Recent Progress on Derivative-Free Trust-Region Methods

Luis Nunes Vicente
University of Coimbra

Sapienza – Università di Roma
June 28, 2016

Seminar Talk # 3

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

- Trust-region methods, based on the restricted minimization of models built from sample sets.

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

- Trust-region methods, based on the restricted minimization of models built from sample sets.

We will talk about trust-region methods for DFO:

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

- Trust-region methods, based on the restricted minimization of models built from sample sets.

We will talk about trust-region methods for DFO:

- Smooth obj. functions (worst case complexity, 1st and 2nd order).

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

- Trust-region methods, based on the restricted minimization of models built from sample sets.

We will talk about trust-region methods for DFO:

- Smooth obj. functions (worst case complexity, 1st and 2nd order).
- Non-smooth obj. functions (smoothing and composite approaches).

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

- Trust-region methods, based on the restricted minimization of models built from sample sets.

We will talk about trust-region methods for DFO:

- Smooth obj. functions (worst case complexity, 1st and 2nd order).
- Non-smooth obj. functions (smoothing and composite approaches).
- Probabilistic methods for deterministic obj. functions.

There are two main classes of rigorous methods in DFO

- Direct-search methods, of directional type.

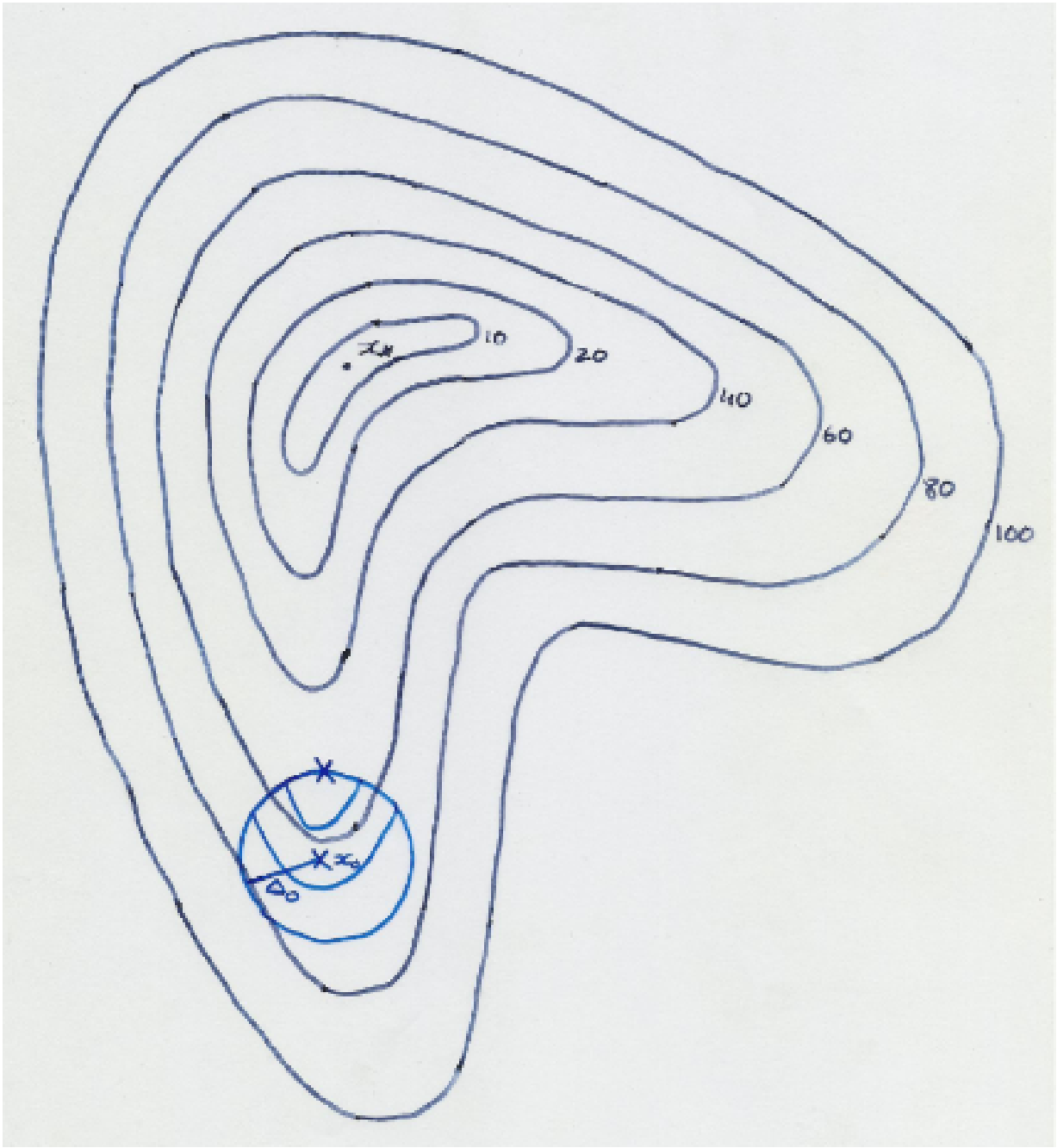
  Achieve descent by using directions spanning positively the search space or randomly generated, and moving in the directions of the best points.

