Trust-Region Methods for the Derivative-Free Optimization of Nonsmooth Black-Box Functions[1]

Luis Nunes Vicente
Lehigh University

ICOTA 2019, Xiangtan University

Dedicated to the 60th Anniversary of Professor Ya-xiang Yuan

[1] joint work with G.Liuzzi, S.Lucidi, F.Rinaldi

Presentation outline

Presentation outline

Problem definition

Problem to be solved:

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

Problem definition

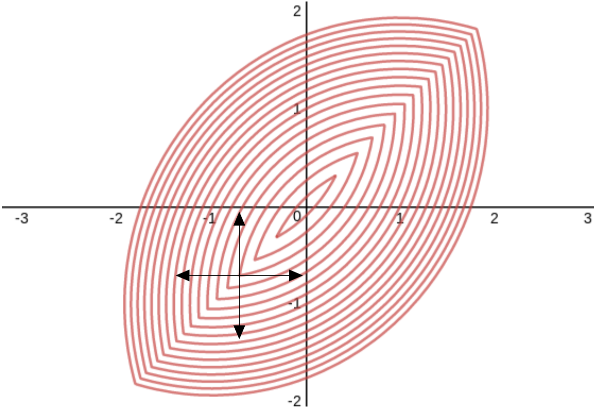
Problem to be solved:

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

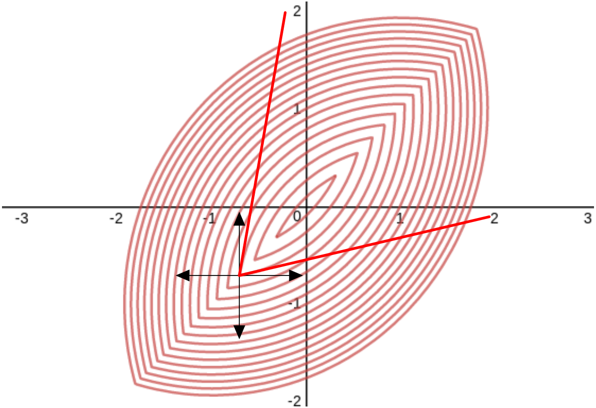
Assumptions:

- $f : \mathbb{R}^n \to \mathbb{R}$ is nonsmooth but Lipschitz continuous
- Mathematical representation of the objective function not available: **No knowledge about source of nonsmoothness**
- Function evaluations costly

Coordinate search failure



Coordinate search failure

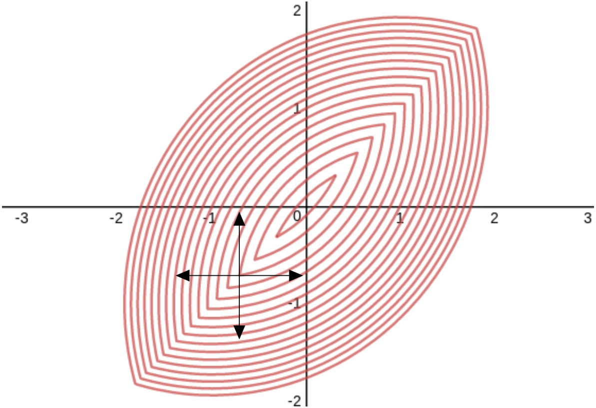


Drawback

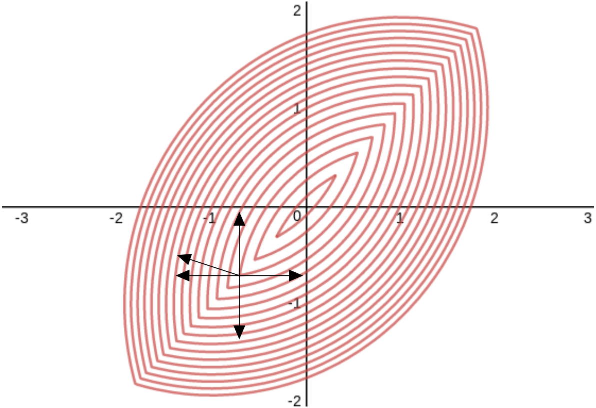
Method gets stuck ... no search direction is in the cone of descent directions!

Convergence to non-stationary point

Asymptotically dense sets of search directions



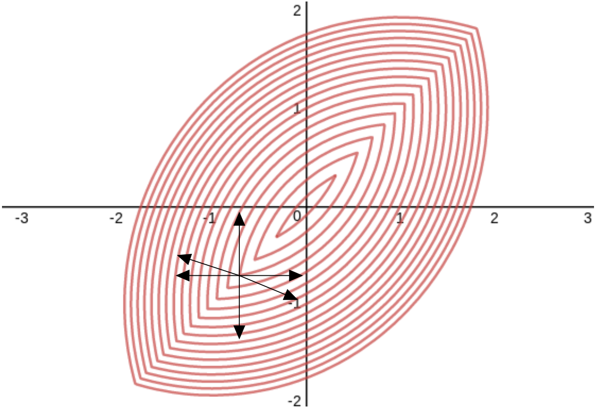
Asymptotically dense sets of search directions



Trick

Enrich the set with new search directions

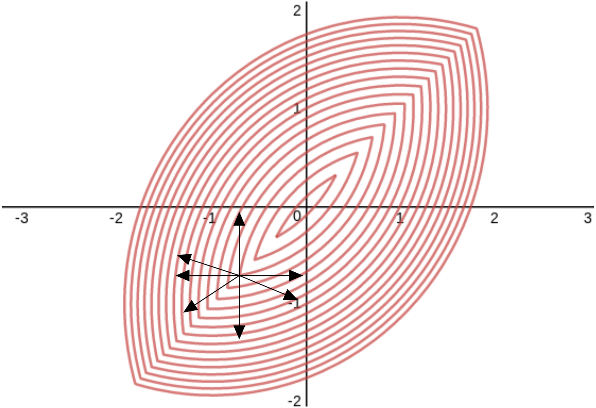
Asymptotically dense sets of search directions



Trick

Enrich the set with new search directions

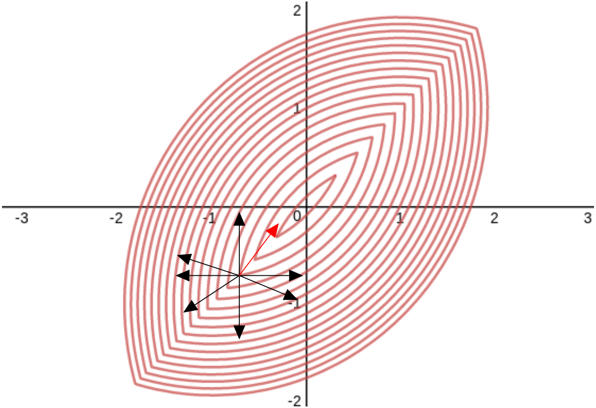
Asymptotically dense sets of search directions



Trick

Enrich the set with new search directions

Asymptotically dense sets of search directions



Trick

Enrich the set with new search directions

How to handle black-box nonsmoothness?

