Optimization in ComPASS-4

Stefan Wild & Jeff Larson

Argonne National Laboratory
Mathematics and Computer Science Division

April 23, 2018

The Plan

1. Optimization Formulations and Taxonomy
   - Stochastic Optimization
   - Multiobjective Optimization
   - Simulation-Based Optimization
   - Derivative-Free Optimization
   - Global Optimization
2. An Example LPA Optimization to Highlight Challenges
3. POPAS
4. Why not Blackbox Optimization
5. APOSMM

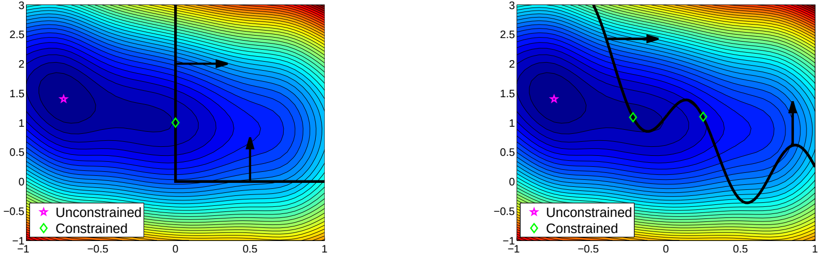
Mathematical/Numerical Nonlinear Optimization

Optimization is the *"science of better"*

Find parameters (controls) $x = (x_1, \ldots, x_n)$ in domain Ω to improve objective f

$$\min \{f(x) : x \in \Omega \subseteq \mathbb{R}^n\}$$

- (Unless Ω is very special) Need to evaluate f at many x to find a good $\hat{x}_*$
- Focus on local solutions: $f(\hat{x}_*) \leq f(x) \; \forall x \in \mathcal{N}(\hat{x}_*) \cap \Omega$
- constraints defined the feasibility region Ω



Stochastic Optimization

Addresses situations where you obtain a nondeterministic quantity $F(x,\xi)$

$$\min\left\{f(x) = \mathrm{E}\left\{F(x,\xi)\right\} : \ x \in \Omega\right\}$$

- $x \in \mathbb{R}^n$ decision variables
- ξ vector of random variables
  - independent of x
  - $P(\xi)$ distribution function for ξ
  - ξ has support Ξ
- $F(x,\cdot)$ functional form of uncertainty for decision x
- $\Omega \subseteq \mathbb{R}^n$ set defined by deterministic constraints
  - Also: stochastic/probabilistic constraints

- Nonstationarity: does $\mathrm{Var}\left\{F(x,\xi)\right\}$ depend on x?

Multiobjective Optimization

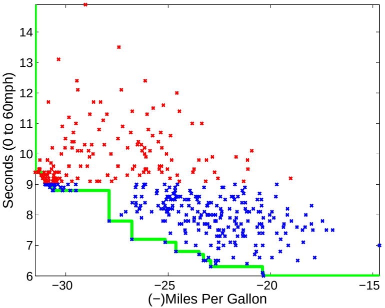
Simultaneously minimize $n_f > 1$ objectives

$$\min_{x\in\Omega} f_1(x), \cdots, f_{n_f}(x)$$

"x^1 dominates x^2" if:

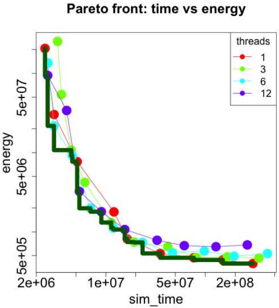
- $f_i(x^1) \le f_i(x^2)$ for all i, and
- $f_i(x^1) < f_i(x^2)$ for at least one i

"x^1 is nondominated in $\mathcal{X}$" if there is no $x^2 \in \mathcal{X}$ that dominates x^1



Pareto optimal solutions: A set $\mathcal{P}$ of points are nondominated in Ω

- Especially useful when missing a currency exchange between objectives
- Significantly more expensive than single-objective optimization



Simulation-Based Optimization

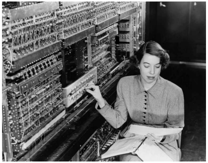
$$\min_{x\in\mathbb{R}^n} \{f(x) = F[\mathbf{S}(\mathbf{x})] : c(\mathbf{S}(\mathbf{x})) \leq 0, x \in \mathcal{B}\}$$

- S (numerical) simulation output, (here deterministic)
- Derivatives $\nabla_x S$ often unavailable or prohibitively expensive to obtain/approximate directly
- Some AD hurdle (e.g., proprietary/legacy/coupled/mixed-language codes)
- Single evaluation of S could take seconds/minutes/hours/days

  Evaluation is a bottleneck for optimization

$\mathcal{B}$ compact, known region (e.g., finite bound constraints)

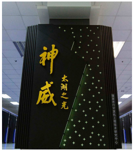
Computing advances have driven this research area...



Argonne's AVIDAC
(1953 vacuum tubes)

Argonne's BlueGene/Q
(2012 0.79M cores)

Argonne's Theta
(2017 0.23M cores)



Sunway TaihuLight
(2016 11M cores)

Derivative-Free/Zero-Order Optimization

"Some derivatives are unavailable for optimization purposes"

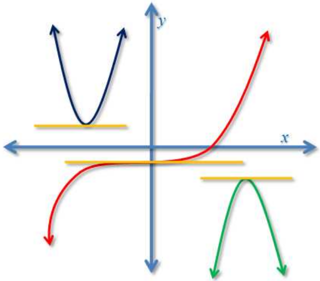
Derivative-Free/Zero-Order Optimization

"Some derivatives are unavailable for optimization purposes"

The Challenge: Optimization is tightly coupled with derivatives

Typical optimality (no noise, smooth functions)

$$\nabla_x f(x^*) + \lambda^T \nabla_x c_E(x^*) = 0, c_E(x^*) = 0$$



(sub)gradients $\nabla_x f$, $\nabla_x c$ enable:

- Faster feasibility
- Faster convergence
  - Guaranteed descent
  - Approximation of nonlinearities
- Better termination
  - Measure of criticality $\|\nabla_x f\|$ or $\|\mathcal{P}_\Omega(\nabla_x f)\|$
- Sensitivity analysis
  - Correlations, standard errors, UQ, ...

Ways to Get Derivatives

(assuming they exist)

Handcoding (HC)

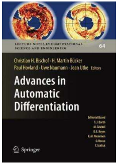
"Army of students/programmers"

- ? Prone to errors/conditioning
- ? Intractable as number of ops increases

Algorithmic/Automatic Differentiation (AD)

"Exact* derivatives!"

- ? No black boxes allowed
- ? Not always automatic/cheap/well-conditioned



Finite Differences (FD)

"Nonintrusive"

- ? Expense grows with n
- ? Sensitive to stepsize choice/noise

→[Moré & W.; SISC 2011], [Moré & W.; TOMS 2012]

. . . then apply derivative-based method (that handles inexact derivatives)

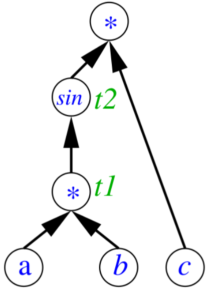
Algorithmic Differentiation

→ [Coleman & Xu; SIAM 2016], [Griewank & Walther; SIAM 2008]

Computational Graph

- $y = sin(a * b) * c$
- Forward and reverse modes
- AD tool provides code for your derivatives

Write codes and formulate problems with AD in mind!

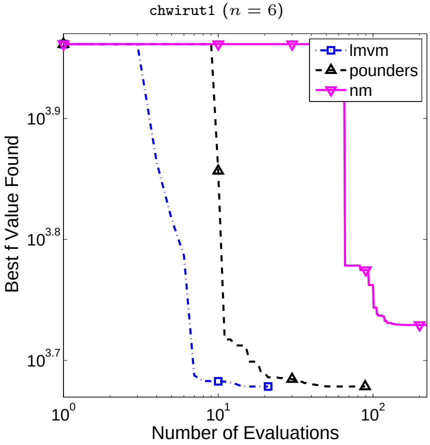


Many tools (see `www.autodiff.org`):

F	`OpenAD`
F/C	`Tapenade`, `Rapsodia`
C/C++	`ADOL-C`, `ADIC`
Matlab	`ADiMat`, `INTLAB`
Python/R	`ADOL-C`

Also done in `AMPL`, `GAMS`, `JULIA`!

