Without a starting point, most (singular) methods quit without finding a reasonable solution
- The GA does not require a starting point but exits without converging after ~1000 evaluations
- The hybrid finds the solution in about ~700 evaluations
- Parallelism means improved wall clock time

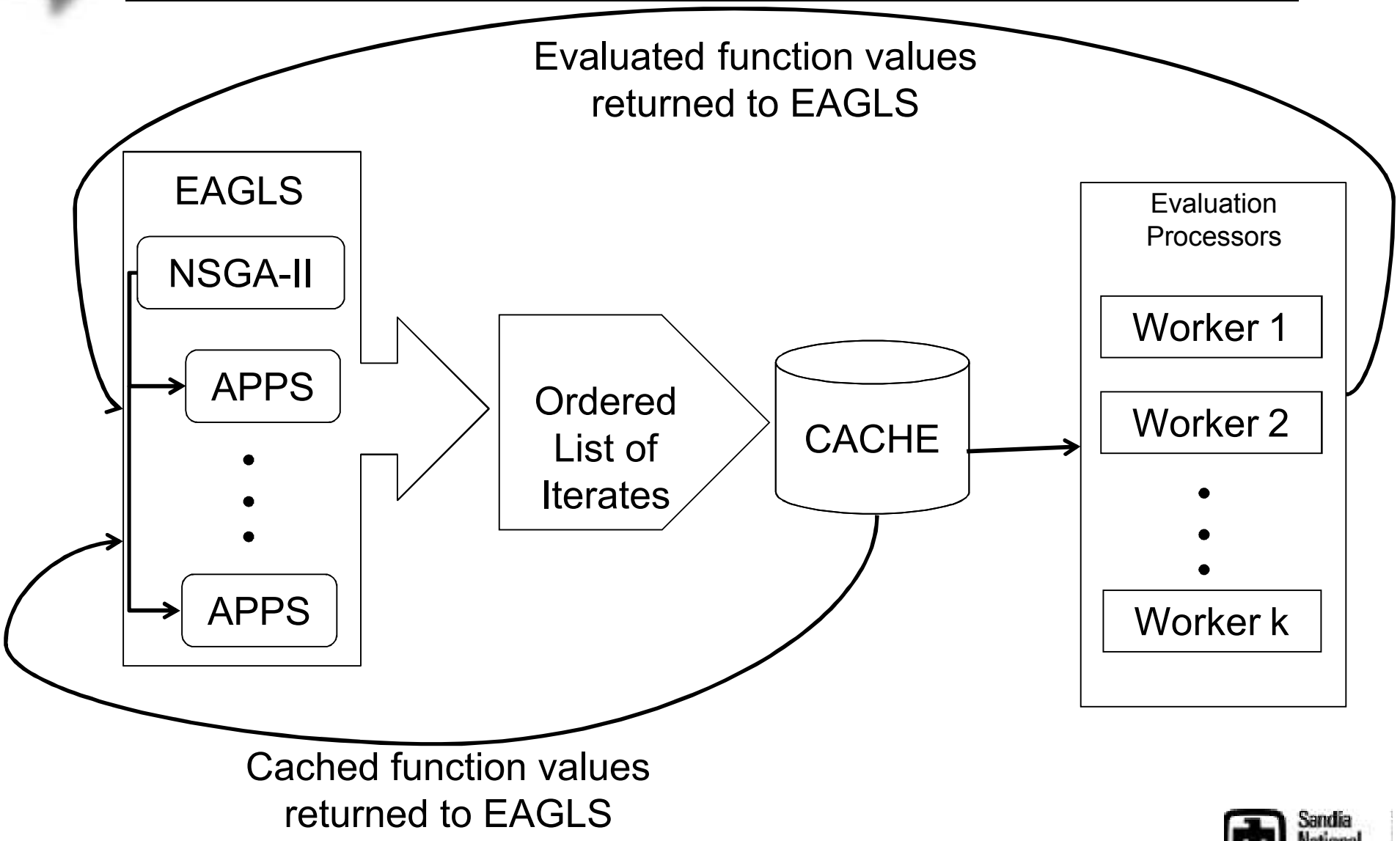
EAGLS

- Current Implementation: EA = NSGAII; LS= APPS
- Generate a population using the GA
- Evaluate the GA population
- Select “promising” points
- Refine promising points using the LS
  - Only real parameters can vary
  - Integer parameters held fixed
- Ranking algorithm prevents prohibitive growth
- Fowler (Clarkson), Griffin (SAS), Hemker (TU-Darmstadt), Parno (MIT)