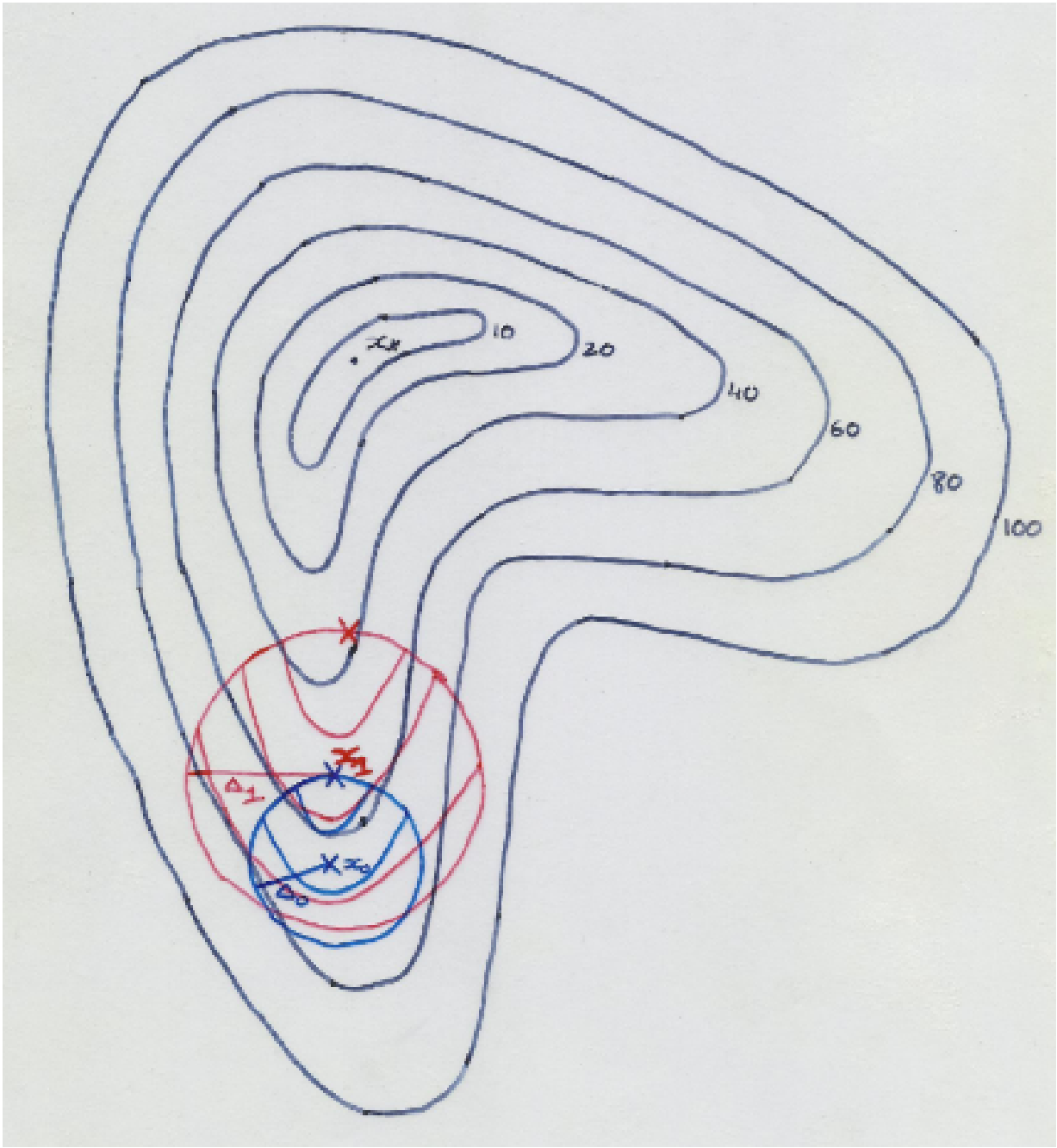
- Trust-region methods, based on the restricted minimization of models built from sample sets.

