Derivative-Free Optimization with Convex Constraints

Joint work with Lindon Roberts (ANU $\rightarrow$ University of Sydney)

Matthew Hough, University of Waterloo (`mhough@uwaterloo.ca`)

ICCOPT 2022, July 25-28, 2022

Outline

1. **Introduction to DFO trust-region methods**
2. Extending to convex constraints
3. Application to least-squares problems

The Problem

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

- $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable and possibly nonconvex
- Assume we cannot evaluate $\nabla f(\boldsymbol{x})$
  - Black-box
  - Noisy
  - Computationally expensive
- Applications: climate modelling, experimental design, machine learning, etc
- Seeking a local minimizer (approx. stationary point: $\|\nabla f(\boldsymbol{x}^*)\| \leq \epsilon$)

Model-Based DFO

- Classic approach:

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

- Instead, approximate:

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

- Find $\boldsymbol{g}_k$ and H_k by interpolating f over a set of points

Model-Based DFO: Algorithm

1. Build local interpolation model $m_k(\boldsymbol{s})$
2. Minimize the model within the trust-region Δ_k to get the step

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

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

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

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

Model-Based DFO: Theory

You may be wondering...

1. What does it mean for our interpolation model to be a 'good approximation'?
2. What convergence/complexity guarantees do we have?

Model-Based DFO: Theory

You may be wondering...

1. What does it mean for our interpolation model to be a 'good approximation'?
2. What convergence/complexity guarantees do we have?

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

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

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

[Conn, Scheinberg & Vicente, 2009]

Model-Based DFO: Theory

You may be wondering...

1. What does it mean for our interpolation model to be a 'good approximation'?
2. What convergence/complexity guarantees do we have?

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 $\ell_t(\boldsymbol{y}_s) = \delta_{s,t}$.

Theorem

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

[Conn, Scheinberg & Vicente, 2009]

Model-Based DFO: Theory

You may be wondering...

1. What does it mean for our interpolation model to be a 'good approximation'?
2. What convergence/complexity guarantees do we have?

Convergence and worst-case complexity for nonconvex functions match what we have for the derivative-based case.

Theorem (convergence)

Suppose f has Lipschitz continuous gradient and is bounded below. Then $\lim_{k\to\infty} ||\nabla f(\boldsymbol{x}_k)||_2 = 0$.

[Conn, Scheinberg & Vicente, 2009]

Model-Based DFO: Theory

You may be wondering...

1. What does it mean for our interpolation model to be a 'good approximation'?
2. What convergence/complexity guarantees do we have?

Convergence and worst-case complexity for nonconvex functions match what we have for the derivative-based case.

Theorem (convergence)

Suppose f has Lipschitz continuous gradient and is bounded below. Then $\lim_{k\to\infty} ||\nabla f(\boldsymbol{x}_k)||_2 = 0$.

Theorem (complexity)

Under the same assumptions as above, we achieve $||\nabla f(\boldsymbol{x}_k)||_2 \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations. (+ dependencies on n, f)

[Conn, Scheinberg & Vicente, 2009]

Outline

1. Introduction to DFO trust-region methods

2. **Extending to convex constraints**

3. Application to least-squares problems

The Problem

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

- $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable and possibly nonconvex
- Assume we cannot evaluate $\nabla f(\boldsymbol{x})$
- $\mathcal{C} \subseteq \mathbb{R}^n$ has nonempty interior, closed, and convex
  - 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, ...

Existing work

- Unrelaxable constraints
  - Only done 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 projection [Conejo et al., 2013]
  - Convergence, no complexity
  - Assume models are always fully linear
- Derivative-based complexity analysis [Cartis, Gould & Toint, 2012]

Existing work

- Unrelaxable constraints
  - Only done 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 projection [Conejo et al., 2013]
  - Convergence, no complexity
  - Assume models are always fully linear
- Derivative-based complexity analysis [Cartis, Gould & Toint, 2012]

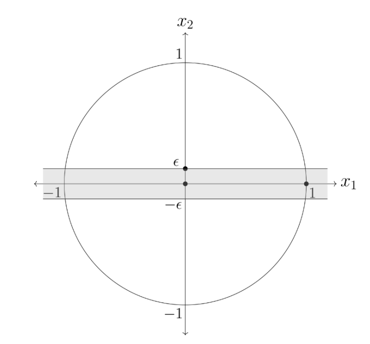
What is the problem? (Larson, Menickelly & Wild, 2019)

Model-based methods are more challenging to design in the presence of unrelaxable constraints because enforcing guarantees of model quality such as full linearity can be difficult. For fixed κ_{ef}, κ_{eg} it may be impossible to obtain a fully linear model using only feasible points.

