Derivative-Free Optimization with Convex Constraints

Joint work with Matthew Hough (Waterloo)

Lindon Roberts, Australian National University (lindon.roberts@anu.edu.au)

Workshop on Optimisation, Metric Bounds, Approximation and Transversality (WOMBAT)
14 December 2021

Outline

1. **Unconstrained derivative-free optimization (DFO)**
2. Convex constraints: algorithm and interpolation geometry
3. Application to least-squares & numerical results

Derivative-Free Optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

- Objective f nonlinear, nonconvex, structure unknown

Derivative-Free Optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

- Objective f nonlinear, nonconvex, structure unknown
- Standard methods locally approximate f by quadratic models (e.g. Taylor series)
- How to calculate derivatives of f to build model?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation (backpropagation)

Derivative-Free Optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

- Objective f nonlinear, nonconvex, structure unknown
- Standard methods locally approximate f by quadratic models (e.g. Taylor series)
- How to calculate derivatives of f to build model?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation (backpropagation)
- Difficulties when function evaluation is black-box, noisy and/or expensive

Derivative-Free Optimization

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

- Objective f nonlinear, nonconvex, structure unknown
- Standard methods locally approximate f by quadratic models (e.g. Taylor series)
- How to calculate derivatives of f to build model?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation (backpropagation)
- Difficulties when function evaluation is black-box, noisy and/or expensive
- Alternative — derivative-free optimization (DFO) [aka "zero-order methods"]
  - Applications in finance, climate, engineering, machine learning, ...

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

$$f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{s}$$

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

$$f(\mathbf{x}_k + \mathbf{s}) \approx m_k(\mathbf{s}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{s} + \frac{1}{2}\mathbf{s}^T \nabla^2 f(\mathbf{x}_k)\mathbf{s}$$

- Instead, approximate

$$f(\mathbf{x}_k + \mathbf{s}) \approx m_k(\mathbf{s}) = f(\mathbf{x}_k) + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2}\mathbf{s}^T H_k \mathbf{s}$$

  and find $\mathbf{g}_k$ and H_k without using derivatives

Model-Based DFO — Basic Ideas

Many approaches: model-based, gradient sampling, direct search, Bayesian, ...

- Classically (e.g. Newton's method),

$$f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{s}$$

- Instead, approximate

$$f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k \boldsymbol{s}$$

  and find $\boldsymbol{g}_k$ and H_k without using derivatives
- How? Interpolate f over a set of points
- Geometry of points good $\Longrightarrow$ interpolation model Taylor-accurate $\Longrightarrow$ convergence

[Powell, 2003; Conn, Scheinberg & Vicente, 2009]

Model-Based DFO — Basic Ideas

Implement in trust-region method:

1. Build interpolation model $m_k(\boldsymbol{s})$
2. Minimize model inside trust region

$$\boldsymbol{s}_k = \arg\min_{\boldsymbol{s}\in\mathbb{R}^n} m_k(\boldsymbol{s}) \quad \text{s.t.} \quad \|\boldsymbol{s}\|_2 \leq \Delta_k.$$

3. Accept/reject step and adjust Δ_k based on quality of new point $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k, & \text{if sufficient decrease,} & \longleftarrow \text{(maybe increase } \Delta_k\text{)} \\ \boldsymbol{x}_k, & \text{otherwise.} & \longleftarrow \text{(decrease } \Delta_k\text{)} \end{cases}$$

4. Update interpolation set: add $\boldsymbol{x}_k + \boldsymbol{s}_k$ to interpolation set
5. If needed, ensure new interpolation set is 'good'

Theoretical Questions

1. What is a 'good' interpolation set/model?
2. What convergence/complexity guarantees do we have?

Model-Based DFO — Theory

Theoretical Questions

1. What is a 'good' interpolation set/model?
2. What convergence/complexity guarantees do we have?

[Conn, Scheinberg & Vicente, 2009]

An interpolation model $f(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{s})$ is fully linear if

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| \leq \kappa \Delta_k^2,$$
$$\|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 \leq \kappa \Delta_k,$$

for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$ (c.f. linear Taylor series).