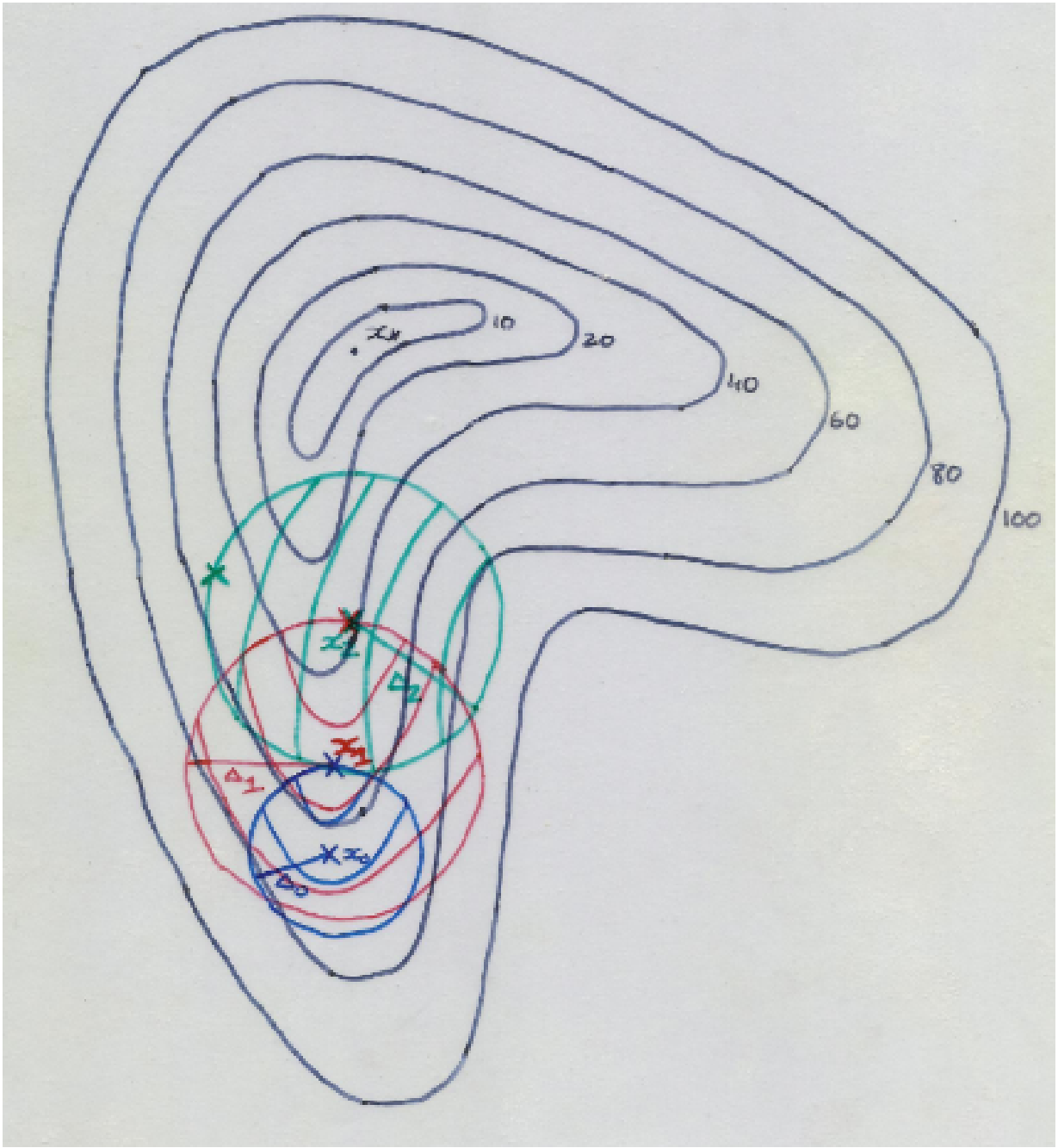
We will talk about trust-region methods for DFO:

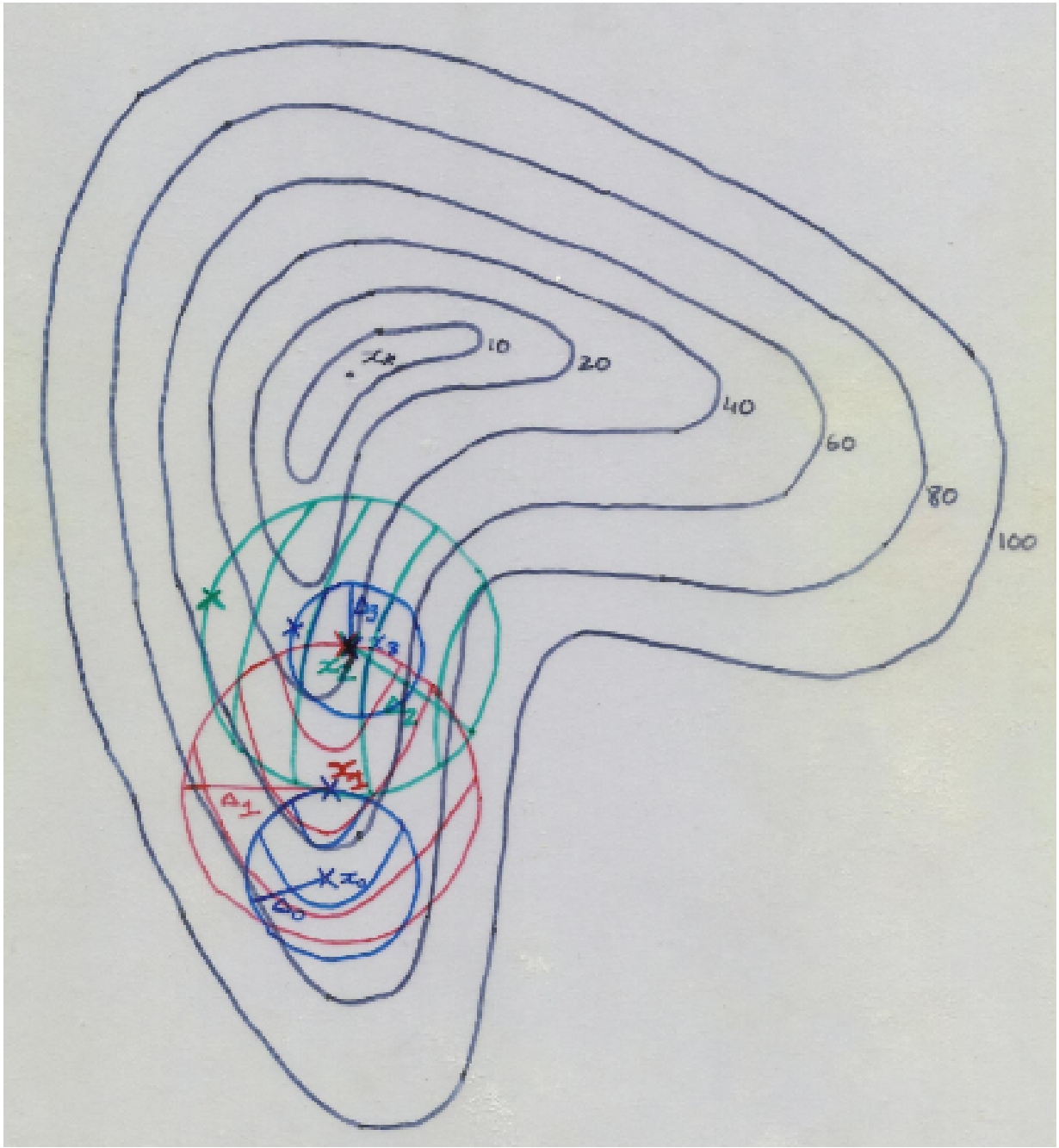
- Smooth obj. functions (worst case complexity, 1st and 2nd order).
- Non-smooth obj. functions (smoothing and composite approaches).
- Probabilistic methods for deterministic obj. functions.
- Stochastic obj. functions.