- Directional approaches based on random directions asymptotically dense on the unit sphere:

  [Audet, Dennis, 2006]
  [LNV, Custódio, 2012]
  [Fasano, Liuzzi, Lucidi, Rinaldi, 2014]

How to handle black-box nonsmoothness?

- Directional approaches based on random directions asymptotically dense on the unit sphere:

  [Audet, Dennis, 2006]
  [LNV, Custódio, 2012]
  [Fasano, Liuzzi, Lucidi, Rinaldi, 2014]

- Approaches based on convex hull of (possibly randomly) sampled approximate gradients:

  [Bagirov, Karasozen, Sezer, 2006]
  [Kiwiel, 2010]
  [Hare, Nutini, 2013]

How to handle black-box nonsmoothness?

- Directional approaches based on random directions asymptotically dense on the unit sphere:

  [Audet, Dennis, 2006]
  [LNV, Custódio, 2012]
  [Fasano, Liuzzi, Lucidi, Rinaldi, 2014]

- Approaches based on convex hull of (possibly randomly) sampled approximate gradients:

  [Bagirov, Karasozen, Sezer, 2006]
  [Kiwiel, 2010]
  [Hare, Nutini, 2013]

- Other convergent approaches require knowledge of nonsmoothness

Basic tools in non-smooth context

Clarke Directional Derivative

Given a point $x \in \mathbb{R}^n$ and a direction $d \in \mathbb{R}^n$, the Clarke directional derivative of f at x along d is defined as

$$f^\circ(x; d) = \limsup_{y \to x, t \downarrow 0} \frac{f(y + td) - f(y)}{t}$$

Basic tools in non-smooth context

Clarke Directional Derivative

Given a point $x \in \mathbb{R}^n$ and a direction $d \in \mathbb{R}^n$, the Clarke directional derivative of f at x along d is defined as

$$f^\circ(x; d) = \limsup_{y \to x, t \downarrow 0} \frac{f(y + td) - f(y)}{t}$$

Clarke Stationary Point

A point $x^* \in \mathbb{R}^n$ is Clarke stationary when $f^\circ(x^*, d) \geq 0$, for all $d \in \mathbb{R}^n$.

Algorithm Basic DFO-TRNS

Initialization. Select $x_0 \in \mathbb{R}^n$, $\eta > 0$, $0 < \gamma_1 < 1 \leq \gamma_2$, $\Delta_0 > 0$, and $p > 0$.

For $k = 0, 1 \ldots$

Generate randomly g_k in the unit sphere. Build a symmetric matrix B_k.

Let

$$\begin{aligned} s_k &\in \underset{\|s\|^2 \leq \Delta_k^2}{\operatorname{argmin}}\ m_k(s) = f(x_k) + {g_k}^\top s + \frac{1}{2} s^\top B_k s \\ \rho_k &= \frac{f(x_k) - f(x_k + s_k)}{\|s_k\|^{1+p}} \end{aligned}$$

If $\rho_k \geq \eta$, **Then** (**Success**), $x_{k+1} \leftarrow x_k + s_k$, $\Delta_{k+1} \leftarrow \gamma_2 \Delta_k$.

Else (**Failure**), $x_{k+1} \leftarrow x_k$, $\Delta_{k+1} \leftarrow \gamma_1 \Delta_k$.

End for

Why will Clarke be nonnegative along limiting TR steps

Mechanism

Predicted reduction

$$o(\|s_k\|) \;=\; \theta\|s\|^{1+p}$$

used in place of reduction in quadratic model $m_k(\mathbf{0}) - m_k(s_k)$.

Why will Clarke be nonnegative along limiting TR steps

Mechanism

Predicted reduction

$$o(\|s_k\|) \;=\; \theta\|s\|^{1+p}$$

used in place of reduction in quadratic model $m_k(\mathbf{0}) - m_k(s_k)$.

Reason

In unsuccessful iterations iterations, one has ($\rho_k < \eta$)

$$f(x_k + s_k) - f(x_k) \;>\; -\eta\|s_k\|^{1+p}$$

Why will Clarke be nonnegative along limiting TR steps

Mechanism

Predicted reduction

$$o(\|s_k\|) \;=\; \theta\|s\|^{1+p}$$

used in place of reduction in quadratic model $m_k(\mathbf{0}) - m_k(s_k)$.

Reason

In unsuccessful iterations iterations, one has $(\rho_k < \eta)$

$$f(x_k + s_k) - f(x_k) \;>\; -\eta\|s_k\|^{1+p}$$

or

$$\frac{f(x_k + \|s_k\|(s_k/\|s_k\|)) - f(x_k)}{\|s_k\|} \;>\; -\theta\|s_k\|^p$$

Why will Clarke be nonnegative along limiting TR steps

Mechanism

Predicted reduction

$$o(\|s_k\|) \;=\; \theta\|s\|^{1+p}$$

used in place of reduction in quadratic model $m_k(\mathbf{0}) - m_k(s_k)$.