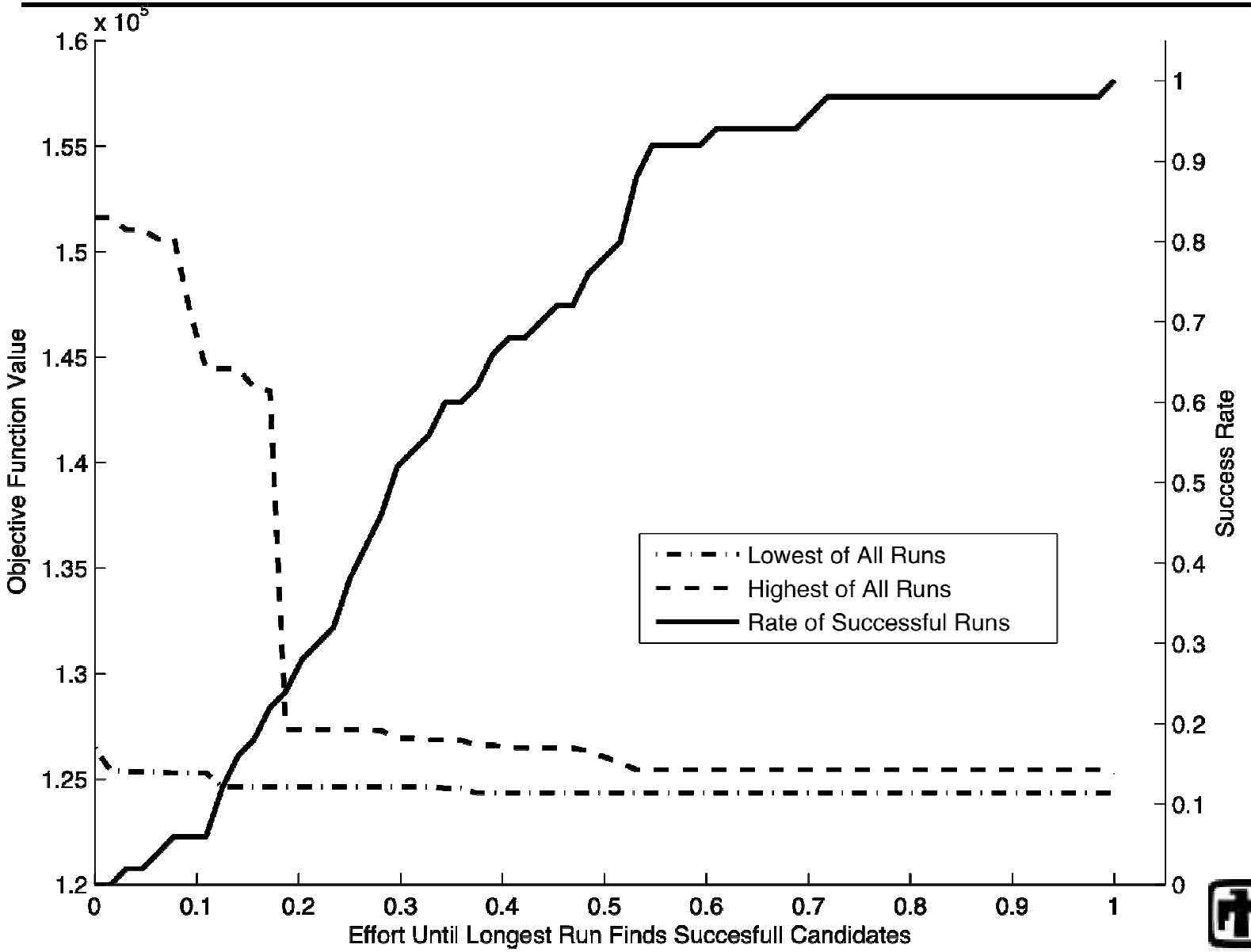


Sandia National Laboratories

EAGLS



EAGLS to SOLVE UNC6



Sandia National Laboratories

UNC6 Summary

Method	Found 5-well Solution?	# of Function Evaluations	Starting Point Required?
APPS	Y	176	Y
IFFCO	Y	< 200	Y
DIRECT	N	---	Y
GA	Y	161	Y *
APPS-TGP	Y	492	N
EAGLS	Y	65	N