10
20
40
60
80
100


10
20
40
60
80
100


10
20
40
60
80
100


10
20
40
60
80
100


10
20
40
60
80
100

Trust-region methods

- One typically minimizes a model m in a trust region $B(x; \Delta)$:

Trust-region subproblem

$$\min_{y \in B(x;\Delta)} m(y)$$

Trust-region methods

- One typically minimizes a model m in a trust region $B(x;\Delta)$:

Trust-region subproblem

$$\min_{y \in B(x;\Delta)} m(y)$$

In derivative-based optimization, one could use:

1st order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top H (y - x)$$

Trust-region methods

- One typically minimizes a model m in a trust region $B(x;\Delta)$:

Trust-region subproblem

$$\min_{y \in B(x;\Delta)} m(y)$$

In derivative-based optimization, one could use:

1st order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y-x) + \frac{1}{2}(y-x)^\top H (y-x)$$

or a 2nd order Taylor:

$$m(y) = f(x) + \nabla f(x)^\top (y-x) + \frac{1}{2}(y-x)^\top \nabla^2 f(x)(y-x)$$

Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

- It is $\mathcal{C}^1$ with Lipschitz continuous gradient.

Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

- It is $\mathcal{C}^1$ with Lipschitz continuous gradient.
- The following error bounds hold:

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^2 \qquad \forall y \in B(x;\Delta).$$

Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

- It is $\mathcal{C}^1$ with Lipschitz continuous gradient.
- The following error bounds hold:

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^2 \qquad \forall y \in B(x;\Delta).$$

For a class of fully linear models, the (unknown) constants $\kappa_{ef}, \kappa_{eg} > 0$ must be independent of x and Δ.

Fully linear models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully linear if

- It is $\mathcal{C}^1$ with Lipschitz continuous gradient.
- The following error bounds hold:

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^2 \qquad \forall y \in B(x;\Delta).$$

For a class of fully linear models, the (unknown) constants $\kappa_{ef}, \kappa_{eg} > 0$ must be independent of x and Δ.

Fully linear models can be quadratic (or even nonlinear).

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

- It is $\mathcal{C}^2$ with Lipschitz continuous Hessian.

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

- It is $\mathcal{C}^2$ with Lipschitz continuous Hessian.
- The following error bounds hold:

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta^2 \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^3 \qquad \forall y \in B(x;\Delta).$$

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

- It is $\mathcal{C}^2$ with Lipschitz continuous Hessian.
- The following error bounds hold:

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta^2 \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^3 \qquad \forall y \in B(x;\Delta).$$

For a class of fully quadratic models, the (unknown) constants $\kappa_{ef}, \kappa_{eg}, \kappa_{eh} > 0$ must be independent of x and Δ.

Fully quadratic models

Given a point x and a trust-region radius Δ, a model $m(y)$ around x is called fully quadratic if

- It is $\mathcal{C}^2$ with Lipschitz continuous Hessian.
- The following error bounds hold:

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\,\Delta \qquad \forall y \in B(x;\Delta)$$

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\,\Delta^2 \qquad \forall y \in B(x;\Delta)$$

$$|f(y) - m(y)| \leq \kappa_{ef}\,\Delta^3 \qquad \forall y \in B(x;\Delta).$$

For a class of fully quadratic models, the (unknown) constants $\kappa_{ef}, \kappa_{eg}, \kappa_{eh} > 0$ must be independent of x and Δ.