The Price of Algorithm Choice: Solvers in PETSc/TAO



Toolkit for Advanced Optimization
[Munson et al.; mcs.anl.gov/tao]

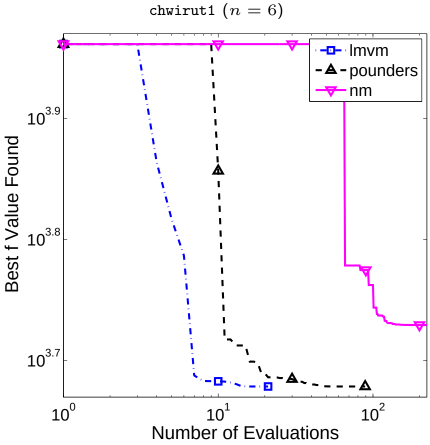
Increasing level of user input:

nm Assumes $\nabla_x f$ unavailable, black box

pounders Assumes $\nabla_x f$ unavailable, exploits problem structure

lmvm Uses available $\nabla_x f$

The Price of Algorithm Choice: Solvers in PETSc/TAO



Toolkit for Advanced Optimization
[Munson et al.; mcs.anl.gov/tao]

Increasing level of user input:

nm Assumes $\nabla_x f$ unavailable, black box

pounders Assumes $\nabla_x f$ unavailable, exploits problem structure

THIS TALK

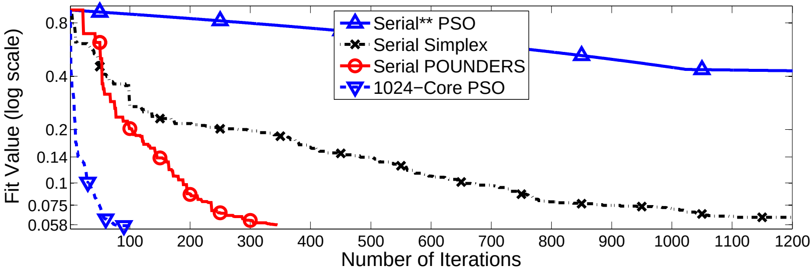
lmvm Uses available $\nabla_x f$

DFO methods should be designed to beat finite-difference-based methods

Observe: Constrained by budget on #evals, method limits solution accuracy/problem size

Why Algorithms Matter: The Accelerator Case

Varying skew quadrupoles to meet beam size targets (in PELEGANT)



- Heuristics often "embarrassingly/naturally parallel"; PSO= particle swarm method
  - Typically through stochastic sampling/evolution
  - 1024 function evaluations per iteration
- Simplex is Nelder-Mead; POUNDERS is model-based trust-region algorithm
  - one function evaluation per iteration

Global Optimization, $\min_{x \in \Omega} f(x)$

Careful:

- Global convergence: Convergence (to a local solution/stationary point) from anywhere in Ω
- Convergence to a global minimizer: Obtain x^* with $f(x^*) \leq f(x) \, \forall x \in \Omega$

Global Optimization, $\min_{x\in\Omega} f(x)$

Careful:

- Global convergence: Convergence (to a local solution/stationary point) from anywhere in Ω
- Convergence to a global minimizer: Obtain x^* with $f(x^*) \leq f(x) \forall x \in \Omega$

Anyone selling you global solutions when derivatives are unavailable:

either assumes more about your problem (e.g., convex f)

or expects you to wait forever

Törn and Žilinskas: An algorithm converges to the global minimum for any continuous f if and only if the sequence of points visited by the algorithm is dense in Ω.

or cannot be trusted

Global Optimization, $\min_{x\in\Omega} f(x)$

Careful:

- Global convergence: Convergence (to a local solution/stationary point) from anywhere in Ω
- Convergence to a global minimizer: Obtain x^* with $f(x^*) \leq f(x) \forall x \in \Omega$

Anyone selling you global solutions when derivatives are unavailable:

either assumes more about your problem (e.g., convex f)

or expects you to wait forever

Törn and Žilinskas: An algorithm converges to the global minimum for any continuous f if and only if the sequence of points visited by the algorithm is dense in Ω.

or cannot be trusted

Instead:

- Rapidly find good local solutions and/or be robust to poor solutions
- Consider multistart approaches and/or structure of multimodality

Why Multistart?

Best minimizer(s) approximate global minimizer x^*, $f(x^*) \leq f(x) \forall x \in \mathcal{D}$

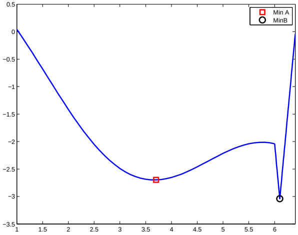
Multiple local minima are often of interest in practice

Design Multiple objectives/constraints might later be of interest

Distinctness j best minimizers have physical meaning

Simulation Errors Spurious local minima from simulator anomalies

Uncertainty Some minima more sensitive to perturbations



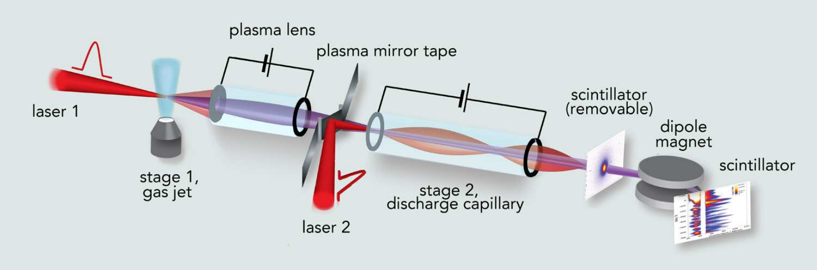
Increased opportunity for parallelism

Trilevel simulation/function $\rightarrow$ local solver $\rightarrow$ global solver

Efficient local solvers

- (Local) surrogate-based, exploit problem structure
  - least-squares objectives, (un)relaxable constraints, known nonsmoothness, . . .

Motivating Example: Staging a Laser Plasma Accelerator



- Electron bunch is injected in a laser-induced plasma wave
  - Typically when laser intensity reaches its first maximum
- Nonlinear effects $\Rightarrow$ plasma wave shrinks and electron bunch is lost
  - Typically because bunch ends up in a defocusing region when laser intensity reaches its (first) minimum

Goal: Shape initial section of capillary to raise the minimum intensity and/or lower the maximum intensity.

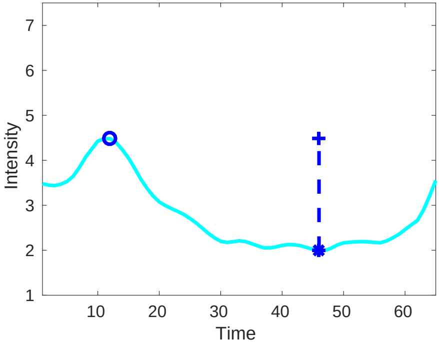
$\rightarrow$For a given x, we compute $v(t; x)$, the (smooth) laser intensity at time t

Under ComPASS-3 with Carlo Benedetti & Jean-Luc Vay (LBNL)

Motivating Example: $\min\{f(x) : x \in \mathcal{D} \subset \mathbb{R}^n\}$

Simulation provides intensity at a discrete set of times $t_1 < \cdots < t_{p=|I|}$:

$$B_i(x) = v(t_i; x), \qquad i \in I$$



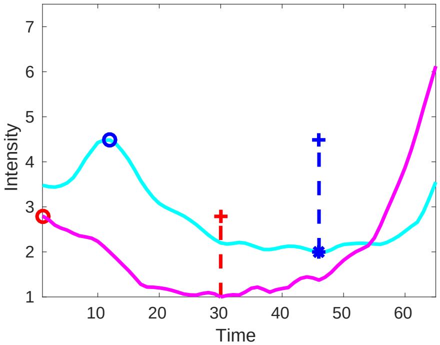
$$f(x) = \max_{i \in \Theta_1(x)} v(t_i; x) - \min_{i \in I} v(t_i; x)$$

$$\Theta_1(x) = \left\{ i \in I : i \leq \max \operatorname{argmin}_{j \in I} v(t_j; x) \right\}$$