*No starting point required by the algorithm; no solution found without a starting point

Not All Hybrids Work

- What should we combine and why?
- How do we combine two or more methods to best exploit the advantages of each?
- Remember the goal: global convergence with the work of a local method

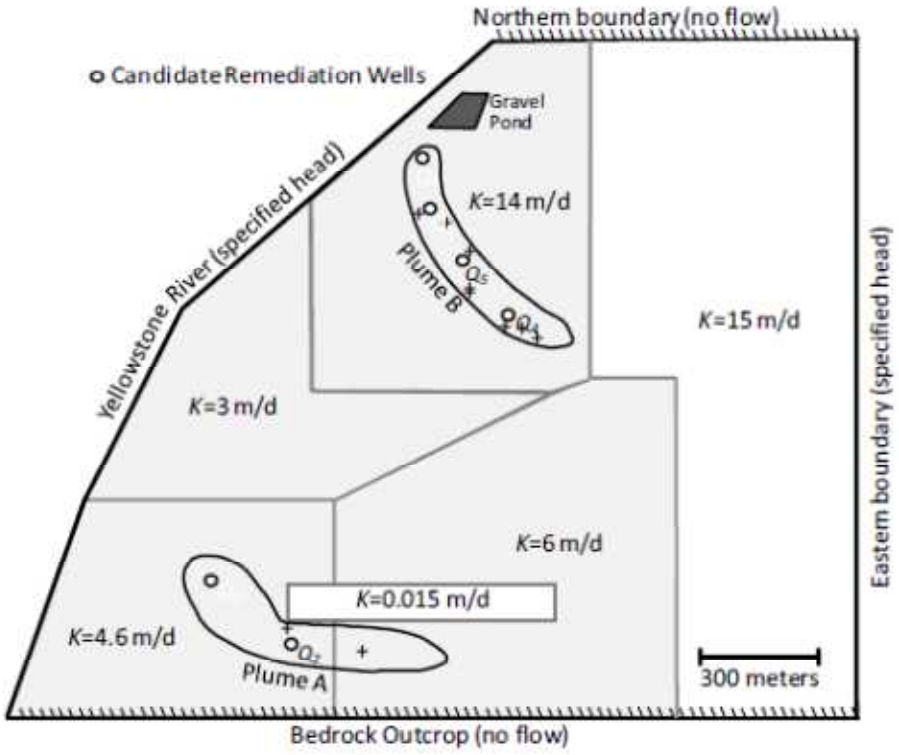
DIRECT-IFFCO

- Idea: Use DIRECT to find starting points for IFFCO
- Carter, Goblansky, Patrick, Kelley, Eslinger (2001) showed this approach to be useful for gas pipeline transmission
- For UNC6
  - Points identified by DIRECT were too close to local minima
  - DIRECT unable to locate any solutions with 5 wells operating on the bound constraint pumping rates
  - IFFCO unable to improve upon points identified by DIRECT
  - Conclusion: not useful for this problem

DIRECT-TGP

- Goal: improve local search capabilities of DIRECT
- Algorithm
  1. Build TGP surrogate using all known function values
  2. Start a local search on the surrogate, constrained to the rectangle using the center of the rectangle as the starting point
  3. Evaluate f at the local minima and return this value
- Initial testing of algorithm is promising for test problems an application with few local minima
- Still problematic for UNC6
- NOTE: DIRECT + local search not a promising approach