Fully quadratic models are only necessary for global convergence/WCC to 2nd order stationary points.

Trust-region methods

Trust-region methods for DFO typically:

- Attempt to form quadratic models (by interpolation/regression and using polynomials or radial basis functions)

$$m_k(x_k + s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s$$

  based on (well poised) sample sets.

$\longrightarrow$ Well poisedness ensures fully linear or fully quadratic models.

Trust-region methods

Trust-region methods for DFO typically:

- Attempt to form quadratic models (by interpolation/regression and using polynomials or radial basis functions)

$$m_k(x_k + s) \;=\; f_k + g_k^\top s + \frac{1}{2} s^\top H_k s$$

  based on (well poised) sample sets.

$\longrightarrow$ Well poisedness ensures fully linear or fully quadratic models.

- Calculate a step s_k by approximately solving the trust-region subproblem

$$\min_{s \in B(x_k; \Delta_k)} \quad m_k(x_k + s).$$

Trust-region methods (a general framework)

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Trust-region methods (a general framework)

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

- Reduce Δ_k only if ρ_k is small and the model is FL/FQ — unsuccessful iterations.

Trust-region methods (a general framework)

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

- Reduce Δ_k only if ρ_k is small and the model is FL/FQ — unsuccessful iterations.

- Successful iterations occur when ρ_k is large (Δ_k kept or increased).

Trust-region methods (a general framework)

- Set x_{k+1} to $x_k + s_k$ (success) or to x_k (unsuccess) and update Δ_k depending on the value of

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

- Reduce Δ_k only if ρ_k is small and the model is FL/FQ — unsuccessful iterations.

- Successful iterations occur when ρ_k is large (Δ_k kept or increased).

- Requiring then model-improving iterations (when ρ_k is small and the model is not certifiably FL/FQ).

  $\longrightarrow$ Do not reduce Δ_k.

Trust-region methods (a general framework)

- Accept new iterates based on simple decrease, i.e., if

$$\rho_k > 0 \quad \Longleftrightarrow \quad f(x_k + s_k) < f(x_k),$$

as long as the model is FL/FQ — acceptable iterations (Δ_k reduced).

Trust-region methods (a general framework)

- Accept new iterates based on simple decrease, i.e., if

$$\rho_k > 0 \quad \Longleftrightarrow \quad f(x_k + s_k) < f(x_k),$$

as long as the model is FL/FQ — acceptable iterations (Δ_k reduced).

- Incorporate a criticality step (1st or 2nd order) when the 'stationarity' of the model is small.

$\longrightarrow$ Internal cycle of reductions of Δ_k — until model is well poised in $B(x_k; \|g_k\|)$.

Scheinberg and Toint (2010) showed that a criticality step is indeed necessary.

Global convergence of the CSV framework

Due to the criticality step, one has for successful iterations:

$$f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\|g_k\| \min\{\|g_k\|, \Delta_k\}) \geq \mathcal{O}(\Delta_k^2).$$

Global convergence of the CSV framework

Due to the criticality step, one has for successful iterations:

$$f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\|g_k\| \min\{\|g_k\|, \Delta_k\}) \geq \mathcal{O}(\Delta_k^2).$$

Theorem (Conn, Scheinberg, and Vicente, 2009)

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

Global convergence of the CSV framework

Due to the criticality step, one has for successful iterations:

$$f(x_k) - f(x_{k+1}) \;\geq\; \mathcal{O}(\|g_k\| \min\{\|g_k\|, \Delta_k\}) \geq\; \mathcal{O}(\Delta_k^2).$$

Theorem (Conn, Scheinberg, and Vicente, 2009)

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

Theorem (CSV, 2009)