Motivating Example: $\min\{f(x) : x \in \mathcal{D} \subset \mathbb{R}^n\}$

Simulation provides intensity at a discrete set of times $t_1 < \cdots < t_{p=|I|}$:

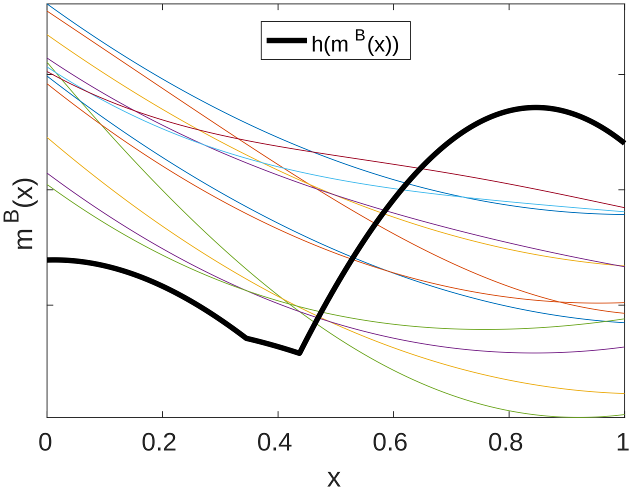
$$B_i(x) = v(t_i; x), \qquad i \in I$$



$$f(x) = \max_{i \in \Theta_1(x)} v(t_i; x) - \min_{i \in I} v(t_i; x)$$

$$\Theta_1(x) = \left\{ i \in I : i \leq \max \operatorname*{argmin}_{j \in I} v(t_j; x) \right\}$$

Slice Through LPA Subproblem



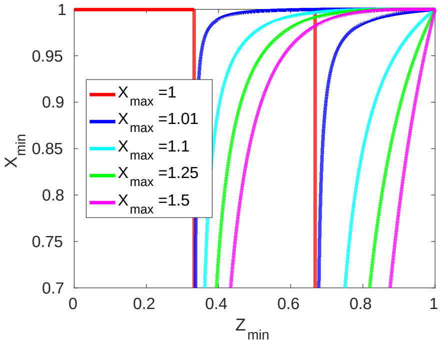
This is a nonsmooth (piecewisesmooth) function of the parameters x

LPA Feasible Region

Variable	Range
Length	$2 \leq L \leq 6$
Plasma channel radius	$1 \leq X_{\text{max}} \leq 1.5$
Minimum channel radius	$0.7 \leq X_{\text{min}} \leq 1$
Longitudinal location	$0 \leq Z_{\text{min}} \leq 1$
Laser focus position	$-1.2 \leq Z_f \leq 0$

$$
\begin{aligned}
c_1(x) = & - X_{\text{max}} Z_{\text{min}}^4 \\
& - (X_{\text{max}} - X_{\text{min}})(2Z_{\text{min}} - 3Z_{\text{min}}^2) \\
\leq & 0
\end{aligned}
$$

$$
\begin{aligned}
c_2(x) = & X_{\text{max}}(Z_{\text{min}}^4 - 4Z_{\text{min}}^3 + 3Z_{\text{min}}^2) \\
& + (X_{\text{max}} - X_{\text{min}})(3Z_{\text{min}}^2 - 4Z_{\text{min}} + 1) \\
\leq & 0
\end{aligned}
$$

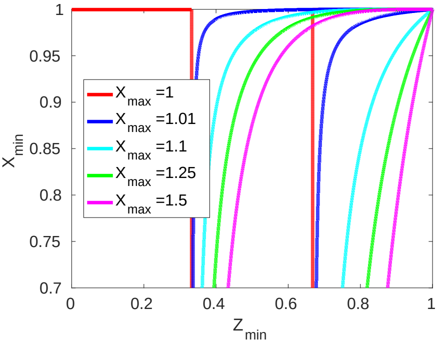


LPA Feasible Region

Variable	Range
Length	$2 \leq L \leq 6$
Plasma channel radius	$1 \leq X_{\max} \leq 1.5$
Minimum channel radius	$0.7 \leq X_{\min} \leq 1$
Longitudinal location	$0 \leq Z_{\min} \leq 1$
Laser focus position	$-1.2 \leq Z_f \leq 0$

$$\begin{aligned} c_1(x) = & - X_{\max} Z_{\min}^4 \\ & - (X_{\max} - X_{\min})(2Z_{\min} - 3Z_{\min}^2) \\ \leq & 0 \end{aligned}$$

$$\begin{aligned} c_2(x) = & X_{\max}(Z_{\min}^4 - 4Z_{\min}^3 + 3Z_{\min}^2) \\ & + (X_{\max} - X_{\min})(3Z_{\min}^2 - 4Z_{\min} + 1) \\ \leq & 0 \end{aligned}$$



$c(x) \leq 0$ are UNRELAXABLE: Simulator (often) fails in $\mathcal{D}^c$

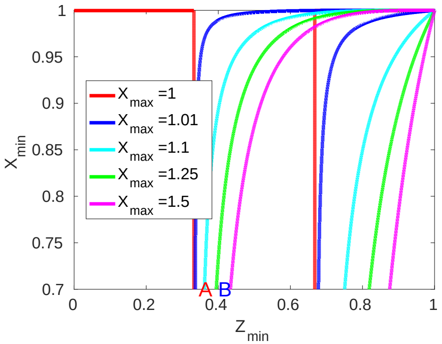
QUAK SBO constraint taxonomy →[Le Digabel & W.; ANL/MCS-P5350-0515]

LPA Feasible Region

Variable	Range
Length	$2 \le L \le 6$
Plasma channel radius	$1 \le X_{\max} \le 1.5$
Minimum channel radius	$0.7 \le X_{\min} \le 1$
Longitudinal location	$0 \le Z_{\min} \le 1$
Laser focus position	$-1.2 \le Z_f \le 0$

$$\begin{aligned} c_1(x) =& - X_{\max} Z_{\min}^4 \\ & - (X_{\max} - X_{\min})(2Z_{\min} - 3Z_{\min}^2) \\ \le& 0 \end{aligned}$$

$$\begin{aligned} c_2(x) =& X_{\max}(Z_{\min}^4 - 4Z_{\min}^3 + 3Z_{\min}^2) \\ & + (X_{\max} - X_{\min})(3Z_{\min}^2 - 4Z_{\min} + 1) \\ \le& 0 \end{aligned}$$



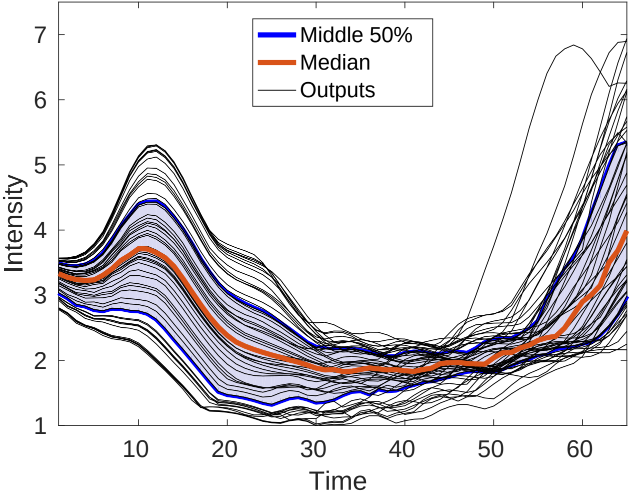
$c(x) \le 0$ are UNRELAXABLE: Simulator (often) fails in $\mathcal{D}^c$

