Black-Box Optimization Algorithms for Problems with Convex Regularizers

joint work with Lindon Roberts (University of Sydney)

Yanjun Liu, Australian National University / Princeton University
yanjun.liu@princeton.edu

ISMP, 26 July 2024

Problem Setup

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x})$$

1. $f(\boldsymbol{x}) := \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x})\right\|^2 = \frac{1}{2}\sum_{i=1}^m r_i(\boldsymbol{x})^2$,
where $\boldsymbol{r}(\boldsymbol{x}) := \left[r_1(\boldsymbol{x}) \dots r_m(\boldsymbol{x})\right]^T$ mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$.
Assume $\boldsymbol{r}(\boldsymbol{x}) \in C^1$ with Jacobian $[J(\boldsymbol{x})]_{i,j} = \frac{\partial r_i(\boldsymbol{x})}{\partial x_j}$.
However, these derivatives might not be accessible!

2. $h : \mathbb{R}^n \to \mathbb{R}$ is a convex but possibly nonsmooth regularization term.
Assume the proximal operator of h is cheap to evaluate.

Motivation

Learning MRI Sampling Patterns (Ehrhardt and Roberts 2021):

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{x}^*(\boldsymbol{\theta}) - \boldsymbol{x}_{true}\|^2 + \|\boldsymbol{\theta}\|_1$$

- $\boldsymbol{\theta} \in \mathbb{R}^d$: weights determining importance of Fourier coefficients of the image
- $\boldsymbol{x}^*(\boldsymbol{\theta})$: reconstruct process is complicated!
- 1-norm: keep sparsity to save time for MRI scan.

Derivative-free optimization (DFO):

- black-box, noisy or expensive to evaluate.
- several approaches: direct search, Nelder-Mead, model-based, $\cdots$

Classical Trust Region Framework

$$\min_{\boldsymbol{x}} \Phi(\boldsymbol{x}), \quad \Phi \text{ is smooth}$$

At k-th iteration:

1. Construct a model function m_k approximating Φ within trust region $B(\boldsymbol{x}_k, \Delta_k)$
2. Find a minimizer of m_k within the trust region
$$\boldsymbol{s}_k \in \arg\min_{\|\boldsymbol{s}\| \leq \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{s})$$
3. Calculate the ratio
$$R_k = \frac{\texttt{objective decrease}}{\texttt{model decrease}} = \frac{\Phi(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}$$
4. Update iterate $\boldsymbol{x}_{k+1}$, trust region radius Δ_{k+1} based on R_k.
(R_k close to 1: step $\boldsymbol{s}_k$ successful)

Convergence:
Under reasonable assumption, stationary measure $\|\nabla \Phi(\boldsymbol{x}_k)\| \to 0$.

Classical Trust Region Framework

$$\min_{\boldsymbol{x}} \Phi(\boldsymbol{x}), \quad \Phi \text{ is smooth}$$

At k-th iteration:

1. Construct a model function m_k approximating Φ within trust region $B(\boldsymbol{x}_k, \Delta_k)$
2. Find a minimizer of m_k within the trust region
$$\boldsymbol{s}_k \in \arg\min_{\|\boldsymbol{s}\| \le \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{s})$$
3. Calculate the ratio
$$R_k = \frac{\texttt{objective decrease}}{\texttt{model decrease}} = \frac{\Phi(\boldsymbol{x}_k) - \Phi(\boldsymbol{x}_k + \boldsymbol{s}_k)}{m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k)}$$
4. Update iterate $\boldsymbol{x}_{k+1}$, trust region radius Δ_{k+1} based on R_k. (R_k close to 1: step $\boldsymbol{s}_k$ successful)

Convergence:

Under reasonable assumption, stationary measure $\|\nabla\Phi(\boldsymbol{x}_k)\| \to 0$.