Theoretical Questions

1. What is a 'good' interpolation set/model?
2. What convergence/complexity guarantees do we have?

[Conn, Scheinberg & Vicente, 2009]

An interpolation set is Λ-poised if

$$\max_t \max_{\|\boldsymbol{s}\|_2 \leq \Delta_k} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda,$$

where ℓ_t is the t-th Lagrange polynomial for the interpolation set (i.e. $\ell_t(\boldsymbol{y}_s) = \delta_{s,t}$).

Theorem

If the interpolation set is Λ-poised and contained in $B(\boldsymbol{x}_k, \Delta_k)$, then the corresponding interpolation model is fully linear with $\kappa = \mathcal{O}(\Lambda)$. (+ dependencies on n, f)

Theoretical Questions

1. What is a 'good' interpolation set/model?
2. What convergence/complexity guarantees do we have?

[Conn, Scheinberg & Vicente, 2009]

Convergence & worst-case complexity for nonconvex functions (match derivative-based trust-region methods).

Theorem

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\|_2 = 0$. Furthermore, we achieve $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations. (+ dependencies on κ, f)

Outline

1. Unconstrained derivative-free optimization (DFO)
2. **Convex constraints: algorithm and interpolation geometry**
3. Application to least-squares & numerical results

Convex Constraints

Now consider the setting

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \text{subject to } x \in \mathcal{C},$$

where $\mathcal{C} \subseteq \mathbb{R}^n$ is a closed, convex set with nonempty interior.

Require:

- Strictly feasible algorithm: never evaluate f at points outside $\mathcal{C}$;
- Access to $\mathcal{C}$ is only through a (cheap) projection operator

Examples: $\mathbb{R}^n$, bound constraints, half-plane, Euclidean ball, ...

Convex Constraints

Existing work:

- Unrelaxable constraints: only for simple cases, no convergence theory
  - Bounds [Powell, 2009; Wild, 2009; Gratton et al., 2011]
  - Linear inequalities [Gumma, Hashim & Ali, 2014; Powell, 2015]
- Convex constraints with projections (our setting): [Conejo et al., 2013]
  - Convergence, no complexity
  - Assume models always fully linear (but how to achieve?)
- Derivative-based complexity analysis [Cartis, Gould & Toint, 2012]

Convex Constraints

Existing work:

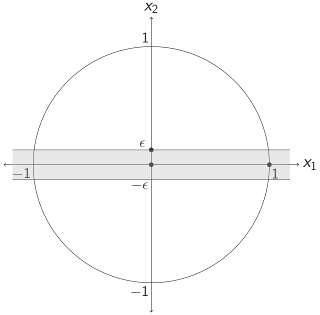
- Unrelaxable constraints: only for simple cases, no convergence theory
  - Bounds [Powell, 2009; Wild, 2009; Gratton et al., 2011]
  - Linear inequalities [Gumma, Hashim & Ali, 2014; Powell, 2015]
- Convex constraints with projections (our setting): [Conejo et al., 2013]
  - Convergence, no complexity
  - Assume models always fully linear (but how to achieve?)
- Derivative-based complexity analysis [Cartis, Gould & Toint, 2012]

Key Problem

Model-based methods are more challenging to design in the presence of unrelaxable constraints because enforcing guarantees of model quality... can be difficult. For a fixed value of κ..., it may be impossible to obtain a fully linear model using only feasible points. [Larson, Menickelly & Wild, 2019]

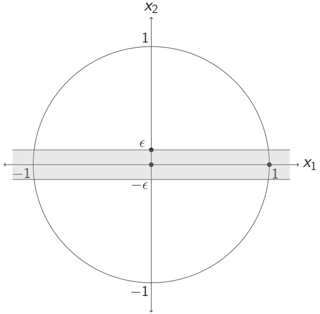
Convex Constraints — The Basic Problem

Why can't we achieve fully linear models using only feasible points?