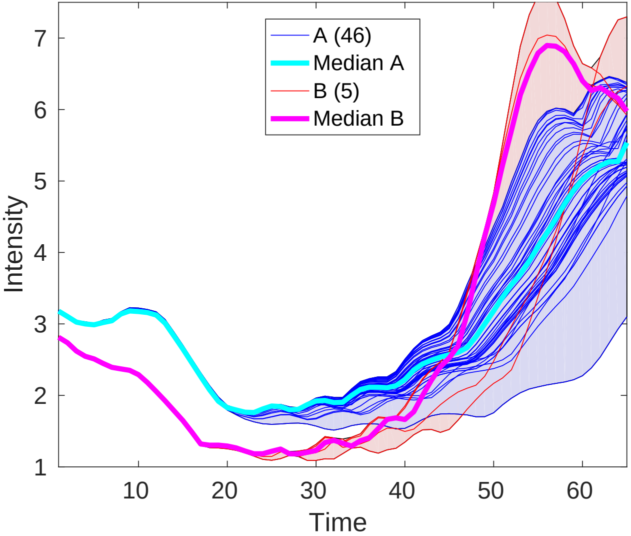
QUAK SBO constraint taxonomy →[Le Digabel & W.; ANL/MCS-P5350-0515]

Numerical Experiments on LPA Problem

Test multimodality:

- 51 starting points x^0 generated uniformly from within $\mathcal{D}$
- Significant variation in $f(x^0)$
- Includes pathological $t_1 = \arg\max_{i \in \Theta_1(x^0)} v(t_i$
- Maximum of $20n$ v evaluations (7.5 minutes each)
- 51 CPU days





Structured POUNDER code

51 Solutions:

- Converge to two solutions (A, B)
- $\approx 10\%$ to B
- Behavior after $t_{\max\{i:i\in\Omega_1\}}$ unconstrained
- $c(x^A), c(x^B) < 0$

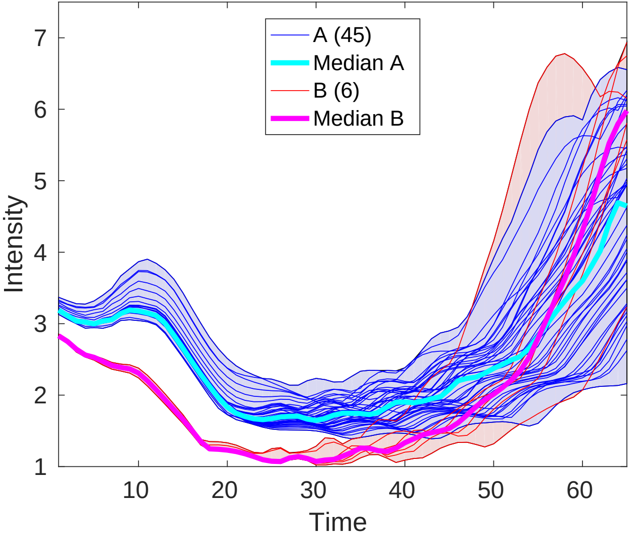
PS solutions remarkably consistent

Solutions Found for LPA Problem

51 Solutions:

- Converge to two solutions (A, B)
- $\approx 10\%$ to B
- Behavior after $t_{\max\{i: i \in \Omega_1\}}$ unconstrained
- $c(x^A), c(x^B) < 0$

PS solutions remarkably consistent



Constrained Nelder-Mead code

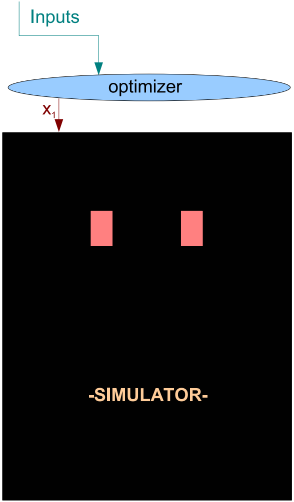
POPAS Activity Proposed for ComPASS-4

Platform for Optimization of Particle Accelerators at Scale

- integrated platform for coordinating the evaluation and numerical optimization of accelerator simulations on leadership-class DOE computers
- orchestrate concurrent evaluations of OSIRIS, QuickPIC, Synergia, and MARS (or combinations thereof) with distinct inputs/parameter values
- account for resource requirements of the above
- API will allow the user to describe the mapping from simulation outputs and the derived quantities of interest used to define objective and constraint quantities

TH: Provide enough information so that optimization is efficient

“Simplest” (=Most Naive) Formulation: Blackbox f



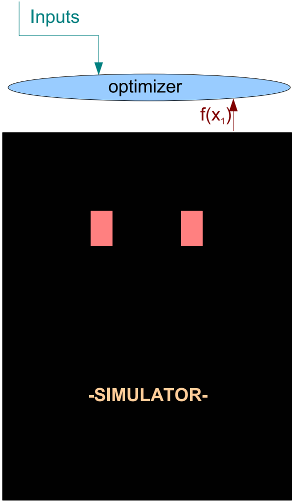
Optimizer gives x, physicist provides $f(x)$

- f can be a blackbox (executable only or proprietary/legacy codes)
- Only give a single output
  - no derivatives with respect to x: $\nabla_x S(x), \nabla^2_{x,x} S(x)$
  - no problem structure

Good solutions guaranteed in the limit, but:

- Computational budget limits number of evaluations

"Simplest" (=Most Naive) Formulation: Blackbox f



Optimizer gives x, physicist provides $f(x)$

- f can be a blackbox (executable only or proprietary/legacy codes)
- Only give a single output
  - no derivatives with respect to x: $\nabla_x S(x), \nabla^2_{x,x} S(x)$
  - no problem structure

Good solutions guaranteed in the limit, but:

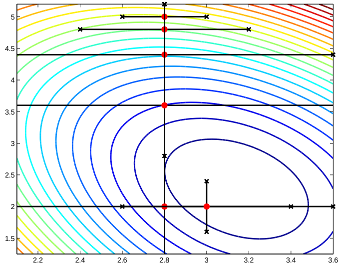
- Computational budget limits number of evaluations

Two main styles of local algorithms

- Direct search methods (pattern search, Nelder-Mead, ...)
- Model- ("surrogate-")based methods (quadratics, radial basis functions, ...)

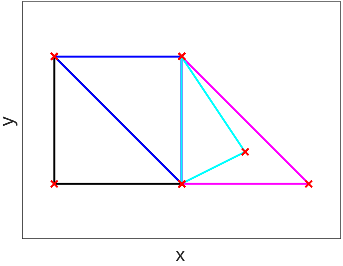
Black-Box Algorithms: Direct Search Methods

Pattern Search + Variants



Easy to parallelize f evaluations

Nelder-Mead + Variants



Popularized by Numerical Recipes

- Rely on indicator functions: $[f(x_k + s) <^? f(x_k)]$ $f(x_k)$, short memory
- Work with **black-box** $f(x)$, do not exploit structure $F[x, S(x)]$
- Convergence results for variety of settings

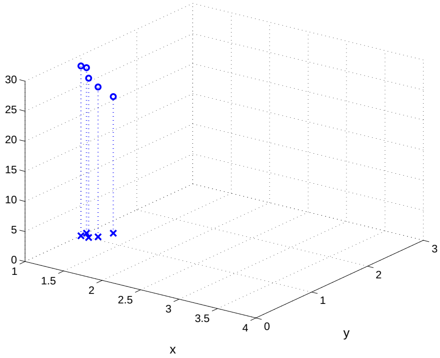
Survey → [Kolda, Lewis, Torczon; SIREV 2003]

Newer NM → [Lagarias, Poonen, Wright; SIOPT 2012]

Tools → `DFL` [Liuzzi et al.], `NOMAD` [Audet et al.], . . .