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x})\right\|^2$$

1. Model construction?
2. Finding minimizer of model function?
3. Update rule?
4. Stationary measure?

Model Construction

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|^2$$

[Cartis and Roberts 2019]

1. Derivative-based (Jacobian $J(\boldsymbol{x})$ available): Taylor expand $\boldsymbol{r}(\boldsymbol{x})$:

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

$$\Phi(\boldsymbol{x}_k + \boldsymbol{s}) \approx f(\boldsymbol{x}_k) + \boldsymbol{r}(\boldsymbol{x}_k)^T J(\boldsymbol{x}_k)\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T J(\boldsymbol{x}_k)^T J(\boldsymbol{x}_k)\boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s})$$

2. Derivative-free: Approximate Jacobian $J(\boldsymbol{x}_k)$ at iterate $\boldsymbol{x}_k$ by J_k:

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

$$\Phi(\boldsymbol{x}_k + \boldsymbol{s}) \approx m_k(\boldsymbol{x}_k + \boldsymbol{s}) := \underbrace{f(\boldsymbol{x}_k) + \boldsymbol{g}_k^T\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T H_k\boldsymbol{s}}_{p_k(\boldsymbol{x}_k+\boldsymbol{s})} + h(\boldsymbol{x}_k + \boldsymbol{s})$$

where $\boldsymbol{g}_k := J_k^T\boldsymbol{r}(\boldsymbol{x}_k)$ and $H_k := J_k^T J_k$ (**symmetric + p.s.d.**).

Calculation of $\boldsymbol{g}_k$ and H_k: For each iteration, maintain an interpolation set $Y_k := \{\boldsymbol{y}_0 := \boldsymbol{x}_k, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$. Interpolation condition:

$$\boldsymbol{m}_k(\boldsymbol{y}_t) = \boldsymbol{r}(\boldsymbol{y}_t), \forall t = 0, 1, \cdots, n$$

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x})\right\|^2$$

1. Model construction? Include h in m_k
2. Finding minimizer of model function?
3. Update rule? Interpolation
4. Stationary measure?

Existing work

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x}_k)\right\|^2$$

Derivative-free:

- Deal with nonsmooth Ψ without exploiting structure:
  1. Model-based: [Audet and Hare 2020]
  2. Direct search: [Audet and Dennis 2006]
- Deal with $f(\boldsymbol{x}) + h(\boldsymbol{c}(\boldsymbol{x}))$
  for f and c black-box smooth, h convex nonsmooth:
  1. [Grapiglia, J. Yuan, and Y.-x. Yuan 2016]:
     - DFO version of [Cartis, Gould, and Toint 2011]
  2. [Garmanjani, Júdice, and Vicente 2016]:
     - convergence, worse-case complexity
     - smooth vs composite approach
  3. [Larson and Menickelly 2024]:
     - model-based trust region

Stationary Measure

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x}_k)\|^2$$

1. If the objective function Φ is smooth: $\|\nabla\Phi(\boldsymbol{x}^*)\| = 0$.
2. If ∇f accessible:

$$l(\boldsymbol{x},\boldsymbol{s}) := f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T\boldsymbol{s} + h(\boldsymbol{x}+\boldsymbol{s})$$
$$\zeta(\boldsymbol{x}) := l(\boldsymbol{x},0) - \min_{\|\boldsymbol{s}\|\le 1} l(\boldsymbol{x},\boldsymbol{s})$$

   We say that $\boldsymbol{x}^*$ is a *critical point* of Φ if $\zeta(\boldsymbol{x}^*) = 0$.
   [Cartis, Gould, and Toint 2011]
3. If ∇f inaccessible: At k-th iteration, after calculating a local approximation p_k of f:

$$\tilde{l}(\boldsymbol{x},\boldsymbol{s}) := f(\boldsymbol{x}) + \nabla p_k(\boldsymbol{x})^T\boldsymbol{s} + h(\boldsymbol{x}+\boldsymbol{s}), \boldsymbol{s}\in\mathbb{R}^n$$
$$\eta(\boldsymbol{x}) := \tilde{l}(\boldsymbol{x},0) - \min_{\|\boldsymbol{s}\|\le 1} \tilde{l}(\boldsymbol{x},\boldsymbol{s}).$$

   Note: If $h \equiv 0$, $\eta(\boldsymbol{x}_k) = \|\boldsymbol{g}_k\|$. [Grapiglia, J. Yuan, and Y.-x. Yuan 2016]

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x})\right\|^2$$

1. Model construction? Include h in m_k
2. Finding minimizer of model function?
3. Update rule? Interpolation
4. Stationary measure? Introduce $\eta(\boldsymbol{x}_k)$ (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{r}(\boldsymbol{x})\right\|^2$$

1. Model construction? Include h in m_k
2. Finding minimizer of model function?
3. Update rule? Interpolation
4. Stationary measure? Introduce $\eta(\boldsymbol{x}_k)$ (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)
   New Problem:
5. For convergence, we need the criticality phase to ensure that Δ_k is comparable to $\eta(\boldsymbol{x}_k)$.
   How to compute the stationary measure $\eta(\boldsymbol{x}_k)$?