Reason

In unsuccessful iterations iterations, one has $(\rho_k < \eta)$

$$f(x_k + s_k) - f(x_k) \;>\; -\eta\|s_k\|^{1+p}$$

or

$$\frac{f(x_k + \|s_k\|(s_k/\|s_k\|)) - f(x_k)}{\|s_k\|} \;>\; -\theta\|s_k\|^p$$

Taking limits will when $\|s_k\| \longrightarrow 0$, will yield the **Clark derivative non-negative along limiting TR steps**

Asymptotic behavior of trust radius

Because in successful iterations we have

$$f(x_k) - f(x_k + s_k) \;>\; \eta \|s_k\|^{1+p} \;>\; \cdots$$

Asymptotic behavior of trust radius

Because in successful iterations we have

$$f(x_k) - f(x_k + s_k) \ > \ \eta \|s_k\|^{1+p} \ > \ \cdots$$

Trust-region radius converges to zero

Assume that f is bounded from below. Any sequence $\{\Delta_k\}$ of trust-region radii produced by Algorithm **Basic DFO-TRNS** is such that

$$\lim_{k \to \infty} \Delta_k \ = \ 0$$

Asymptotic behavior of trust radius

Because in successful iterations we have

$$f(x_k) - f(x_k + s_k) \; > \; \eta \|s_k\|^{1+p} \; > \; \cdots$$

Trust-region radius converges to zero

Assume that f is bounded from below. Any sequence $\{\Delta_k\}$ of trust-region radii produced by Algorithm **Basic DFO-TRNS** is such that

$$\lim_{k \to \infty} \Delta_k \; = \; 0$$

The result is true even if B_k is unbounded as long as bounded by a power of Δ_k^{-q}, $q \in (0, 1)$.

Why will the TR steps also cover the unit sphere

Property

Any sequence $\{(x_k, s_k, \Delta_k)\}$ *generated by Algorithm* **Basic DFO-TRNS** *is such that*

$$s_k \;=\; -\Delta_k D_k g_k$$

with $D_k \in \mathbb{R}^{n\times n}$ *satisfying*

$$\lim_{k\to\infty} D_k \;=\; I$$

Why will the TR steps also cover the unit sphere

Property

Any sequence $\{(x_k, s_k, \Delta_k)\}$ *generated by Algorithm* **Basic DFO-TRNS** *is such that*

$$s_k = -\Delta_k D_k g_k$$

with $D_k \in \mathbb{R}^{n\times n}$ *satisfying*

$$\lim_{k\to\infty} D_k = I$$

Theorem

Property is satisfied when TR subproblems are solved up to optimality

Main result

$\{x_k\}$ sequence generated by Algorithm **Basic DFO-TRNS**. ASSUME:

- x^* limit point of $\{x_k\}_K$ with $K \subseteq \{k : \Delta_{k+1} < \Delta_k\}$
- $\{g_k\}_K$ dense in the unit sphere

Then x^* Clarke stationary

How to deal with nonsmoothness in practice?

IDEA: Embed **Basic DFO-TRNS** into an existing code

- Exploit the fact that f differentiable almost everywhere
- Combine **Basic DFO-TRNS** with approach for smooth problems (for quadratic term $\frac{1}{2}s^\top B_k s$):
  - DFO-TR Algorithm [A.S. Bandeira, K. Scheinberg, L.N. Vicente, 2012]

How to deal with nonsmoothness in practice?

IDEA: Embed Basic DFO-TRNS into an existing code

- Exploit the fact that f differentiable almost everywhere
- Combine **Basic DFO-TRNS** with approach for smooth problems (for quadratic term $\frac{1}{2}s^\top B_k s$):
  - DFO-TR Algorithm [A.S. Bandeira, K. Scheinberg, L.N. Vicente, 2012]

We weight the quadratic term of the model:

$$\omega \frac{1}{2} s^\top B_k s$$

with $\omega \in [0, 1]$

Test problems

- 51 nonsmooth problems
- Dimension $10 \leq n \leq 30$
- From 2 different collections [L. Lukšan, J. Vlček, 2000] [J.J. Moré, S. M. Wild, 2009]

Numerical results

Test problems

- 51 nonsmooth problems
- Dimension $10 \leq n \leq 30$
- From 2 different collections [L. Lukšan, J. Vlček, 2000] [J.J. Moré, S. M. Wild, 2009]

Analysis of numerical performance

- Performance and data profiles used [J.J. Moré, S. M. Wild, 2009]
- Budget $=$ **10000** function evaluations

Numerical results

Test problems

- 51 nonsmooth problems
- Dimension $10 \leq n \leq 30$
- From 2 different collections [L. Lukšan, J. Vlček, 2000] [J.J. Moré, S. M. Wild, 2009]

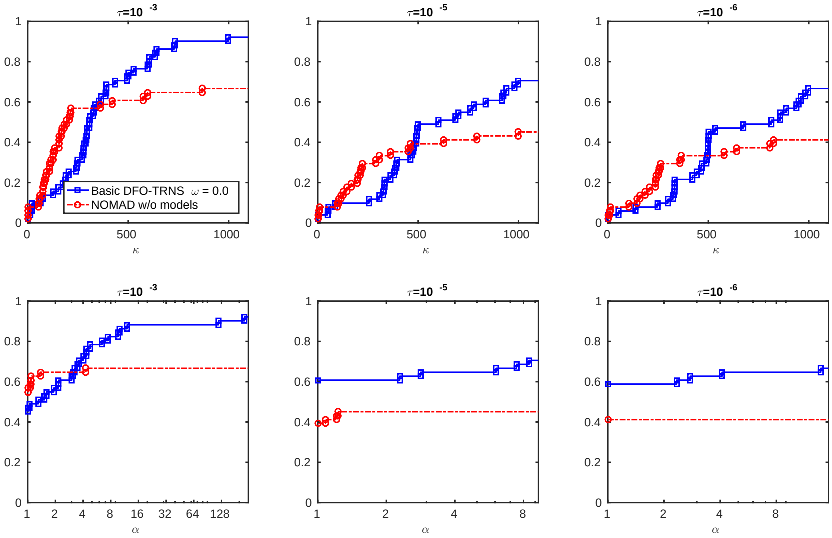
Analysis of numerical performance

- Performance and data profiles used [J.J. Moré, S. M. Wild, 2009]
- Budget $=$ **10000** function evaluations

Software Used

- **Basic DFO-TRNS** (Combined with DFO-TR) $\omega = 0$
- **NOMAD** package [C. Audet et al.] without models

Basic DFO-TRNS vs NOMAD (w/o models)



Comments

- **Basic DFO-TRNS** outperforms **NOMAD** (w/o models)
- Randomly generating g_k (when needed) helps!

Comments

- **Basic DFO-TRNS** outperforms **NOMAD** (w/o models)
- Randomly generating g_k (when needed) helps!
- However **Basic DFO-TRNS** wastes function evaluations
- And **NOMAD** (WITH models) is more efficient...

Comments

- **Basic DFO-TRNS** outperforms **NOMAD** (w/o models)
- Randomly generating g_k (when needed) helps!
- However **Basic DFO-TRNS** wastes function evaluations
- And **NOMAD** (WITH models) is more efficient...