Using fully linear models (when f is C^1 and bounded below),

$$\lim_{k \to +\infty} \|\nabla f(x_k)\| \;=\; 0.$$

Global convergence of the CSV framework

Due to the criticality step, one has for successful iterations:

$$f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\|g_k\| \min\{\|g_k\|, \Delta_k\}) \geq \mathcal{O}(\Delta_k^2).$$

Theorem (Conn, Scheinberg, and Vicente, 2009)

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

Theorem (CSV, 2009)

Using fully linear models (when f is C^1 and bounded below),

$$\lim_{k \to +\infty} \|\nabla f(x_k)\| = 0.$$

Theorem (CSV, 2009)

Using fully quadratic models (when f is C^2 and bounded below),

$$\lim_{k \to +\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}[\nabla^2 f(x_k)]\right\} = 0.$$

How to establish WCC for the CSV framework

- Change in ared/pred:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \quad c_1 \geq 0,\ p > 1.$$

How to establish WCC for the CSV framework

- Change in ared/pred:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \quad c_1 \geq 0,\ p > 1.$$

$\longrightarrow$ When $c_1 = 0$ we recover the traditional scenario.

How to establish WCC for the CSV framework

- Change in ared/pred:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \quad c_1 \geq 0,\ p > 1.$$

$\longrightarrow$ When $c_1 = 0$ we recover the traditional scenario.

$\longrightarrow$ We need the flexibility of p to achieve an optimal WCC for the smoothing approach.

How to establish WCC for the CSV framework

- Change in ared/pred:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \quad c_1 \geq 0,\ p > 1.$$

$\longrightarrow$ When $c_1 = 0$ we recover the traditional scenario.

$\longrightarrow$ We need the flexibility of p to achieve an optimal WCC for the smoothing approach.

- Each inner iteration of the criticality step is now considered as a regular trust-region iteration (leading to an optimal WCC).

How to establish WCC for the CSV framework

- Change in ared/pred:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - c_1 \Delta_k^p}{m_k(x_k) - m_k(x_k + s_k)} \quad c_1 \geq 0,\ p > 1.$$

  $\longrightarrow$ When $c_1 = 0$ we recover the traditional scenario.

  $\longrightarrow$ We need the flexibility of p to achieve an optimal WCC for the smoothing approach.

- Each inner iteration of the criticality step is now considered as a regular trust-region iteration (leading to an optimal WCC).

- The global convergence properties are retained.

WCC of DFTR (1st order, smooth case)

Remember $f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\Delta_k^2) + \mathcal{O}(\Delta_k^p)$ for successful iterations.

Lemma

If Δ_k is reduced, then $\|\nabla f(x_k)\| \leq \mathcal{O}(\Delta_k) + \mathcal{O}(\Delta_k^{p-1})$.

WCC of DFTR (1st order, smooth case)

Remember $f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\Delta_k^2) + \mathcal{O}(\Delta_k^p)$ for successful iterations.

Lemma

If Δ_k is reduced, then $\|\nabla f(x_k)\| \leq \mathcal{O}(\Delta_k) + \mathcal{O}(\Delta_k^{p-1})$.

Theorem

To drive $\|\nabla f(x_k)\|$ below $\epsilon \in (0,1)$, the # of successful iterations is

$$|\mathcal{S}| \leq \mathcal{O}(\epsilon^{-2}) = \mathcal{O}(\epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}).$$

WCC of DFTR (1st order, smooth case)

Remember $f(x_k) - f(x_{k+1}) \geq \mathcal{O}(\Delta_k^2) + \mathcal{O}(\Delta_k^p)$ for successful iterations.

Lemma

If Δ_k is reduced, then $\|\nabla f(x_k)\| \leq \mathcal{O}(\Delta_k) + \mathcal{O}(\Delta_k^{p-1})$.