Use $\mathcal{C} = \{(x_1, x_2) \in \mathbb{R}^2 : |x_2| \leq \epsilon\}$ with interpolation points $(0, 0)$, $(1, 0)$ and $(0, \epsilon)$. Get $\Lambda = \mathcal{O}(\epsilon^{-1})$ $\implies$ large interpolation errors. Cannot be improved using feasible points.

Why can't we achieve fully linear models using only feasible points?



Use $\mathcal{C} = \{(x_1, x_2) \in \mathbb{R}^2 : |x_2| \leq \epsilon\}$ with interpolation points $(0,0)$, $(1,0)$ and $(0,\epsilon)$. Get $\Lambda = \mathcal{O}(\epsilon^{-1}) \implies$ large interpolation errors. Cannot be improved using feasible points.

Note: $\Lambda = \mathcal{O}(1)$ if only consider $|\ell_t(\boldsymbol{x}_k + \boldsymbol{s})|$ inside the feasible region!

Convex Constraints — Geometry

Old definition of Λ-poised set:

$$\max_t \max_{\|\boldsymbol{s}\|_2 \leq \Delta_k} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda.$$

Gives very large values of Λ if all interpolation points must be feasible.

Convex Constraints — Geometry

Old definition of Λ-poised set:

$$\max_{t} \max_{\|\boldsymbol{s}\|_2 \leq \Delta_k} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda.$$

Gives very large values of Λ if all interpolation points must be feasible.

New definition:

$$\max_{t} \max_{\substack{\boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C} \\ \|\boldsymbol{s}\|_2 \leq \Delta_k}} |\ell_t(\boldsymbol{x}_k + \boldsymbol{s})| \leq \Lambda.$$

- Only care about Lagrange polynomial size inside the feasible region (since the algorithm will never look elsewhere).
- Gives smaller values of Λ — better interpolation error?

Convex Constraints — Geometry

Fully linear: for all $\|\boldsymbol{s}\|_2 \leq \Delta_k$

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| \leq \kappa \Delta_k^2,$$
$$\|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 \leq \kappa \Delta_k.$$

This is stronger than we really need!

Convex Constraints — Geometry

Fully linear: for all $\|\boldsymbol{s}\|_2 \le \Delta_k$

$$|f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| \le \kappa \Delta_k^2,$$
$$\|\nabla f(\boldsymbol{x}_k + \boldsymbol{s}) - \nabla m_k(\boldsymbol{s})\|_2 \le \kappa \Delta_k.$$

This is stronger than we really need! New definition adapted to $\mathcal{C}$:

$$\max_{\substack{\boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C} \\ \|\boldsymbol{s}\|_2 \le \Delta_k}} |f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| \le \kappa \Delta_k^2,$$

$$\max_{\substack{\boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C} \\ \|\boldsymbol{s}\|_2 \le 1}} |(\nabla f(\boldsymbol{x}_k) - \nabla m_k(0))^T \boldsymbol{s}| \le \kappa \Delta_k.$$

Theorem (Hough & R., 2021)

If the interpolation set is contained in $B(\boldsymbol{x}_k, \Delta_k) \cap \mathcal{C}$ and [new] Λ-poised, then the corresponding linear interpolation model is [new] fully linear with $\kappa = \mathcal{O}(\Lambda)$.