QUESTION

How to improve performance of **Basic DFO-TRNS**?

Comments

- **Basic DFO-TRNS** outperforms **NOMAD** (w/o models)
- Randomly generating g_k (when needed) helps!
- However **Basic DFO-TRNS** wastes function evaluations
- And **NOMAD** (WITH models) is more efficient...

QUESTION

How to improve performance of **Basic DFO-TRNS**?

ANSWER

Improve description of linear term in the model

Comments

- **Basic DFO-TRNS** outperforms **NOMAD** (w/o models)
- Randomly generating g_k (when needed) helps!
- However **Basic DFO-TRNS** wastes function evaluations
- And **NOMAD** (WITH models) is more efficient...

QUESTION

How to improve performance of **Basic DFO-TRNS**?

ANSWER

Improve description of linear term in the model

IDEA

Use a Bundle-like approach to handle nonsmoothness

Presentation outline

1. Introduction
2. A basic TR method for black-box functions
3. An advanced TR model

How to handle nosmoothness in TR methods

Smooth setting

Trust-region model is given by the sum of a linear term and a quadratic one

$$\bar{m}_k(s) + \frac{1}{2}s^\top B_k s \;=\; f(x_k) + \nabla f(x_k)^\top s + \frac{1}{2}s^\top B_k s$$

How to handle nosmoothness in TR methods

Smooth setting

Trust-region model is given by the sum of a linear term and a quadratic one

$$\bar{m}_k(s) + \frac{1}{2}s^\top B_k s \;=\; f(x_k) + \nabla f(x_k)^\top s + \frac{1}{2}s^\top B_k s$$

Nonsmooth setting

Use a new linear term $m_k(s) \;=\; f(x_k) + f^\circ(x_k; s)$

$$\bar{m}_k(s) \;=\; \max_{\xi \in \partial f(x_k)} \left\{ f(x_k) + \xi^\top s \right\}$$

How to handle nosmoothness in TR methods

Smooth setting

Trust-region model is given by the sum of a linear term and a quadratic one

$$\bar{m}_k(s) + \frac{1}{2}s^\top B_k s \;=\; f(x_k) + \nabla f(x_k)^\top s + \frac{1}{2}s^\top B_k s$$

Nonsmooth setting

Use a new linear term $m_k(s) \;=\; f(x_k) + f^\circ(x_k; s)$

$$\bar{m}_k(s) \;=\; \max_{\xi \in \partial f(x_k)} \left\{ f(x_k) + \xi^\top s \right\}$$

Remark

Since the set $\partial f(x_k)$ is unknown, the above model cannot be used in practice!

Classic bundle approach (convex case)

How to replace $\partial f(x_k)$

Exploit the information obtained on a set of points $\{y^j : j \in J_k\}$ approaching x_k, where J_k is an index set.

Classic bundle approach (convex case)

How to replace $\partial f(x_k)$

Exploit the information obtained on a set of points $\{y^j : j \in J_k\}$ approaching x_k, where J_k is an index set.

New model

Approximate model: $\bar{m}_k(s) \;=\; \max_{\xi \in \partial f(x_k)} \left\{ f(x_k) + \xi^\top s \right\}$

with the following model: $\bar{m}_k(s) \;=\; \max_{j \in J_k} \left\{ f(y^j) + (\xi^j)^\top (x_k + s - y^j) \right\}$

where $\xi^j \in \partial f(y^j), j \in J_k$.

Classic bundle approach (convex case)

How to replace $\partial f(x_k)$

Exploit the information obtained on a set of points $\{y^j : j \in J_k\}$ approaching x_k, where J_k is an index set.

New model

Approximate model: $\bar{m}_k(s) \; = \; \max_{\xi \in \partial f(x_k)} \left\{ f(x_k) + \xi^\top s \right\}$

with the following model: $\bar{m}_k(s) \; = \; \max_{j \in J_k} \left\{ f(y^j) + (\xi^j)^\top (x_k + s - y^j) \right\}$

where $\xi^j \in \partial f(y^j), j \in J_k$.

How to rewrite the model

Final model is

$$\bar{m}_k(s) \; = \; \max_{j \in J_k} \left\{ f(x_k) + (\xi^j)^\top s - \beta_k^j \right\}$$

with $\beta_k^j \; = \; f(x_k) + (\xi^j)^\top (y^j - x_k) - f(y^j)$ displacement related to point y^j.

The derivative-free context

Main Issue

- Element $\xi \in \partial f(y)$ cannot be computed.
- How to adapt the bundle approach to our derivative-free setting?

The derivative-free context

Main Issue

- Element $\xi \in \partial f(y)$ cannot be computed.
- How to adapt the bundle approach to our derivative-free setting?

Idea: Replace the information of ξ

Use a set of randomly generated normalized directions

$$G_k \;=\; \{g_i : \|g_i\| = 1, i \in I_k\}$$

where I_k is another index set.

A nonsmooth model

Displacements

Compute for each $(i, j) \in I_k \times J_k$, the displacements

$$\beta_k^{ij} \;=\; \max\left\{0, f(x_k) - f(y_k^j) + (g_i)^\top (y_k^j - x_k)\right\}$$

A nonsmooth model

Displacements

Compute for each $(i, j) \in I_k \times J_k$, the displacements

$$\beta_k^{ij} \; = \; \max\left\{0, f(x_k) - f(y_k^j) + (g_i)^\top (y_k^j - x_k)\right\}$$

New model

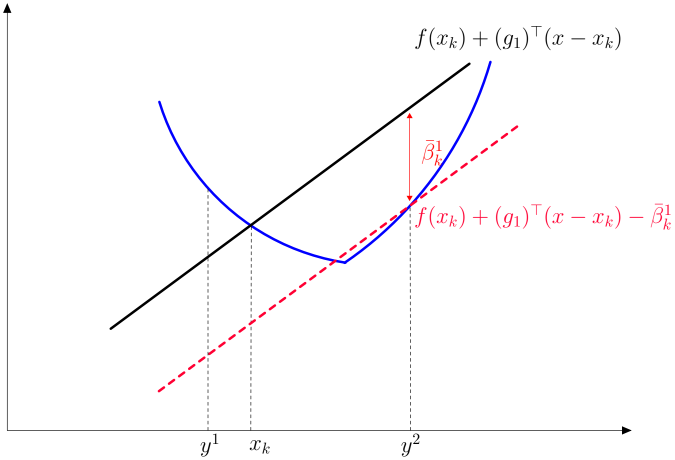
Introduce the following model

$$\bar{m}_k(s) \; = \; \max_{i \in I_k}\left\{ f(x_k) + (g_i)^\top s - \bar{\beta}_k^i \right\}$$