Implementation: Inexact Solver for Two subproblems

1. Calculating approximate stationary measure:

$$\begin{aligned}
\tilde{l}(\boldsymbol{x}_k, \boldsymbol{s}) &:= f(\boldsymbol{x}_k) + \nabla p_k(\boldsymbol{x}_k)^T \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s}) \\
&= f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s}) \\
\eta_1(\boldsymbol{x}_k) &:= \tilde{l}(\boldsymbol{x}_k, 0) - \min_{\|\boldsymbol{s}\| \le 1} \tilde{l}(\boldsymbol{x}_k, \boldsymbol{s})
\end{aligned}$$

2. Calculating step size $\boldsymbol{s}_k$:

$$\boldsymbol{s}_k \in \arg \min_{\|\boldsymbol{s}\| \le \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}^T H_k \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s})$$

Implementation: Inexact Solver for Two subproblems

1. Calculating approximate stationary measure:

$$\begin{aligned} \tilde{l}(\boldsymbol{x}_k, \boldsymbol{s}) &:= f(\boldsymbol{x}_k) + \nabla p_k(\boldsymbol{x}_k)^T \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s}) \\ &= f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s}) \\ \eta_1(\boldsymbol{x}_k) &:= \tilde{l}(\boldsymbol{x}_k, 0) - \min_{\|\boldsymbol{s}\| \le 1} \tilde{l}(\boldsymbol{x}_k, \boldsymbol{s}) \end{aligned}$$

2. Calculating step size $\boldsymbol{s}_k$:

$$\boldsymbol{s}_k \in \arg\min_{\|\boldsymbol{s}\| \le \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{s}) = f(\boldsymbol{x}_k) + \boldsymbol{g}_k{}^T \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}^T H_k \boldsymbol{s} + h(\boldsymbol{x}_k + \boldsymbol{s})$$

$\Rightarrow$Both are of the form: convex smooth + convex nonsmooth s.t. ball constraint. Specifically, given $\boldsymbol{g}$, $H, h, \boldsymbol{x}, r$, and $C := B(0, r)$,

$$\min_{\boldsymbol{d}} G(\boldsymbol{d}) := \underbrace{\boldsymbol{g}^T \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^T H \boldsymbol{d}}_{\text{smooth}} + \underbrace{h(\boldsymbol{x} + \boldsymbol{d})}_{\text{nonsmooth}} + \underbrace{I_C(\boldsymbol{d})}_{\text{nonsmooth}} .$$

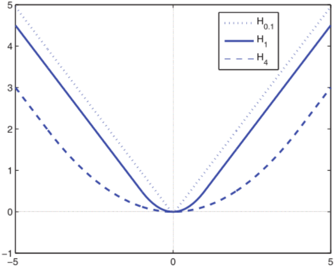
Implementation: Inexact Solver for Two subproblems

At k-th iteration, given $\boldsymbol{g}$, $H, h, \boldsymbol{x}, r$, and $C := B(0, r)$ and

$$\min_{\boldsymbol{d}} G(\boldsymbol{d}) := \underbrace{\boldsymbol{g}^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T H \boldsymbol{d}}_{\text{smooth}} + \underbrace{h(\boldsymbol{x} + \boldsymbol{d})}_{\text{nonsmooth}} + \underbrace{I_C(\boldsymbol{d})}_{\text{nonsmooth}} .$$

IDEA: Replacing the nonsmooth h by its smooth approximation. Given a smoothing parameter $\mu > 0$, smoothing h by its *Moreau envelope*:

$$M_h^{\mu}(\boldsymbol{y}) := \min_{\boldsymbol{z}} \left\{ h(\boldsymbol{z}) + \frac{1}{2\mu} \|\boldsymbol{y} - \boldsymbol{z}\|^2 \right\}$$



Implementation: Inexact Solver for Two Subproblems

Smoothed version:

$$\Rightarrow \min_{\boldsymbol{d}} G_\mu(\boldsymbol{d}) := \underbrace{\boldsymbol{g}^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T H \boldsymbol{d} + M_h^\mu(\boldsymbol{x}+\boldsymbol{d})}_{\text{smooth convex}=:F_\mu(\boldsymbol{d})} + \underbrace{I_C(\boldsymbol{d})}_{\text{nonsmooth convex}}.$$

Now applying accelerated proximal gradient method (FISTA):

- $\nabla F_\mu(\boldsymbol{d}) = \boldsymbol{g} + H\boldsymbol{d} + \nabla M_h^\mu(\boldsymbol{x}+\boldsymbol{d})$
- proximal operator of I_C is the projection operator P_C onto C.