What can we do?

Convex Constraints: The Problem

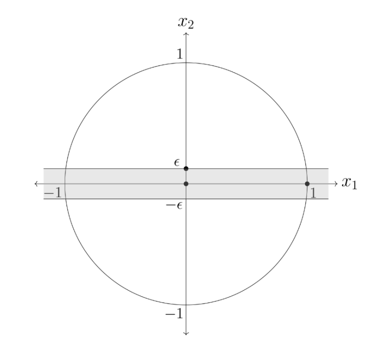
- $\mathcal{C} = \{(x_1, x_2) \ : \ |x_2| \leq \epsilon\} \subseteq \mathbb{R}^2$
- $Y = \{(0,0), (1,0), (0,\epsilon)\} \subseteq B(\mathbf{0}, 1)$



In $B(\mathbf{0}, 1)$, points are Λ-poised with $\Lambda = \mathcal{O}(\epsilon^{-1}) \implies$ large interpolation error. We cannot improve this using only feasible points.

Convex Constraints: The Problem

- $\mathcal{C} = \{(x_1, x_2) \ : \ |x_2| \leq \epsilon\} \subseteq \mathbb{R}^2$
- $Y = \{(0,0), (1,0), (0,\epsilon)\} \subseteq B(\mathbf{0}, 1)$



In $B(\mathbf{0}, 1)$, points are Λ-poised with $\Lambda = \mathcal{O}(\epsilon^{-1}) \implies$ large interpolation error. We cannot improve this using only feasible points.

If we only consider $|\ell_t(\boldsymbol{x}_k + \boldsymbol{s})|$ inside the feasible region $\implies \Lambda = \mathcal{O}(1)$.

Convex Constraints: Geometry

Recall the old definition of a Λ-poised set:

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

Convex Constraints: Geometry

Recall the old definition of a Λ-poised set:

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

New definition:

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

- We only care about the magnitude of Lagrange polynomials inside the feasible region.
- Gives smaller values of Λ.

Convex Constraints: Geometry

Recall the old definition of a fully linear model:

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

This is unnecessarily strong for our case!

Convex Constraints: Geometry

Recall the old definition of a fully linear model:

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

This is unnecessarily strong for our case!

New definition:

$$\max_{\substack{\boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C} \\ ||\boldsymbol{s}||_2 \leq \Delta_k}} |f(\boldsymbol{x}_k + \boldsymbol{s}) - m_k(\boldsymbol{s})| \leq \kappa_{ef} \Delta_k^2,$$
$$\max_{\substack{\boldsymbol{x}_k + \boldsymbol{s} \in \mathcal{C} \\ ||\boldsymbol{s}||_2 \leq 1}} ||\left(\nabla f(\boldsymbol{x}_k) - \nabla m_k(\mathbf{0})\right)^T s||_2 \leq \kappa_{eg} \Delta_k$$

Theorem (Hough & Roberts, 2021)

If the set of interpolation points is contained in $B(\boldsymbol{x}_k, \Delta_k) \cap \mathcal{C}$ and is Λ-poised, then the corresponding **linear** interpolation model is fully linear with $\kappa_{ef}, \kappa_{eg} = \mathcal{O}(\Lambda)$.

Convex Constraints: The Algorithm

1. Build local interpolation model $m_k(\boldsymbol{s})$
2. Minimize the model within the trust-region Δ_k to get the step

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

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

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

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

Convex Constraints: Convergence & Complexity

A new question arises...

How can we measure stationarity? i.e. how do we measure progress?

Convex Constraints: Convergence & Complexity

A new question arises...

How can we measure stationarity? i.e. how do we measure progress?

$$\pi^f(\boldsymbol{x}) := \left| \min_{\substack{x+s\in\mathcal{C} \\ ||s_2||\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{s} \right|$$

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 $\boldsymbol{x}$, assuming ∇f is Lipschitz continuous [Cartis, Gould & Toint, 2012]

Convex Constraints: Convergence & Complexity

A new question arises...