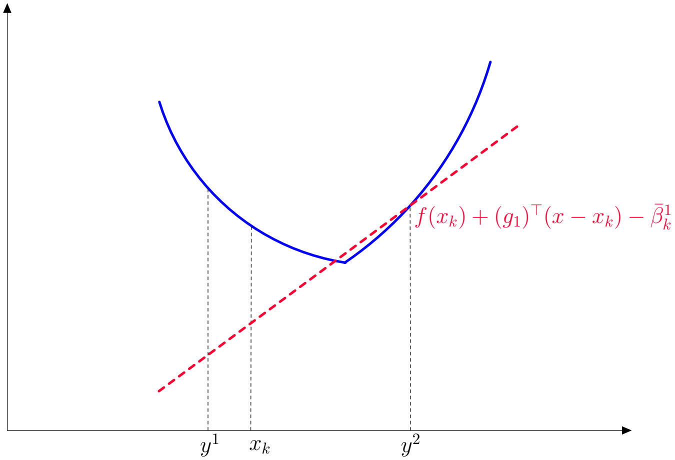
where

$$\bar{\beta}_k^i \; = \; \max_{j \in J_k}\{\beta_k^{ij}\}$$

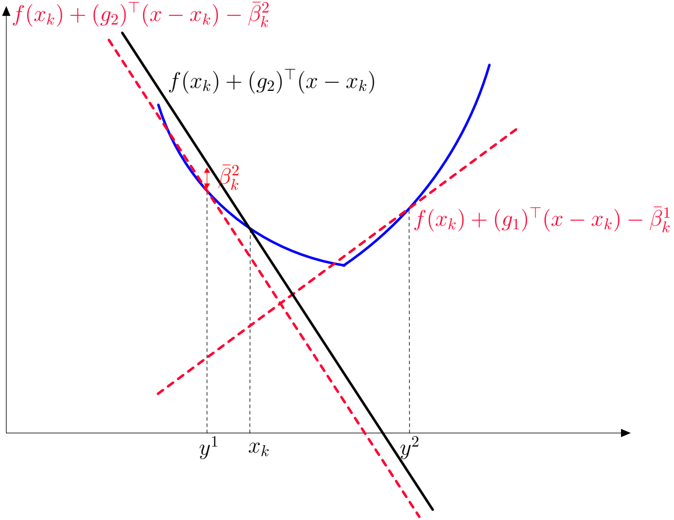
Example of our nonsmooth model in convex case



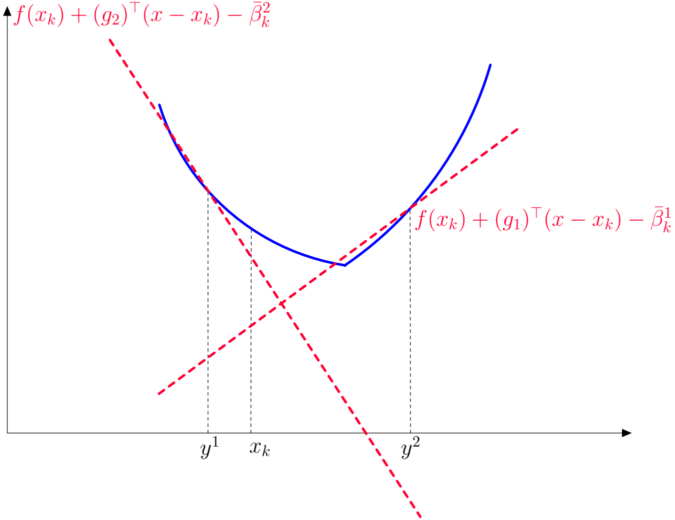
Example of our nonsmooth model in convex case



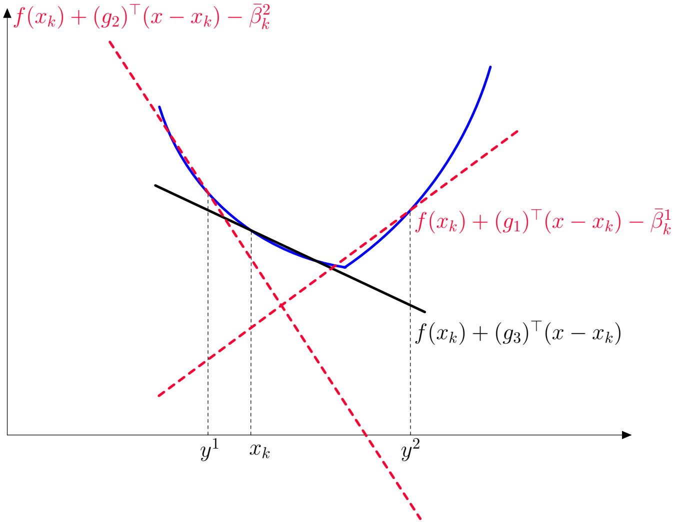
Example of our nonsmooth model in convex case



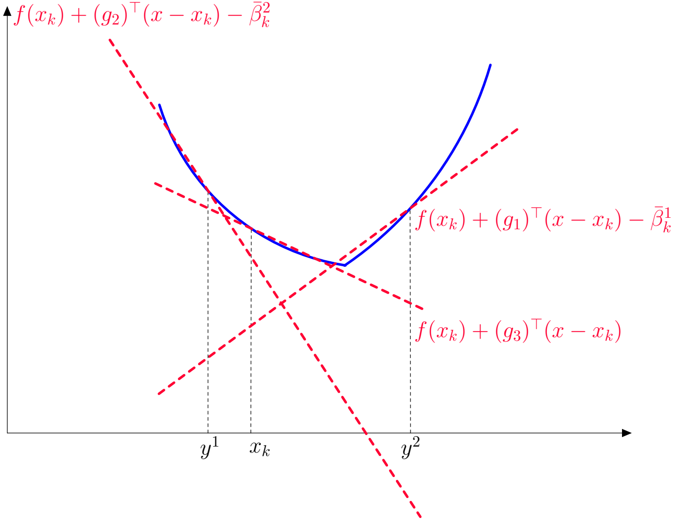
Example of our nonsmooth model in convex case