Algorithm (Solving two subproblems: Smooth-FISTA (Beck 2017))

Given smoothing parameter $\mu > 0$.

1. Set $\boldsymbol{d}^0 = \boldsymbol{y}^0 = 0$, $t_0 = 1$, and step size $L = \|H\| + \frac{1}{\mu}$.
2. For $j = 0, 1, 2, \ldots$
3. set $\boldsymbol{d}^{j+1} = P_C\left(\boldsymbol{y}^j - \frac{1}{L}\nabla F_\mu(\boldsymbol{y}^j)\right)$;
4. compute $\boldsymbol{y}^{j+1} = \boldsymbol{d}^{j+1} + \left(\frac{t_j - 1}{t_{j+1}}\right)(\boldsymbol{d}^{j+1} - \boldsymbol{d}^j)$.

Implementation: Inexact Solver for Two Subproblems

At k-th iteration, given $\boldsymbol{g}$, $H, h, \boldsymbol{x}, r$, and $C := B(0, r)$,

$$\min_{\boldsymbol{d}} G(\boldsymbol{d}) := \underbrace{\boldsymbol{g}^T\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T H\boldsymbol{d}}_{\text{smooth}} + \underbrace{h(\boldsymbol{x}+\boldsymbol{d})}_{\text{nonsmooth}} + \underbrace{I_C(\boldsymbol{d})}_{\text{nonsmooth}} .$$

Theorem (S-FISTA, (Beck 2017))

Suppose that h is convex and L_h-Lipschitz continuous. Let $\{\boldsymbol{d}^j\}_{j\geq 0}$ be the sequence generated by S-FISTA. For an accuracy level $\epsilon > 0$, if the smoothing parameter μ and the number of iterations J are set as

$$\mu = \frac{2\epsilon}{L_h(L_h + \sqrt{L_h^2 + 2\|H\|\epsilon})} \quad \text{and} \quad J = \frac{r(2L_h + \sqrt{2\|H\|\epsilon})}{\epsilon}, \tag{1}$$

then for any $j \geq J$, it holds that $G(\boldsymbol{d}^j) - G(\boldsymbol{d}^*) \leq \epsilon$.

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|^2$$

1. Model construction? Include h in m_k
2. Finding minimizer of model function? Using S-FISTA!
3. Update rule? Interpolation
4. Stationary measure? Introduce $\eta(\boldsymbol{x}_k)$ (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)

   New Problem:
5. For convergence, we need the criticality phase to ensure that Δ_k is comparable to $\eta(\boldsymbol{x}_k)$.

   How to compute the stationary measure $\eta(\boldsymbol{x}_k)$? Using S-FISTA!

Questions

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x})\|^2$$

1. Model construction? Include h in m_k
2. Finding minimizer of model function? Using S-FISTA!
3. Update rule? Interpolation
4. Stationary measure? Introduce $\eta(\boldsymbol{x}_k)$ (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)
   New Problem:
5. For convergence, we need the criticality phase to ensure that Δ_k is comparable to $\eta(\boldsymbol{x}_k)$.
   How to compute the stationary measure $\eta(\boldsymbol{x}_k)$? Using S-FISTA!

But S-FISTA is inexact! New issues:

- To get our algorithm work, What is the accuracy we need the stationary measure computed to?
- What is the sufficient decrease condition for computing trust region steps?

$\Rightarrow$How to pick ϵ in both cases?

Implementation: Choosing Accuracy Level

Theoretically, we discussed:

- Model construction: include h in m_k
- Stationary measure: introduce η (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)

Practically, how to implement the algorithm?

- How to find a minimizer of m_k within the trust region?
- For convergence, we have the criticality phase to ensure Δ_k is comparable to $\eta(\boldsymbol{x}_k)$.
  How to compute the stationary measure $\eta(\boldsymbol{x}_k)$?

Solution: Using S-FISTA!

Implementation: Choosing Accuracy Level

Theoretically, we discussed:

- Model construction: include h in m_k
- Stationary measure: introduce η (if $h \equiv 0$, equal to the $\|\boldsymbol{g}_k\|$)

Practically, how to implement the algorithm?

- How to find a minimizer of m_k within the trust region?
- For convergence, we have the criticality phase to ensure Δ_k is comparable to $\eta(\boldsymbol{x}_k)$.
  How to compute the stationary measure $\eta(\boldsymbol{x}_k)$?