How can we measure stationarity? i.e. how do we measure progress?

$$\pi^f(\boldsymbol{x}) := \left| \min_{\substack{x+s\in\mathcal{C} \\ ||s_2||\leq 1}} \nabla f(\boldsymbol{x})^T \boldsymbol{s} \right|$$

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 $\boldsymbol{x}$, assuming ∇f is Lipschitz continuous [Cartis, Gould & Toint, 2012]
- If m_k is fully linear, then $|\pi^f(\boldsymbol{x}_k) - \pi^{m_k}(\boldsymbol{x}_k)| \leq \kappa_{eg}\Delta_k$ [Hough & Roberts, 2021]

Convex Constraints: Convergence & Complexity

Convergence and worst-case complexity for nonconvex functions match what we have for the unconstrained case:

Theorem (convergence) (Hough & Roberts, 2021)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi^f(\boldsymbol{x}_k) = 0$.

Convex Constraints: Convergence & Complexity

Convergence and worst-case complexity for nonconvex functions match what we have for the unconstrained case:

Theorem (convergence) (Hough & Roberts, 2021)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi^f(\boldsymbol{x}_k) = 0$.

Theorem (complexity) (Hough & Roberts, 2021)

Under the same assumptions as above, 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

Convergence and worst-case complexity for nonconvex functions match what we have for the unconstrained case:

Theorem (convergence) (Hough & Roberts, 2021)

If f has Lipschitz continuous gradient and is bounded below, then we have $\lim_{k\to\infty} \pi^f(\boldsymbol{x}_k) = 0$.

Theorem (complexity) (Hough & Roberts, 2021)

Under the same assumptions as above, we achieve $\pi^f(\boldsymbol{x}_k) \leq \epsilon$ for the first time after at most $\mathcal{O}(\epsilon^{-2})$ iterations.

Requires two important algorithms:

- Check a model is fully linear
- Change the interpolation set to make the model fully linear if it is not

Outline

1. Introduction to DFO trust-region methods
2. Extending to convex constraints
3. **Application to least-squares problems**

The Least-Squares Case

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

- Typically (Gauss-Newton) linearize $\boldsymbol{r}$ at $\boldsymbol{x}_k$ using the Jacobian:

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

- But in DFO, Jacobian is not available:

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

- Find J_k by interpolation [Cartis & Roberts, 2019]

The Least-Squares Case

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

- Typically (Gauss-Newton) linearize $\boldsymbol{r}$ at $\boldsymbol{x}_k$ using the Jacobian:

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

- But in DFO, Jacobian is not available:

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

- Find J_k by interpolation [Cartis & Roberts, 2019]

Either way, end up with a local quadratic model

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

Least-Squares Implementation

Update the state-of-the-art solver **DFO-LS**: [Cartis et al., 2019]

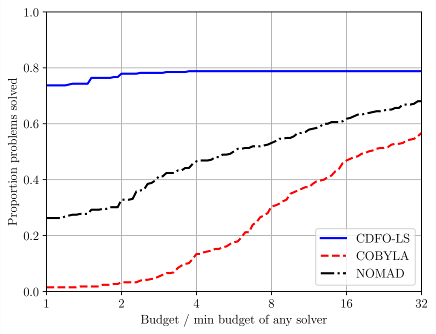
- Use FISTA to solve the constrained trust-region subproblem
- Requires Dykstra's algorithm to project onto $B(\boldsymbol{x}_k, \Delta_k) \cap \mathcal{C}$
- Github: `numericalalgorithmsgroup/dfols`

Tested on a collection of 58 low-dimensional least-squares problems with box/ball/halfspace/second-order cone constraints.

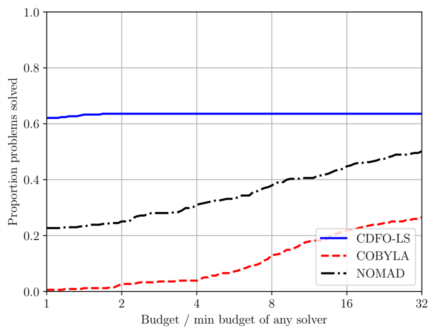
Limited solvers to compare to (none exploit the least-squares structure):

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

Numerical Results



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



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

Measuring the proportion of problems solved vs. the number of objective evaluations (higher is better).

Summary

- General model-based DFO method for convex-constrained problems
- Match/generalize existing convergence and complexity results
- Develop new theory of Λ-poisedness and full linearity[1]
- New software for least-squares problems

Future Work

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

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

[1] Only for linear/composite models at present