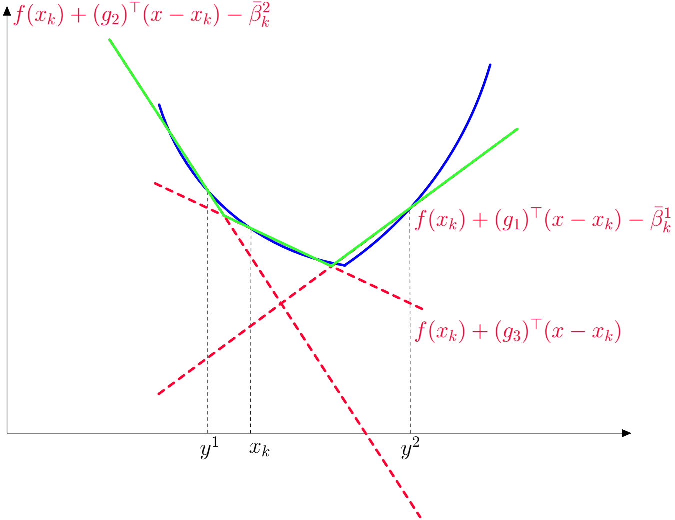
Example of our nonsmooth model in convex case



Example of our nonsmooth model in convex case



Example of our nonsmooth model in convex case



Dealing with nonconvexity

Bundle idea

- Hyperplanes might cut off solution.
- Lower all hyperplanes down.
- Amount will depend on the distance to the generating point.

Dealing with nonconvexity

Bundle idea

- Hyperplanes might cut off solution.
- Lower all hyperplanes down.
- Amount will depend on the distance to the generating point.

New displacements in the DFO case

Set

$$\beta_k^{ij} = \max\{0, f(x_k) - f(y_k^j) - (g_i)^\top (x_k - y_k^j) + \quad ? \quad \}$$

$$\bar{\beta}_k^i = \max_j \{\beta_k^{ij}\}$$

Dealing with nonconvexity

Bundle idea

- Hyperplanes might cut off solution.
- Lower all hyperplanes down.
- Amount will depend on the distance to the generating point.

New displacements in the DFO case

Set

$$\beta_k^{ij} = \max\{0, f(x_k) - f(y_k^j) - (g_i)^\top (x_k - y_k^j) + \gamma \|x_k - y_k^j\|^2\}$$

$$\bar{\beta}_k^i = \max_j \{\beta_k^{ij}\}$$

Dealing with nonconvexity

Bundle idea

- Hyperplanes might cut off solution.
- Lower all hyperplanes down.
- Amount will depend on the distance to the generating point.

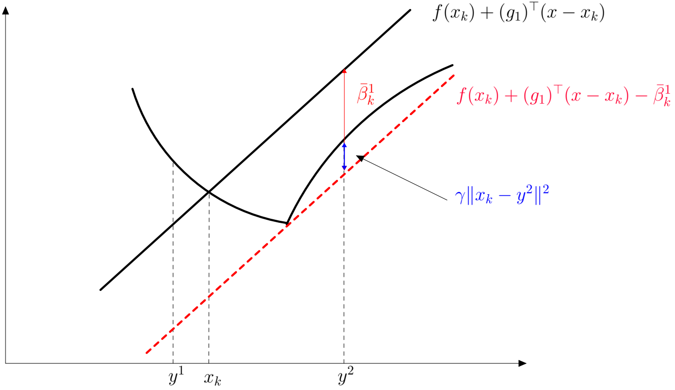
New displacements in the DFO case

Set

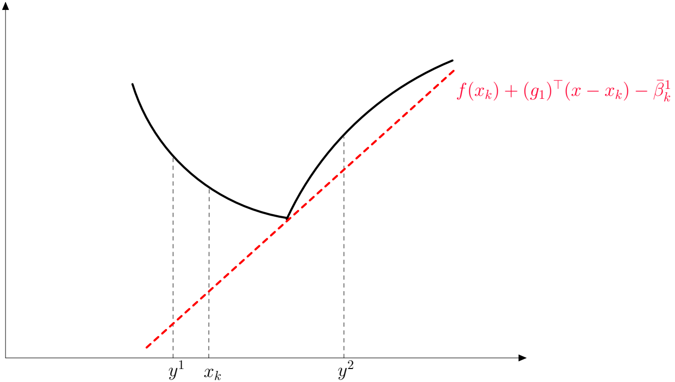
$$\beta_k^{ij} = \max\{0, f(x_k) - f(y_k^j) - (g_i)^\top (x_k - y_k^j) + \gamma \|x_k - y_k^j\|^2\}$$

$$\bar{\beta}_k^i = \max_j \{\beta_k^{ij}\}$$

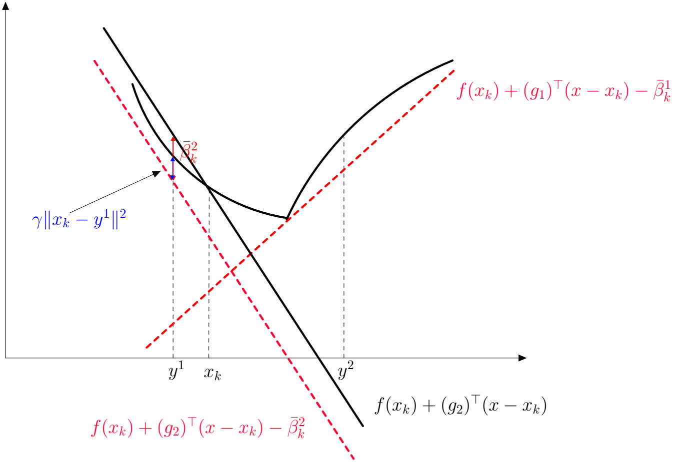
Example of our nonsmooth model in nonconvex case