Convex Constraints — Algorithm

Algorithm almost identical to unconstrained case:

1. Build interpolation model $m_k(\boldsymbol{s})$
2. Minimize model inside trust region

$$\boldsymbol{s}_k = \arg\min_{\boldsymbol{s}\in\mathbb{R}^n} m_k(\boldsymbol{s}) \quad \text{s.t.} \quad \|\boldsymbol{s}\|_2 \leq \Delta_k \quad \text{and } \boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C}.$$

3. Accept/reject step and adjust Δ_k based on quality of new point $f(\boldsymbol{x}_k + \boldsymbol{s}_k)$

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k + \boldsymbol{s}_k, & \text{if sufficient decrease,} \qquad \longleftarrow \text{(maybe increase } \Delta_k\text{)} \\ \boldsymbol{x}_k, & \text{otherwise.} \qquad \longleftarrow \text{(decrease } \Delta_k\text{)} \end{cases}$$

4. Update interpolation set: add $\boldsymbol{x}_k + \boldsymbol{s}_k$ to interpolation set
5. If needed, ensure new interpolation set is [new] Λ-poised

Convex Constraints — Convergence/Complexity

For convergence results, first need to ask

Question

What is a suitable measure of stationarity?

Convex Constraints — Convergence/Complexity

For convergence results, first need to ask

Question

What is a suitable measure of stationarity?

$$\pi^f(\boldsymbol{x}) := \left| \min_{\substack{\boldsymbol{x}+\boldsymbol{s}\in\mathcal{C} \\ \|\boldsymbol{s}\|_2\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{s} \right|$$

Convex Constraints — Convergence/Complexity

For convergence results, first need to ask

Question

What is a suitable measure of stationarity?

$$\pi^f(\boldsymbol{x}) := \left| \min_{\substack{\boldsymbol{x}+\boldsymbol{s}\in\mathcal{C} \\ \|\boldsymbol{s}\|_2\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{s} \right|$$

[Conn, Gould & Toint, 2000]

Useful properties:

- $\pi^f(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$
- $\pi^f(\boldsymbol{x}^*) = 0$ if and only if $\boldsymbol{x}^*$ is a KKT point
- If $\mathcal{C} = \mathbb{R}^n$, then $\pi^f(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|_2$
- $\pi^f(\boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{x}$ (if ∇f is Lipschitz) [Cartis, Gould & Toint, 2012]

Convex Constraints — Convergence/Complexity

For convergence results, first need to ask

Question

What is a suitable measure of stationarity?

$$\pi^f(\boldsymbol{x}) := \left| \min_{\substack{\boldsymbol{x}+\boldsymbol{s}\in\mathcal{C} \\ \|\boldsymbol{s}\|_2\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{s} \right|$$

Useful properties: [Conn, Gould & Toint, 2000]

- $\pi^f(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$
- $\pi^f(\boldsymbol{x}^*) = 0$ if and only if $\boldsymbol{x}^*$ is a KKT point
- If $\mathcal{C} = \mathbb{R}^n$, then $\pi^f(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|_2$
- $\pi^f(\boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{x}$ (if ∇f is Lipschitz) [Cartis, Gould & Toint, 2012]
- If m_k is [new] fully linear, then $|\pi^f(\boldsymbol{x}_k) - \pi^{m_k}(\boldsymbol{x}_k)| \leq \kappa\Delta_k$ [Hough & R., 2021]

We can match the unconstrained convergence & complexity results:

Theorem (Hough & R., 2021)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi^f(\boldsymbol{x}_k) = 0$. Furthermore, we achieve $\pi^f(\boldsymbol{x}_k) \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations.

Convex Constraints — Convergence/Complexity

We can match the unconstrained convergence & complexity results:

Theorem (Hough & R., 2021)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi^f(\boldsymbol{x}_k) = 0$. Furthermore, we achieve $\pi^f(\boldsymbol{x}_k) \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations.

Requires the existence of procedures to:

- Verify if a model is fully linear
- If a model is not fully linear, change the interpolation set to make it fully linear

For our new definition of Λ-poisedness, can use (almost) the same approach as for unconstrained case.