Solution: Using S-FISTA!

But S-FISTA is inexact! New issues:

- To get our algorithm work, What is the accuracy we need the stationary measure computed to?
- What is the sufficient decrease condition for computing the step size?

$\Rightarrow$How to pick ϵ in both cases?

Implementation: Choosing Accuracy Level

Inaccurate estimation:

1. Stationary measure $\eta(\boldsymbol{x}_k) := \tilde{l}(\boldsymbol{x}_k, 0) - \min_{\|\boldsymbol{s}\|\leq 1} \tilde{l}(\boldsymbol{x}_k, \boldsymbol{s})$: Applying S-FISTA until

$$\eta(\boldsymbol{x}_k) - \overline{\eta}(\boldsymbol{x}_k) \leq \epsilon_1 \Delta_k$$

2. Step size $\boldsymbol{s}_k \in \arg\min_{\|\boldsymbol{s}\|\leq \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{s})$: Applying S-FISTA until

$$m_k(\boldsymbol{x}_k+\boldsymbol{s}_k) - \arg\min_{\|\boldsymbol{s}\|\leq\Delta_k} m_k(\boldsymbol{x}_k+\boldsymbol{s}) \leq (1-\epsilon_2)\overline{\eta}(\boldsymbol{x}_k)\min\left\{\Delta_k, \frac{\overline{\eta}(\boldsymbol{x}_k)}{\max\{1, \|H_k\|\}}\right\}.$$

Remark: This implies a generalized *Cauchy decrease condition*:

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \epsilon_2 \overline{\eta}(\boldsymbol{x}_k)\min\left\{\Delta_k, \frac{\overline{\eta}(\boldsymbol{x}_k)}{\max\{1, \|H_k\|\}}\right\}.$$

Full Algorithm

Calculate approximate stationary measure $\eta(\boldsymbol{x}_k)$ *inaccurately*: applying using S-FISTA until

$$\eta(\boldsymbol{x}_k) - \overline{\eta}(\boldsymbol{x}_k) \leq \epsilon_1 \Delta_k$$

1. If $\overline{\eta}(\boldsymbol{x}_k) < \epsilon$, go to the criticality phase.
2. Construct a model function m_k within trust region $B(\boldsymbol{x}_k, \Delta_k)$:

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

3. Find a minimizer of m_k within the trust region: using *inexact* solver S-FISTA to get a step $\boldsymbol{s}_k$ satisfying $\|\boldsymbol{s}_k\| \leq \Delta_k$, $m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \leq m_k(\boldsymbol{x}_k)$ and

$$m_k(\boldsymbol{x}_k) - m_k(\boldsymbol{x}_k + \boldsymbol{s}_k) \geq \epsilon_2 \overline{\eta}(\boldsymbol{x}_k) \min \left\{ \Delta_k, \frac{\overline{\eta}(\boldsymbol{x}_k)}{\max\{1, \|H_k\|\}} \right\}.$$

4. Calculate the decrease ratio R_k
5. Update iterate $\boldsymbol{x}_{k+1}$, trust region radius Δ_{k+1} based on R_k and *interpolation set*.
   (R_k close to 1 & good geometry of *interpolation set*: step $\boldsymbol{s}_k$ successful)

Convergence & Complexity

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \Phi(\boldsymbol{x}) = f(\boldsymbol{x}) + h(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{r}(\boldsymbol{x}_k)\|^2$$

Convergence and worst-case complexity match for the case $h = 0$.
Assumptions:

- f is continuously differentiable; ∇f is Lipschitz continuous.
- h is convex (possibly nonsmooth) and Lipschitz continuous.
- (standard) the model Hessians $\|H_k\|$ are uniformly bounded.

Theorem (Convergence - *true* stationary measure)

$$\lim_{k\to\infty} \zeta(\boldsymbol{x}_k) = 0.$$

Theorem (Complexity)

For $\epsilon \in (0, 1]$, the number of iterations until $\Psi_1(\boldsymbol{x}_k) < \epsilon$ for the first time is at most $k = \mathcal{O}(\epsilon^{-2})$, same as the unregularized DFO.