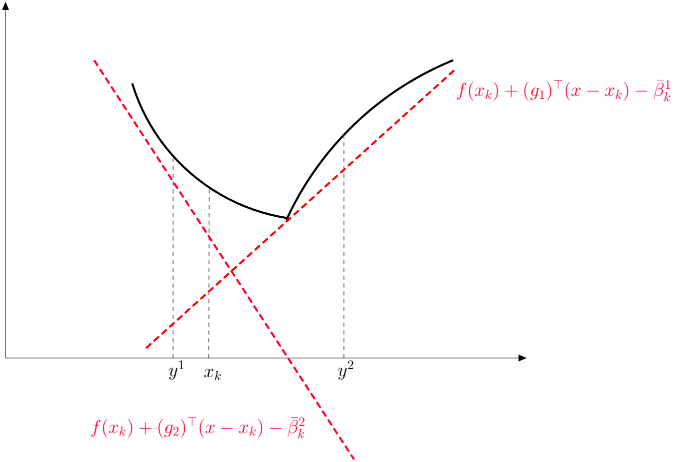
Example of our nonsmooth model in nonconvex case



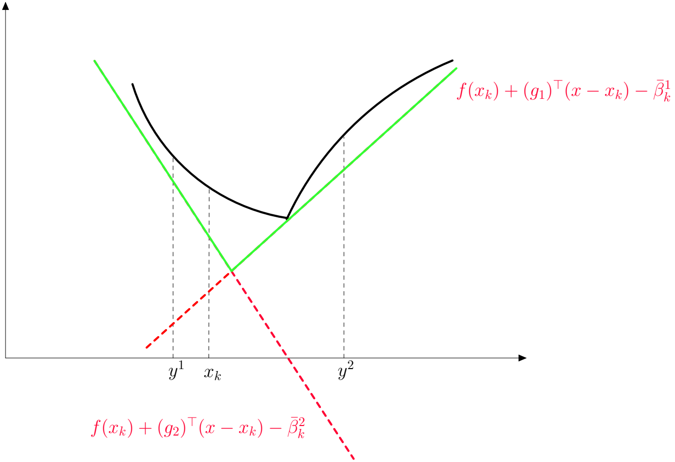
Example of our nonsmooth model in nonconvex case



Example of our nonsmooth model in nonconvex case



Example of our nonsmooth model in nonconvex case



Our nonsmooth TR model

Nonsmooth approximating TR model

$$m_k(s) \;=\; \max_{i\in I_k}\left\{ f(x_k) + (g_i)^\top s - \tilde{\beta}_k^i \right\} + \frac{1}{2} s^\top B_k s$$

where

- I_k is an index set for generated directions g_i
- $\tilde{\beta}_k^i$ are as in the pictures...
- B_k is a symmetric matrix built out from interpolation or regression on a sample set of points

A revelation: the convex combination vector

The TR is equivalent to:

$$\begin{aligned} \min_{s,\alpha} \quad & \tfrac{1}{2} s^\top B_k s + \alpha \\ \text{s.t.} \quad & [f(x_k) - \tilde{\beta}_k^i] + (g_i)^\top s \;\le\; \alpha, \quad \forall\, i \in I_k \\ & \|s\|^2 \;\le\; \Delta_k^2 \end{aligned}$$

A revelation: the convex combination vector

The TR is equivalent to:

$$\begin{aligned} \min_{s,\alpha} \quad & \tfrac{1}{2} s^\top B_k s + \alpha \\ \text{s.t.} \quad & [f(x_k) - \tilde{\beta}_k^i] + (g_i)^\top s \;\le\; \alpha, \quad \forall\, i \in I_k \\ & \|s\|^2 \;\le\; \Delta_k^2 \end{aligned}$$

Such a step solves an auxiliary problem:

$$\begin{aligned} \min \quad & \tilde{m}_k(s) = f(x_k) + \tilde{g}_k{}^\top s + \frac{1}{2} s^\top B_k s \\ \text{s.t.} \quad & \|s\|^2 \le \Delta_k^2 \end{aligned}$$

where

$$\tilde{g}_k \;=\; \sum_{i \in I_k} \lambda_i g_i \quad \text{with} \quad \sum_{i \in I_k} \lambda_i = 1, \quad \text{and} \quad \lambda_i \ge 0,\ \forall i \in I_k$$

and λ are the multipliers associated with the α constraints

Algorithm Advanced DFO-TRNS

Initialization. Select $x_0 \in \mathbb{R}^n$, $\eta, \gamma > 0$, $0 < \gamma_1 < 1 \leq \gamma_2$, $\bar{\epsilon} > 0$, $\Delta_0 > 0$, and $p > 0$. Set $G_0 = \emptyset$.

For $k = 0, 1 \ldots$

Generate randomly g_k in the unit sphere. Build a symmetric matrix B_k.

Set $G_k = G_{k-1} \cup \{g_k\}$.

Let s be a solution of NTRS for this G_k, and λ the associate multipliers.

If $\|\tilde{g}_k\| < \bar{\epsilon}\Delta_k^{\frac{1}{2}}$ **Then** Reset $G_k = \{g_k\}$. Let s be a solution of NTRS for G_k.

Set $s_k = s$ and $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{\|s_k\|^{p+1}}$.

If $\rho_k \geq \eta$

Then set $\texttt{SUCCESS} \leftarrow \texttt{true}$, $x_{k+1} \leftarrow x_k + s_k$, $\Delta_{k+1} \leftarrow \gamma_2 \Delta_k$,

Else set $\texttt{SUCCESS} \leftarrow \texttt{false}$, $x_{k+1} \leftarrow x_k$, $\Delta_{k+1} \leftarrow \gamma_1 \Delta_k$.

End If

End For

Convergence analysis

Assumption on the sample set

Points in the sample set $\{y_k^j : \ j \in J_k\}$ verify

$$\|x_k - y_k^j\| \ \leq \ \gamma \Delta_k, \quad \forall j \in J_k$$

Comments

- Assumptions and results are the **SAME** as for the basic version
- Convergence analysis follows **REMARKABLY** the lines of the basic version
- Main difference: use of the convex combination vector $\tilde{g}_k$ instead of g_k

Preliminary Numerical Results

Problems & Analysis

- Same problems as before
- Performance and data profiles used
- Budget = **10000** function evaluations