Outline

1. Unconstrained derivative-free optimization (DFO)
2. Convex constraints: algorithm and interpolation geometry
3. **Application to least-squares & numerical results**

DFO for Least-Squares — Basic Framework

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2, \qquad \boldsymbol{r}(\boldsymbol{x}) \in \mathbb{R}^m$$

Classical Gauss-Newton

Derivative-Free Gauss-Newton

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2, \qquad \boldsymbol{r}(\boldsymbol{x}) \in \mathbb{R}^m$$

Classical Gauss-Newton | **Derivative-Free Gauss-Newton**

- Linearize $\boldsymbol{r}$ at $\boldsymbol{x}_k$ using Jacobian

$$\boldsymbol{r}(\boldsymbol{x}_k+\boldsymbol{s}) \approx \boldsymbol{m}_k(\boldsymbol{s}) = \boldsymbol{r}(\boldsymbol{x}_k)+J(\boldsymbol{x}_k)\boldsymbol{s}$$

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2, \qquad \boldsymbol{r}(\boldsymbol{x}) \in \mathbb{R}^m$$

Classical Gauss-Newton

- Linearize $\boldsymbol{r}$ at $\boldsymbol{x}_k$ using Jacobian

$$\boldsymbol{r}(\boldsymbol{x}_k+\boldsymbol{s}) \approx \boldsymbol{m}_k(\boldsymbol{s}) = \boldsymbol{r}(\boldsymbol{x}_k)+J(\boldsymbol{x}_k)\boldsymbol{s}$$

Derivative-Free Gauss-Newton

- Jacobian not available: use

$$\boldsymbol{m}_k(\boldsymbol{s}) = \boldsymbol{r}(\boldsymbol{x}_k) + J_k\boldsymbol{s}$$

- Find J_k using linear interpolation [Cartis & R., 2019]

DFO for Least-Squares — Basic Framework

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|_2^2, \qquad \boldsymbol{r}(\boldsymbol{x}) \in \mathbb{R}^m$$

Classical Gauss-Newton

- Linearize $\boldsymbol{r}$ at $\boldsymbol{x}_k$ using Jacobian

$$\boldsymbol{r}(\boldsymbol{x}_k+\boldsymbol{s}) \approx \boldsymbol{m}_k(\boldsymbol{s}) = \boldsymbol{r}(\boldsymbol{x}_k)+J(\boldsymbol{x}_k)\boldsymbol{s}$$

Derivative-Free Gauss-Newton

- Jacobian not available: use

$$\boldsymbol{m}_k(\boldsymbol{s}) = \boldsymbol{r}(\boldsymbol{x}_k) + J_k\boldsymbol{s}$$

- Find J_k using linear interpolation [Cartis & R., 2019]

In both cases, get a local quadratic model

$$f(\boldsymbol{x}_k+\boldsymbol{s}) \approx m_k(\boldsymbol{s}) = \frac{1}{2}\|\boldsymbol{m}_k(\boldsymbol{s})\|_2^2$$

New: Linear interpolation with feasible points gives fully linear quadratic models

Least-Squares Implementation

New changes implemented in state-of-the-art solver DFO-LS [Cartis et al., 2019]

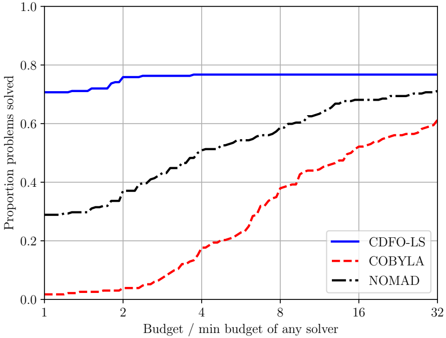
- Use FISTA to compute search direction (subject to feasibility & trust-region constraint) + Dykstra's algorithm to project onto $B(\boldsymbol{x}_k, \Delta_k) \cap \mathcal{C}$
- Github: `numericalalgorithmsgroup/dfols`

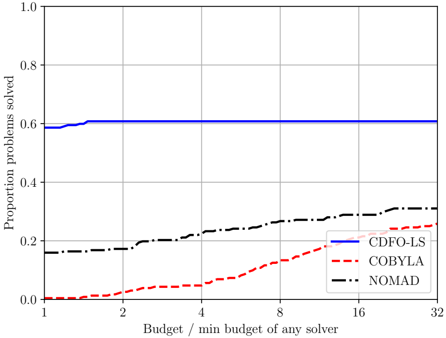
Test on collection of 58 low-dimensional least-squares problems with box/ball/halfspace constraints.