Numerical Experiments

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

- Use S-FISTA to calculate the generalized stationary measure and trust region subproblem with regularization *inaccurately*.
- Extend the safety phase from DFO-LS to the case with regularization: detect insufficient decrease generated by the step size $\|\boldsymbol{s}_k\|$ before evaluating $f(x_k)$.
- Require the proximal operator of h easy-to-compute

Tested on a collection of 53 low-dimensional, unconstrained nonlinear least squares (from [Moré and Wild 2009]) with 1-norm regularization.

Numerical Experiments

We compare DFO-LSR to:
NOMAD - direct search DFO solver (not exploit the least-squares structure). [Le Digabel 2011]
Measuring the proportion of problems solved vs. the number of evaluations

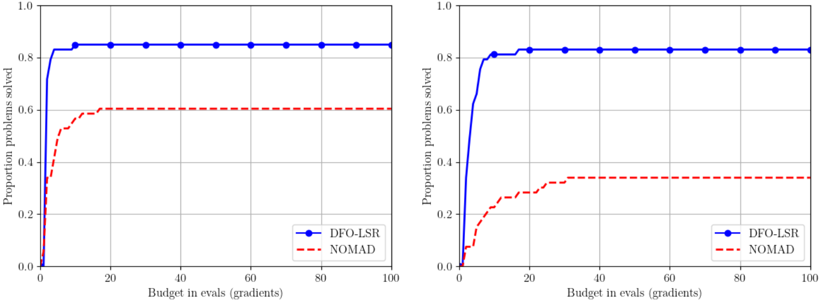


Figure: Left: accuracy level $\tau = 10^{-1}$; Right: accuracy level $\tau = 10^{-3}$

Summary and Future Work

Summary:

- Generalize model-based DFO method for minimizing nonconvex smooth function with convex regularizers
- Applying S-FISTA to compute stationary measure and step size inaccurately, with practical implementation and theoretical analysis (results matching with unregularized DFO)
- New software for least-squares problems with convex regularizers

Future work:

- Adapt to model functions as the sum of derivative-free but *possibly nonconvex quadratic* approximation and convex regularizer.

: https://github.com/yanjunliu-regina/dfols

: https://arxiv.org/abs/2407.14915

: https://yanjunliu-regina.github.io/talk/yanjun_liu_ismp_2024.pdf

References I

- Audet, Charles and J. E. Dennis (2006). "Mesh Adaptive Direct Search Algorithms for Constrained Optimization". In: *SIAM Journal on Optimization* 17.1, pp. 188–217.
- Audet, Charles and Warren Hare (2020). "Model-Based Methods in Derivative-Free Nonsmooth Optimization". In: *Numerical Nonsmooth Optimization: State of the Art Algorithms*. Springer International Publishing, pp. 655–691.
- Beck, Amir (2017). *First-order methods in optimization*. SIAM.
- Cartis, Coralia, Jan Fiala, et al. (2019). "Improving the flexibility and robustness of model-based derivative-free optimization solvers". In: *ACM Transactions on Mathematical Software (TOMS)* 45.3, pp. 1–41.
- Cartis, Coralia, Nicholas IM Gould, and Philippe L Toint (2011). "On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming". In: *SIAM Journal on Optimization* 21.4, pp. 1721–1739.

References II

Cartis, Coralia and Lindon Roberts (2019). “A derivative-free Gauss–Newton method”. In: *Mathematical Programming Computation* 11, pp. 631–674.

Ehrhardt, Matthias J and Lindon Roberts (2021). “Inexact derivative-free optimization for bilevel learning”. In: *Journal of mathematical imaging and vision* 63.5, pp. 580–600.

Garmanjani, Rohollah, Diogo Júdice, and Luís Nunes Vicente (2016). “Trust-region methods without using derivatives: worst case complexity and the nonsmooth case”. In: *SIAM Journal on Optimization* 26.4, pp. 1987–2011.

Grapiglia, Geovani Nunes, Jinyun Yuan, and Ya-xiang Yuan (2016). “A derivative-free trust-region algorithm for composite nonsmooth optimization”. In: *Computational and Applied Mathematics* 35.2, pp. 475–499.

References III

Larson, Jeffrey and Matt Menickelly (2024). "Structure-aware methods for expensive derivative-free nonsmooth composite optimization". In: *Mathematical Programming Computation* 16.1, pp. 1–36.

Le Digabel, Sébastien (2011). "Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm". In: *ACM Transactions on Mathematical Software (TOMS)* 37.4, pp. 1–15.

Moré, Jorge J and Stefan M Wild (2009). "Benchmarking derivative-free optimization algorithms". In: *SIAM Journal on Optimization* 20.1, pp. 172–191.