Making the Most of Little Information About Smooth f

- Overhead of the optimization routine is minimal (negligible?) relative to cost of evaluating simulation



Bank of data, $\{x_i, f(x_i)\}_{i=1}^{k}$:

= Points (& function values) evaluated so far

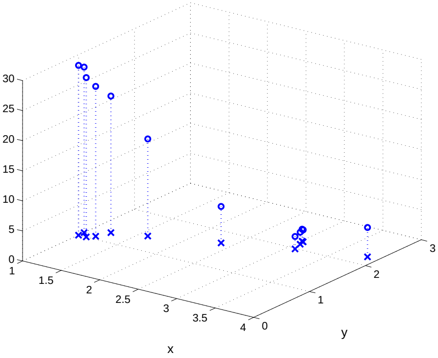
= Everything known about f

Goal:

- Make use of growing Bank as optimization progresses
- Limit unnecessary evaluations (geometry/approximation)

Making the Most of Little Information About Smooth f

◇ Overhead of the optimization routine is minimal (negligible?) relative to cost of evaluating simulation



Bank of data, $\{x_i, f(x_i)\}_{i=1}^{k}$:

= Points (& function values) evaluated so far

= Everything known about f

Goal:

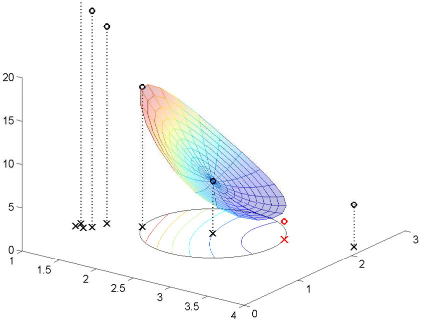
◇ Make use of growing Bank as optimization progresses

◇ Limit unnecessary evaluations

(geometry/approximation)

Making the Most of Little Information About Smooth f

- Overhead of the optimization routine is minimal (negligible?) relative to cost of evaluating simulation



Bank of data, $\{x_i, f(x_i)\}_{i=1}^{k}$:

= Points (& function values) evaluated so far

= Everything known about f

Goal:

- Make use of growing Bank as optimization progresses
- Limit unnecessary evaluations (geometry/approximation)

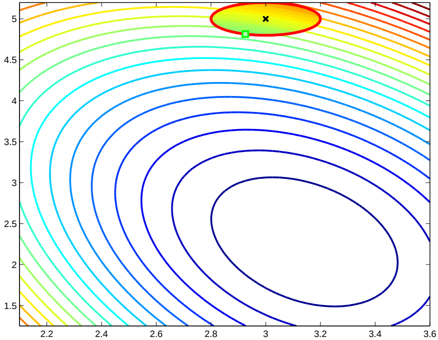
Derivative-Free, Model-Based Trust-Region Algorithms

Substitute $\min\{m_k(x) : x \in \mathcal{B}_k\}$ (TRSP) for $\min f(x)$

f expensive, no ∇f

m_k cheap, analytic derivatives

Trust region:
$\mathcal{B}_k = \{x \in \Omega : \|x - x^k\| \leq \Delta_k\}$



Basic algorithm

- Build model $m_k (\approx f$ in $\mathcal{B}_k)$
- $x^+ \approx \arg\min\{m_k(x) : x \in \mathcal{B}_k\}$
- $\rho_k = \frac{f(x^k) - f(x^+)}{m_k(x^k) - m_k(x^+)}$
- If $\rho_k \geq \eta_1 > 0$, accept $x^{k+1} = x^+$;
  Elseif m_k is valid in $\mathcal{B}_k$, shrink Δ_k
  Else, improve m_k in $\mathcal{B}_k$

ORBIT: [W., Regis, Shoemaker, SISC 2008]

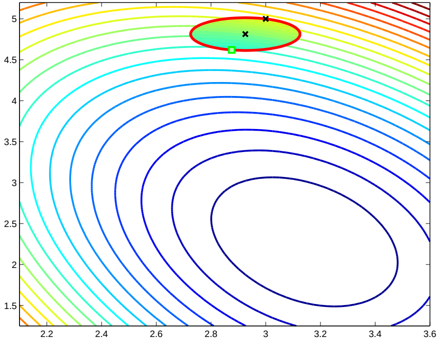
Derivative-Free, Model-Based Trust-Region Algorithms

Substitute $\min\{m_k(x) : x \in \mathcal{B}_k\}$ (TRSP) for $\min f(x)$

f expensive, no ∇f

m_k cheap, analytic derivatives

Trust region:
$\mathcal{B}_k = \{x \in \Omega : \|x - x^k\| \leq \Delta_k\}$



Basic algorithm

- Build model $m_k (\approx f$ in $\mathcal{B}_k)$
- $x^+ \approx \arg\min\{m_k(x) : x \in \mathcal{B}_k\}$
- $\rho_k = \frac{f(x^k) - f(x^+)}{m_k(x^k) - m_k(x^+)}$
- If $\rho_k \geq \eta_1 > 0$, accept $x^{k+1} = x^+$;
  Elseif m_k is valid in $\mathcal{B}_k$, shrink Δ_k
  Else, improve m_k in $\mathcal{B}_k$

ORBIT: [W., Regis, Shoemaker, SISC 2008]

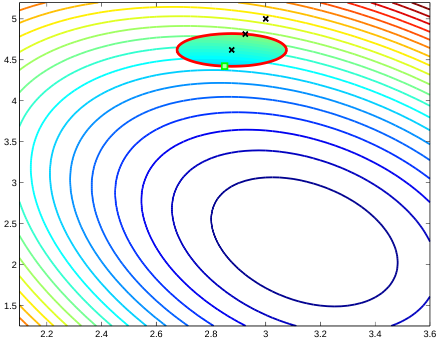
Derivative-Free, Model-Based Trust-Region Algorithms

Substitute $\min\{m_k(x) : x \in \mathcal{B}_k\}$ (TRSP) for $\min f(x)$

f expensive, no ∇f

m_k cheap, analytic derivatives

Trust region:
$\mathcal{B}_k = \{x \in \Omega : \|x - x^k\| \leq \Delta_k\}$



Basic algorithm

- Build model $m_k (\approx f$ in $\mathcal{B}_k)$
- $x^+ \approx \arg\min\{m_k(x) : x \in \mathcal{B}_k\}$
- $\rho_k = \frac{f(x^k) - f(x^+)}{m_k(x^k) - m_k(x^+)}$
- If $\rho_k \geq \eta_1 > 0$, accept $x^{k+1} = x^+$;
  Elseif m_k is valid in $\mathcal{B}_k$, shrink Δ_k
  Else, improve m_k in $\mathcal{B}_k$

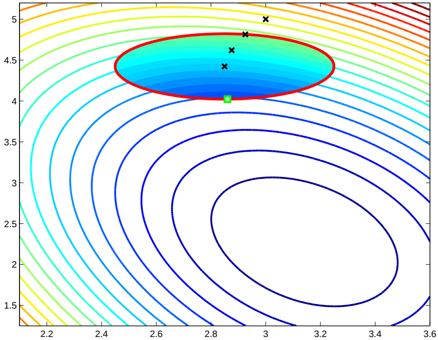
ORBIT: [W., Regis, Shoemaker, SISC 2008]

Derivative-Free, Model-Based Trust-Region Algorithms

Substitute $\min \{m_k(x) : x \in \mathcal{B}_k\}$ (TRSP) for $\min f(x)$

f expensive, no ∇f

m_k cheap, analytic derivatives



Trust region:
$\mathcal{B}_k = \{x \in \Omega : \|x - x^k\| \le \Delta_k\}$

Basic algorithm

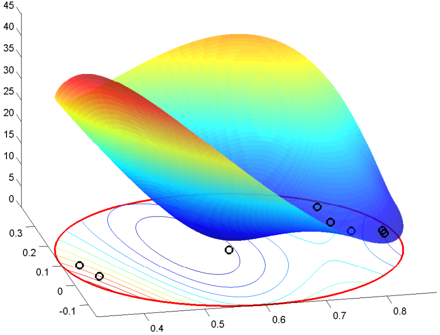
- Build model $m_k (\approx f$ in $\mathcal{B}_k)$
- $x^+ \approx \arg\min\{m_k(x) : x \in \mathcal{B}_k\}$
- $\rho_k = \frac{f(x^k) - f(x^+)}{m_k(x^k) - m_k(x^+)}$
- If $\rho_k \ge \eta_1 > 0$, accept $x^{k+1} = x^+$;
  Elseif m_k is valid in $\mathcal{B}_k$, shrink Δ_k
  Else, improve m_k in $\mathcal{B}_k$

ORBIT: [W., Regis, Shoemaker, SISC 2008]