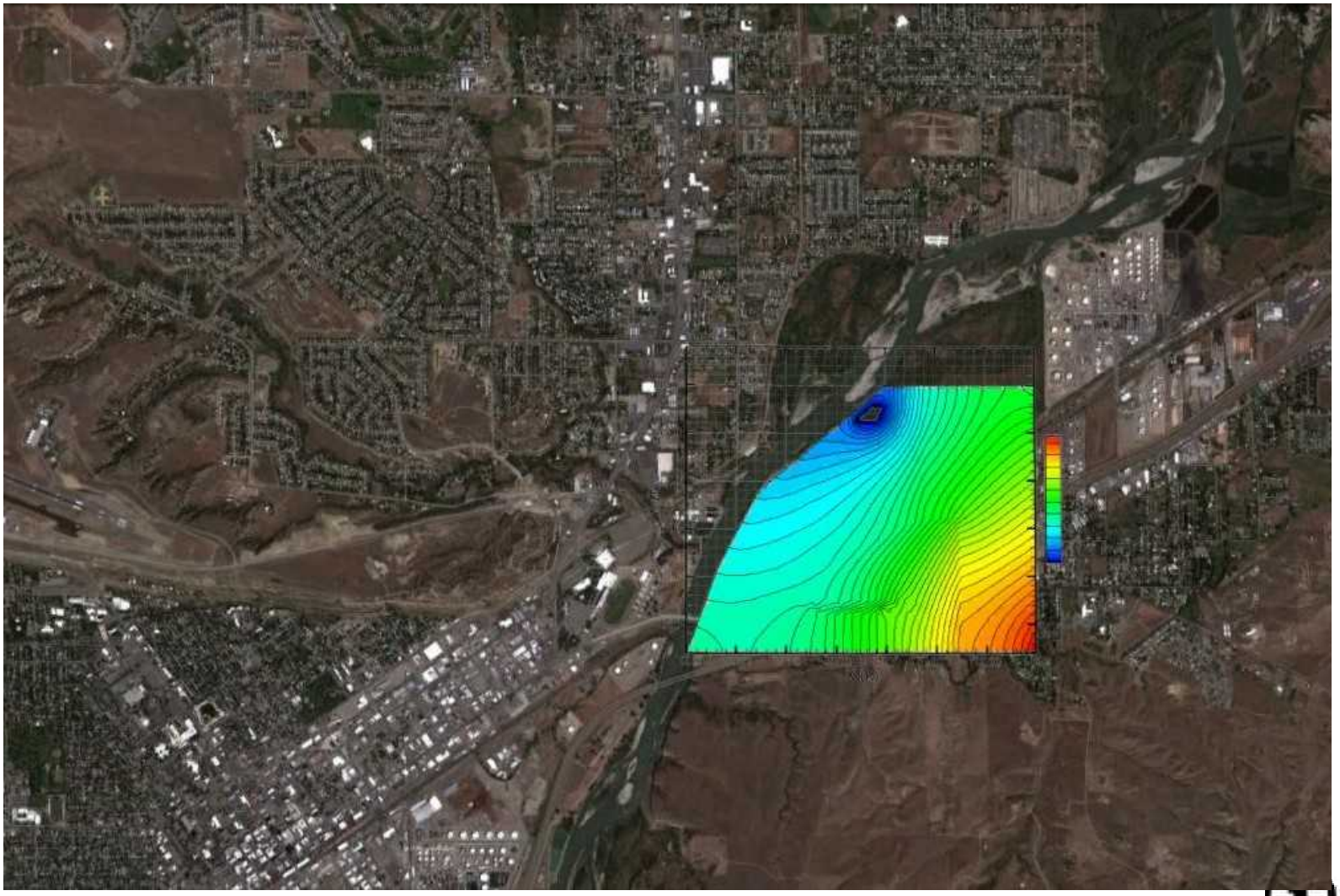
Lockwood Solvent Groundwater Plume Site



- Located near Billings, MT
- Two separate plumes of chlorinated solvents migrating towards the Yellowstone River
- Six pump-and-treat wells proposed to prevent further expansion
- Minimize pumping while successfully containing plumes

Sandia National Laboratories

Pressure Gradient



Sandia National Laboratories

Results

- Six well solutions at specified sites (Mattot)
  - Tried 10 different derivative-free methods
  - Wide range of solutions
  - Minimized cost ranged from $23K-$50K
- Determine the location of the six wells too
  - Reduces the cost to $19K
- Are six wells needed?
  - Some 6 well solutions showed no pumping or a very low pumping rate for 1 or 2 of the wells
  - Reformulate problem: allow the wells to be moved, add a cost for installation, add a discrete variable to include/exclude wells from the design
  - EAGLS found a two well solution (one well in each plume)

What's Next

- SQUaC: Simultaneous Quantification of Uncertainty and Calibration
  - Examining iterates after the completion of the optimization creates a sequential UQ process
  - Consider a more parallel process where UQ and optimization are being performed simultaneously
  - Use a hybrid approach with a new version of TGP for mixed variable parameter sets; SA info for free
- Berry Farm Planning
  - Salinas Valley growers
  - Minimize water usage
  - Follow planting rules
  - MINLP with many constraints