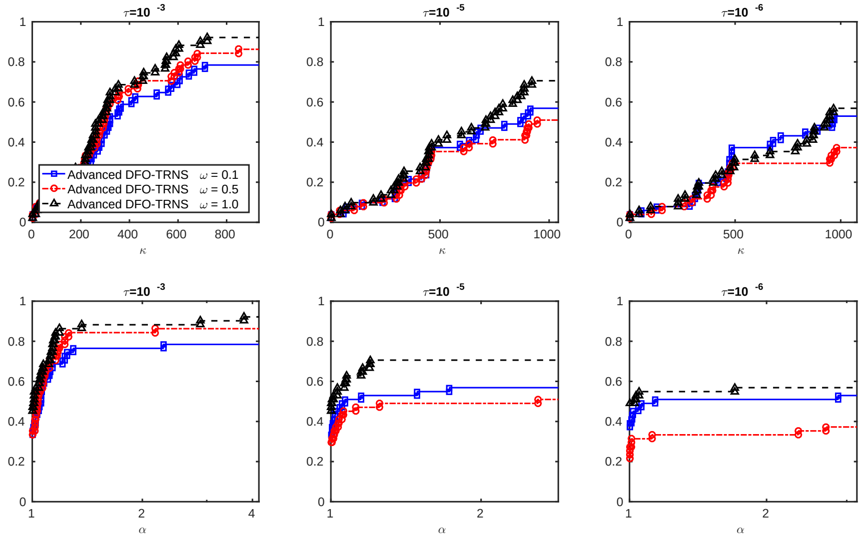
Software Used

- **Basic DFO-TRNS** (Combined with DFO-TR)
- **Advanced DFO-TRNS** (Combined with DFO-TR)
- **NOMAD** package [C. Audet et al.]

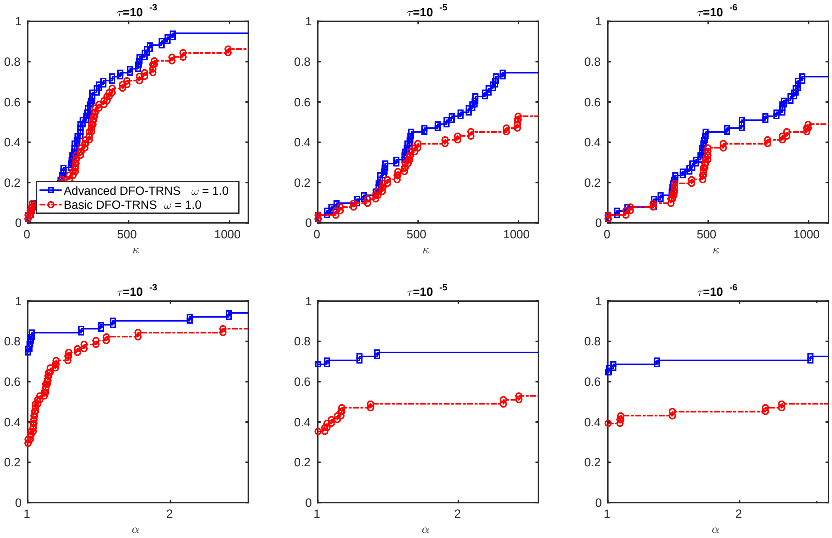
Details

- Different ω for the quadratic term in the models
- **NOMAD** used was WITH models

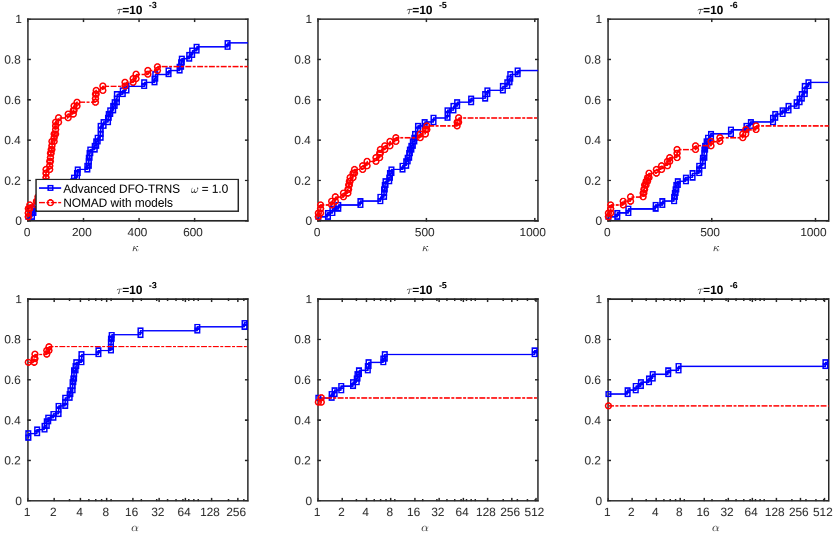
Comparison of **Adv. DFO-TRNS** varying ω's in $\frac{\omega}{2}s^\top B_k s$



Basic vs Adv. DFO-TRNS with $\omega = 1$



Adv. DFO-TRNS $(\omega = 1)$ vs NOMAD with models



Conclusions

- Proposed and analyzed model-based DFO methods for nonsmooth **black-box** functions

Conclusions

- Proposed and analyzed model-based DFO methods for nonsmooth **black-box** functions
- Basic approach (random linear term):
  - Good results when compared with **NOMAD** w/o models
  - Quite robust but not really efficient

Conclusions

- Proposed and analyzed model-based DFO methods for nonsmooth **black-box** functions
- Basic approach (random linear term):
  - Good results when compared with **NOMAD** w/o models
  - Quite robust but not really efficient
- Bundle-like approach (random max-linear term):
  - Better than Basic approach
  - Good results when compared with **NOMAD** using models

Conclusions

- Proposed and analyzed model-based DFO methods for nonsmooth **black-box** functions
- Basic approach (random linear term):
  - Good results when compared with **NOMAD** w/o models
  - Quite robust but not really efficient
- Bundle-like approach (random max-linear term):
  - Better than Basic approach
  - Good results when compared with **NOMAD** using models
- Matlab code available upon request
- Open questions: How to address larger instances? What if f is stochastic?

Conclusions

- Proposed and analyzed model-based DFO methods for nonsmooth **black-box** functions
- Basic approach (random linear term):
  - Good results when compared with **NOMAD** w/o models
  - Quite robust but not really efficient
- Bundle-like approach (random max-linear term):
  - Better than Basic approach
  - Good results when compared with **NOMAD** using models
- Matlab code available upon request
- Open questions: How to address larger instances? What if f is stochastic?

For further details: G. Liuzzi, S. Lucidi, F. Rinaldi and L.N. Vicente, *Trust-region methods for the derivative-free optimization of nonsmooth black-box functions*, SIAM Journal on Optimization, 29 (2019) 3012-3035