Radial Basis Function Interpolation Models

Given

- base point x_k
- interpolation points $\mathcal{Y} = \{y_j\}_{j=1}^{|\mathcal{Y}|} \subset \mathbb{R}^n$
- values $f(x_k + y_j)$ for $j = 1, \ldots, |\mathcal{Y}|$
- radial kernel $\phi : \mathbb{R}_+ \to \mathbb{R}$



Unique coefficients λ and polynomial p define interpolating RBF model

$$m_k^f(x_k + s) = \sum_{j=1}^{|\mathcal{Y}|} \lambda_j \phi(\|s - y_j\|) + p(s),$$

Structure in Simulation-Based Optimization, $\min f(x) = F[x, S(x)]$

f is often not a black box S

- **NLS** Nonlinear least squares

$$f(x) = \sum_i (S_i(x) - d_i)^2$$

- **CNO** Composite (nonsmooth) optimization

$$f(x) = h(S(x))$$

- **SKP** Not all variables enter simulation

$$f(x) = g(x_I, x_J) + h(S(x_J))$$

- **BLO** Bilevel optimization

$$\min\{S_1(x_I, x_J) : x_I \in \arg\max_y S_2(y, x_J)\}$$

- **SCO** Only some constraints depend on simulation

$$\min\{f(x) : c_1(x) = 0, c_S(x) = 0\}$$

...

Model-based methods offer one way to exploit such structure

Nonlinear Least Squares $f(x) = \frac{1}{2}\sum_i R_i(x)^2$

Obtain a vector of output $R_1(x), \ldots, R_p(x)$

- Model each R_i

$$R_i(x) \approx m_k^{R_i}(x) = R_i(x_k) + (x - x_k)^\top g_k^{(i)} + \frac{1}{2}(x - x_k)^\top H_k^{(i)}(x - x_k)$$

- Approximate:

$$\begin{aligned}
\nabla f(x) &= \sum_i \nabla \mathbf{R_i(x)} R_i(x) \qquad \longrightarrow \sum_i \nabla m_k^{R_i}(x) R_i(x) \\
\nabla^2 f(x) &= \sum_i \nabla \mathbf{R_i(x)} \nabla \mathbf{R_i(x)}^\top + \sum_i R_i(x) \nabla^2 \mathbf{R_i(x)} \\
&\longrightarrow \sum_i \nabla m_k^{R_i}(x) \nabla m_k^{R_i}(x)^\top + \sum_i R_i(x) \nabla^2 m_k^{R_i}(x)
\end{aligned}$$

- Model f via Gauss-Newton or similar

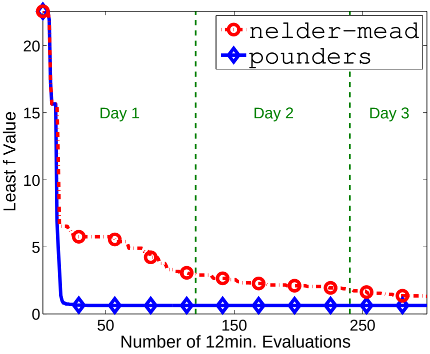
regularized Hessians →DFLS [Zhang, Conn, Scheinberg]

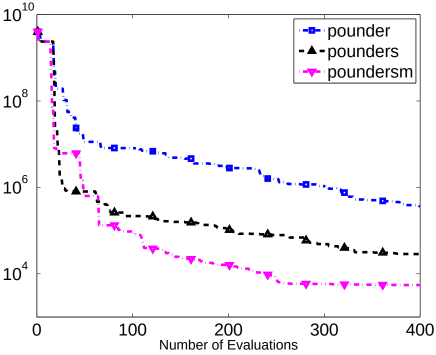
full Newton →POUNDERS [W., Moré]

POUNDERS for χ^2 (=Nonlinear Least Squares Calibration)

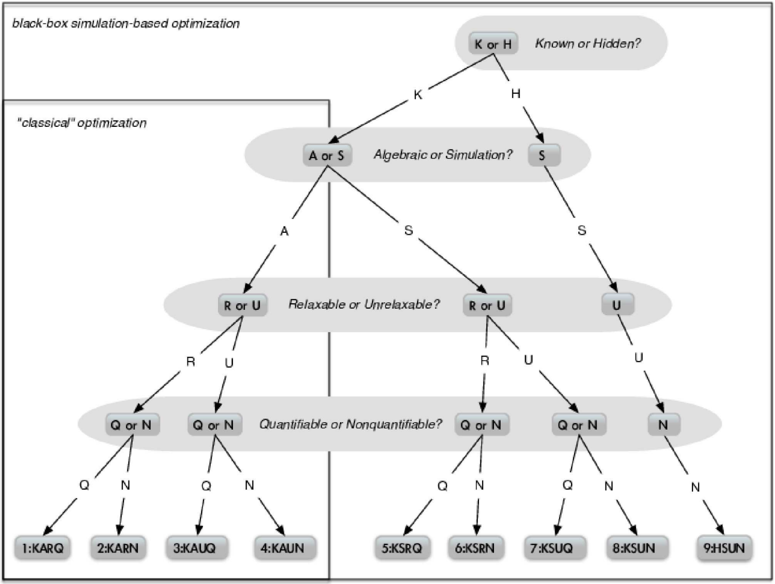
POUNDERS (in PETSc/TAO) well tested for calibration problems:

$$f(x) \propto \sum_{i,j} W_{i,j} \left(S(x;\theta^i) - d_i\right) \left(S(x;\theta^j) - d_j\right)$$





Constraints in Simulation-Based Optimization



[le Digabel, W.; 2017]; [Regis, W.; OMS, 2017]

Why Expressing Constraint Functions Matters

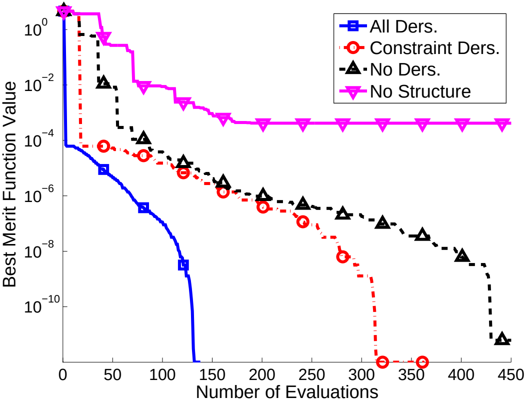
Augmented Lagrangian methods, $L_A(x, \lambda; \mu) = f(x) - \lambda^T c(x) + \frac{1}{\mu}\|c(x)\|^2$

$\min_x \{f(x) : c(x) = 0\}$

Four choices:

1. Penalize constraints
2. Treat c and f both as (separate) black boxes
3. Work with f and $\nabla_x c$
4. Have both $\nabla_x f$ and $\nabla_x c$

→ *With Slava Kungurtsev*



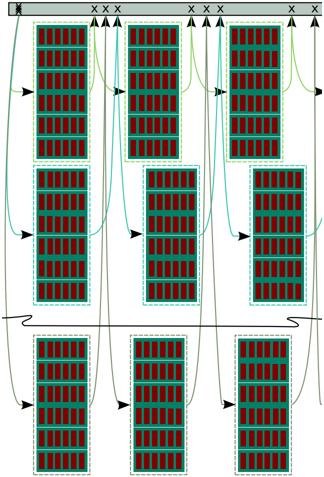
$n = 15$, 11 constraints

What is APOSMM?

Asynchronous Parallel Optimization Solver for Multiple Minima

- Better account for dynamic number of local runs
- Decouple local run from fixed resource
- Anticipate nontrivial Var[time $(f(x))$]

[Larson & W. Asynchronously Parallel Optimization Solver for Finding Multiple Minima, Math. Program. Comput., 2018.]