Few codes to test against (none using the least-squares structure)!.

- NOMAD: direct search DFO, model constraints using extreme barrier (i.e. $f(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{C}$) [Le Digabel, 2011]
- COBYLA: model-based DFO with (derivative-free) inequality constraints [Powell, 1994]

Performance profiles at different accuracy levels





Low accuracy, $\tau = 10^{-1}$

High accuracy, $\tau = 10^{-5}$

[% problems solved vs. # objective evals; higher is better]

Conclusions & Future Work

Conclusions

- General model-based DFO method for convex-constrained problems
- Match/generalize existing convergence & complexity results
- Developed comprehensive new theory of Λ-poisedness/full linearity
  - Currently only for (composite) linear interpolation
- New software for least-squares problems

Future Work

- Second-order theory
- Generalize interpolation theory to quadratic interpolation

[arXiv:2111.05443, Github: `numericalalgorithmsgroup/dfols`]

References i

C. Cartis, J. Fiala, B. Marteau, and L. Roberts, *Improving the flexibility and robustness of model-based derivative-free optimization solvers*, ACM Transactions on Mathematical Software, 45 (2019), pp. 1–41.

C. Cartis, N. I. M. Gould, and P. L. Toint, *An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity*, IMA Journal of Numerical Analysis, 32 (2012), pp. 1662–1695.

C. Cartis and L. Roberts, *A derivative-free Gauss-Newton method*, Mathematical Programming Computation, 11 (2019), pp. 631–674.

P. Conejo, E. Karas, L. Pedroso, A. Ribeiro, and M. Sachine, *Global convergence of trust-region algorithms for convex constrained minimization without derivatives*, Applied Mathematics and Computation, 220 (2013), pp. 324–330.

A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, vol. 1 of MPS-SIAM Series on Optimization, MPS/SIAM, Philadelphia, 2000.

A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, vol. 8 of MPS-SIAM Series on Optimization, MPS/SIAM, Philadelphia, 2009.

S. Gratton, P. L. Toint, and A. Tröltzsch, *An active-set trust-region method for derivative-free nonlinear bound-constrained optimization*, Optimization Methods and Software, 26 (2011), pp. 873–894.

E. A. E. Gumma, M. H. A. Hashim, and M. M. Ali, *A derivative-free algorithm for linearly constrained optimization problems*, Computational Optimization and Applications, 57 (2014), pp. 599–621.

M. Hough and L. Roberts, *Model-based derivative-free methods for convex-constrained optimization*, arXiv preprint arXiv:2111.05443, (2021).

J. Larson, M. Menickelly, and S. M. Wild, *Derivative-free optimization methods*, Acta Numerica, 28 (2019), pp. 287–404.

S. Le Digabel, *Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm*, ACM Transactions on Mathematical Software, 37 (2011).

M. J. D. Powell, *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, Springer Netherlands, Dordrecht, 1994, pp. 51–67.

———, *On trust region methods for unconstrained minimization without derivatives*, Mathematical Programming, 97 (2003), pp. 605–623.

———, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Tech. Rep. DAMTP 2009/NA06, University of Cambridge, 2009.

———, *On fast trust region methods for quadratic models with linear constraints*, Mathematical Programming Computation, 7 (2015), pp. 237–267.

S. M. Wild, *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*, PhD thesis, Cornell University, 2009.