The (A)POSMM Algorithm

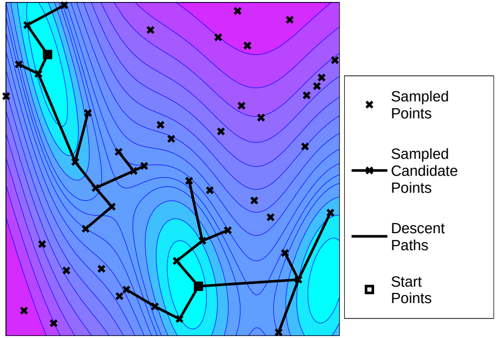
Repeat:

- Receive from worker(s) $w_\ell \in W$ that has evaluated its point
- If point was a sample point, update $r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\mathrm{vol}\,(\mathcal{D}) \frac{5\Gamma\left(1+\frac{n}{2}\right)\log(|\mathcal{S}_k|)}{|\mathcal{S}_k|}}$
- If point was a local optimization point, add subsequent point in the run (not in $\mathcal{H}_k$) to Q_L if not terminated
- Start run(s) at all point(s) now satisfying conditions, adding subsequent point from each run to Q_L
- Merge/collapse runs within Q_L
- Send point(s) from Q_L and/or $\mathcal{R}$ to worker(s)

W	Set of workers	(level of concurrency $|W|$)
$\mathcal{R}$	Stream of sample points (from $\mathcal{D}$)	
$\mathcal{S}_k$	Sample points after iteration k	
Q_L	Queue of local optimization points (needed by $\mathcal{A}$)	
$\mathcal{H}_k$	History after k evaluations	

Basic Idea: Multi Level Single Linkage (MLSL) Clustering

Where to start $\mathcal{A}$ in kth iteration [Rinnooy Kan & Timmer (MathProg, 1987)]



Ex.: It. 1 Exploration

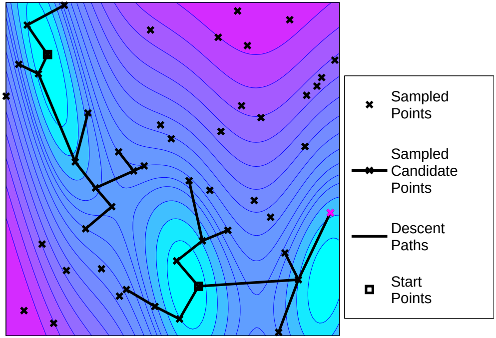
Start $\mathcal{A}$ at each sample point $x^i \in \mathcal{S}_k$ provided:

- $\mathcal{A}$ has not been started from x^i, and
- no other sample point $x^j \in \mathcal{S}_k$ with $f(x^j) < f(x^i)$ is within a distance

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\mathrm{vol}\,(\mathcal{D}) \frac{5\Gamma\left(1+\frac{n}{2}\right)\log(kN)}{kN}},$$

Basic Idea: Multi Level Single Linkage (MLSL) Clustering

Where to start $\mathcal{A}$ in kth iteration [Rinnooy Kan & Timmer (MathProg, 1987)]



Ex.: It. 1 Exploration

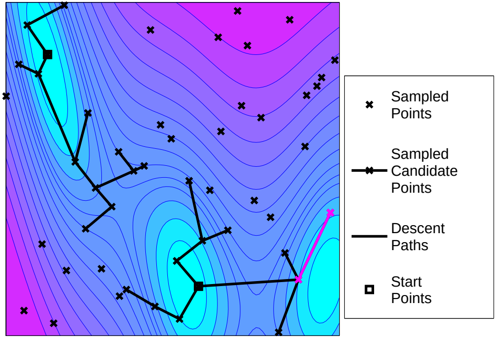
Start $\mathcal{A}$ at each sample point $x^i \in \mathcal{S}_k$ provided:

- $\mathcal{A}$ has not been started from x^i, and
- no other sample point $x^j \in \mathcal{S}_k$ with $f(x^j) < f(x^i)$ is within a distance

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\mathrm{vol}\,(\mathcal{D}) \frac{5\Gamma\left(1+\frac{n}{2}\right)\log(kN)}{kN}},$$

Basic Idea: Multi Level Single Linkage (MLSL) Clustering

Where to start $\mathcal{A}$ in kth iteration [Rinnooy Kan & Timmer (MathProg, 1987)]



Ex.: It. 1 Exploration

Start $\mathcal{A}$ at each sample point $x^i \in \mathcal{S}_k$ provided:

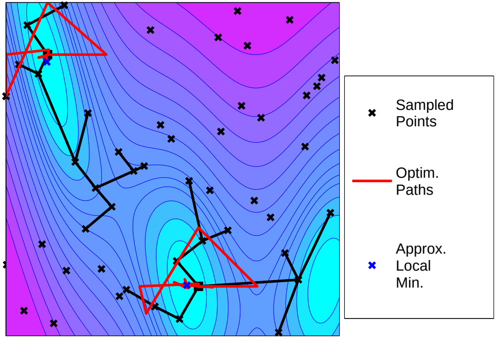
- $\mathcal{A}$ has not been started from x^i, and
- no other sample point $x^j \in \mathcal{S}_k$ with $f(x^j) < f(x^i)$ is within a distance

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\mathrm{vol}\,(\mathcal{D}) \frac{5\Gamma\left(1+\frac{n}{2}\right)\log(kN)}{kN}},$$

Thm [RK-T]- With probability 1, MLSL will start finitely many local runs.

Basic Idea: Multi Level Single Linkage (MLSL) Clustering

Where to start $\mathcal{A}$ in kth iteration [Rinnooy Kan & Timmer (MathProg, 1987)]



Ex.: It. 1 Refinement

Start $\mathcal{A}$ at each sample point $x^i \in \mathcal{S}_k$ provided:

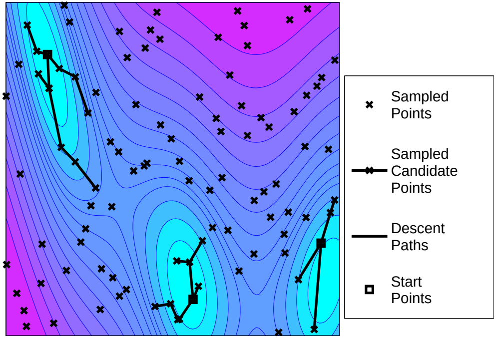
- $\mathcal{A}$ has not been started from x^i, and
- no other sample point $x^j \in \mathcal{S}_k$ with $f(x^j) < f(x^i)$ is within a distance

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\mathrm{vol}\,(\mathcal{D}) \frac{5\Gamma\left(1+\frac{n}{2}\right)\log(kN)}{kN}},$$

Thm [RK-T]- With probability 1, MLSL will start finitely many local runs.

Basic Idea: Multi Level Single Linkage (MLSL) Clustering

Where to start $\mathcal{A}$ in kth iteration [Rinnooy Kan & Timmer (MathProg, 1987)]



Ex.: It. 2 Exploration

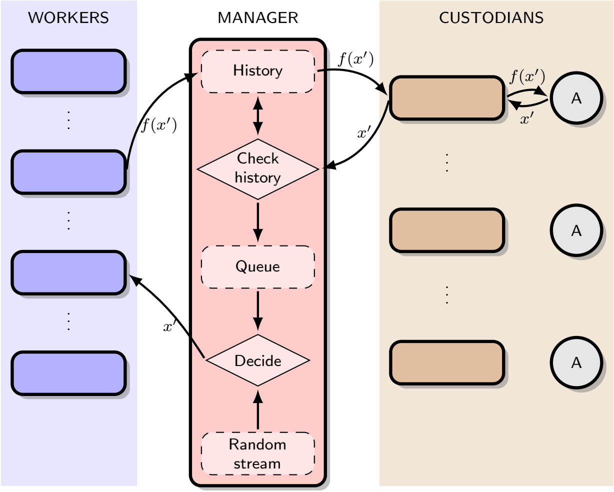
Start $\mathcal{A}$ at each sample point $x^i \in \mathcal{S}_k$ provided:

- $\mathcal{A}$ has not been started from x^i, and
- no other sample point $x^j \in \mathcal{S}_k$ with $f(x^j) < f(x^i)$ is within a distance

$$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\text{vol}\left(\mathcal{D}\right) \frac{5\Gamma\left(1 + \frac{n}{2}\right) \log(kN)}{kN}},$$

Thm [RK-T]- With probability 1, MLSL will start finitely many local runs.

(A)POSMM Framework

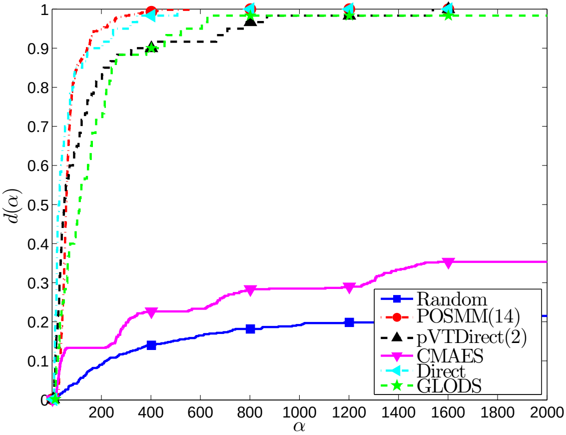


Data Profiles: Ability to Find Approximate Global Minimizer

600 GKLS problems

`(A)POSMM`

- Makes rapid progress to f_G
- Outperforms other algorithms (even while demanding 14-fold concurrency) evaluations



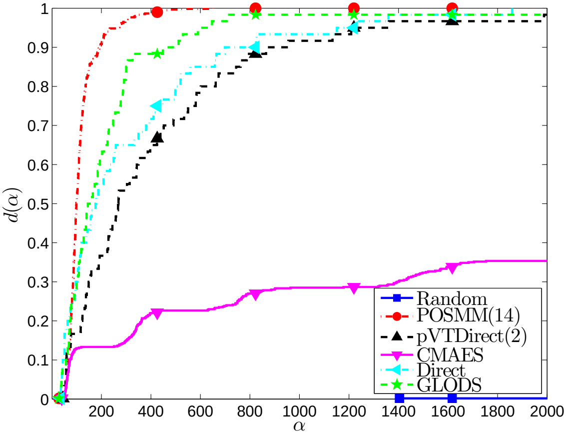
$$\tau = 10^{-2}$$
$$f(x) - f_G \leq (1-\tau)\left(f(x^0) - f_G\right)$$

Data Profiles: Ability to Find Approximate Global Minimizer

600 GKLS problems

(A)POSMM

- Makes rapid progress to f_G
- Outperforms other algorithms (even while demanding 14-fold concurrency) evaluations



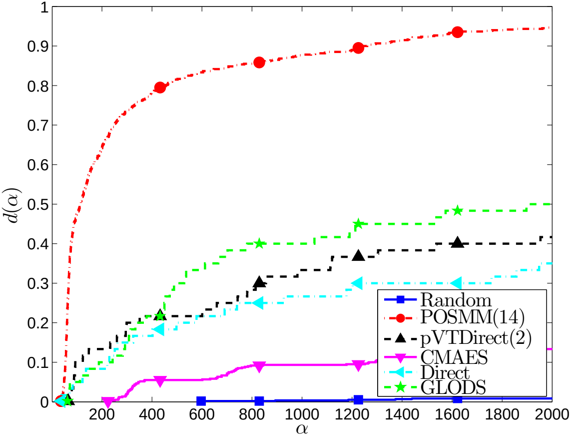
$$\tau = 10^{-5}$$
$$f(x) - f_G \leq (1-\tau)\left(f(x^0) - f_G\right)$$

Data Profiles: Ability to Find j Best Minimizers

600 GKLS problems

(A)POSMM

- Designed to find more than just the global minimizer
- Extends lead for tighter tolerances



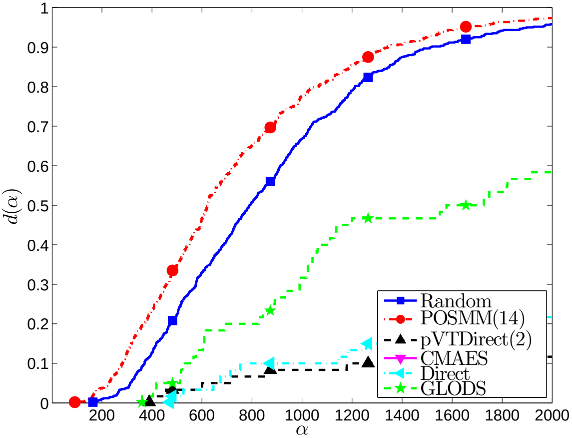
distance $\tau = 10^{-5}$, $j = 2$ minimizers

Data Profiles: Ability to Find j Best Minimizers

600 GKLS problems

(A)POSMM

- Designed to find more than just the global minimizer
- Extends lead for tighter tolerances



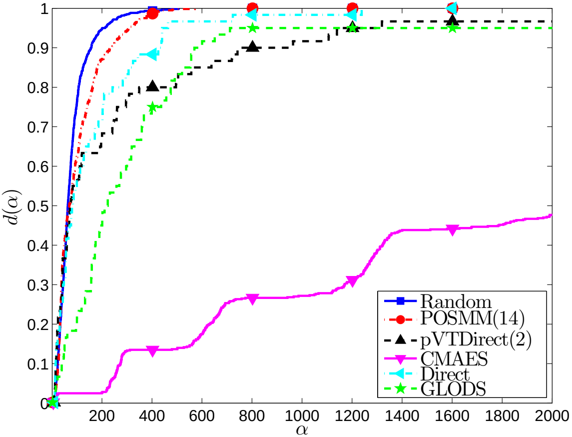
distance $\tau = 10^{-3}$, $j = 7$ minimizers

Data Profiles: Ability to Find j Best Minimizers

600 GKLS problems

(A)POSMM

- Designed to find more than just the global minimizer
- Extends lead for tighter tolerances



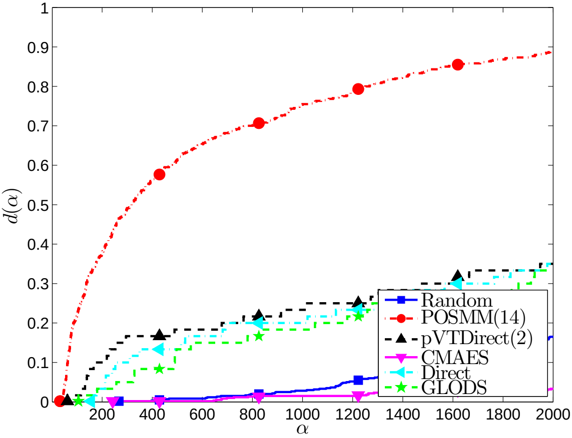
distance $\tau = 10^{-2}$, $j = 3$ minimizers

Data Profiles: Ability to Find j Best Minimizers

600 GKLS problems

(A)POSMM

- Designed to find more than just the global minimizer
- Extends lead for tighter tolerances



distance $\tau = 10^{-4}$, $j = 3$ minimizers

Argonne/Optimization Milestones in ComPASS-4

Activity	**Institution(s)**	**Sec**	**Year**
Develop API for POPAS prototype	ANL, FNAL, UCLA	§ 2.4	1
Identify optimizable elements in the MARS and Synergia PIP-II models; connect with POPAS prototype	FNAL, ANL	§ 2.1.1	2
Use MARS-Synergia-POPAS prototype for preliminary optimization	FNAL, ANL	§ 2.1.1	3
Include prototype of structure-exploiting optimization algorithm for standard PIC/QuickPIC simulations; enable basic execution of all ComPASS-4 codes in POPAS	ANL, FNAL, UCLA	§ 2.4	3
Link numerical optimization algorithm to POPAS; Remove file I/O layer from POPAS	ANL, FNAL, UCLA	§ 2.4	3
Connect IOTA Synergia model with POPAS	FNAL, ANL	§ 2.1.1	3
Release POPAS; apply POPAS to standard PIC/QuickPIC and Synergia	ANL, FNAL, UCLA	§ 2.4	4
Refine MARS-Synergia-POPAS	FNAL, ANL	§ 2.1.1	4
Apply IOTA Synergia-POPAS	FNAL, ANL	§ 2.1.1	4
Carry out parameter optimization on PWFA-LC relevant parameters using QuickPIC	UCLA, FNAL, ANL	§ 2.5.2	5