Theorem

To drive $\|\nabla f(x_k)\|$ below $\epsilon \in (0,1)$, the # of successful iterations is

$$|\mathcal{S}| \leq \mathcal{O}(\epsilon^{-2}) = \mathcal{O}(\epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}).$$

Theorem

To drive $\|\nabla f(x_k)\|$ below $\epsilon \in (0,1)$, the # of other iterations is

$$|\mathcal{N}| \leq \mathcal{O}(|\mathcal{S}| + \epsilon^{-1}) = \mathcal{O}(|\mathcal{S}| + \epsilon^{-\frac{1}{\min(p-1,1)}}).$$

WCC of DFTR (1st order, smooth case)

Assumption

For FL models we assume $\kappa = \mathcal{O}(\sqrt{n}L_{\nabla f})$, *where* $\kappa = \max\{\kappa_{ef}, \kappa_{eg}\}$.

WCC of DFTR (1st order, smooth case)

Assumption

For FL models we assume $\kappa = \mathcal{O}(\sqrt{n} L_{\nabla f})$, *where* $\kappa = \max\{\kappa_{ef}, \kappa_{eg}\}$.

$p = 2$ is indeed optimal in the bounds:

Theorem

To drive the norm of the gradient below $\epsilon \in (0, 1)$, *DFTR takes at most*

$$\mathcal{O}\left(n\, L_{\nabla f}^2\, \epsilon^{-2}\right) \;=\; \mathcal{O}\left(\left(L_{\nabla f}\sqrt{n}\right)^{\frac{\max(p,2)}{\min(p-1,1)}} \epsilon^{-\frac{\max(p,2)}{\min(p-1,1)}}\right).$$

iterations.

WCC of DFTR (1st order, smooth case)

Corollary ($p = 2$)

The DFTR method generates a sequence $\{x_k\}_{k\geq 0}$ *such that:*

$$\min_{0\leq j\leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

WCC of DFTR (1st order, smooth case)

Corollary ($p = 2$)

The DFTR method generates a sequence $\{x_k\}_{k \geq 0}$ *such that:*

$$\min_{0 \leq j \leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-2}\right)$$

iterations to reduce the gradient below $\epsilon \in (0, 1)$.

WCC of DFTR (1st order, smooth case)

Corollary ($p = 2$)

The DFTR method generates a sequence $\{x_k\}_{k \geq 0}$ *such that:*

$$\min_{0 \leq j \leq k} \|\nabla f(x_j)\| \;=\; \mathcal{O}(1/\sqrt{k})$$

and takes at most

$$\mathcal{O}\left(n\,\epsilon^{-2}\right)$$

iterations to reduce the gradient below $\epsilon \in (0, 1)$.

- The # of fevals must be multiplied by n: $\mathcal{O}\left(n^2\,\epsilon^{-2}\right)$.

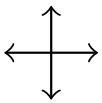
- R. Garmanjani, D. Júdice, and LNV, Trust-region methods without using derivatives: Worst case complexity and the non-smooth case, Tech. Report 15-03, Dept. Mathematics, Univ. Coimbra, (2015).

Optimal order of n in the WCC bounds

In Direct Search (DS) one knows that the n^2 factor in $\mathcal{O}(n^2\epsilon^{-2})$ is approximately optimal in the sense that $\min_{D \text{ PSS}} \frac{|D|}{\text{cm}(D)^2} \geq \mathcal{O}(n^2)$.

Optimal order of n in the WCC bounds

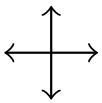
In Direct Search (DS) one knows that the n^2 factor in $\mathcal{O}(n^2\epsilon^{-2})$ is approximately optimal in the sense that $\min_{D \text{ PSS}} \frac{|D|}{\text{cm}(D)^2} \geq \mathcal{O}(n^2)$.

For instance, the PSS  gives us $\frac{2n}{(1/\sqrt{n})^2} = 2n^2$.

- M. Dodangeh, LNV, and Z. Zhang, On the optimal order of the worst case complexity of direct search, to appear in Optimization Letters.

Optimal order of n in the WCC bounds

In Direct Search (DS) one knows that the n^2 factor in $\mathcal{O}(n^2\epsilon^{-2})$ is approximately optimal in the sense that $\min_{D \text{ PSS}} \frac{|D|}{\text{cm}(D)^2} \geq \mathcal{O}(n^2)$.

For instance, the PSS  gives us $\frac{2n}{(1/\sqrt{n})^2} = 2n^2$.

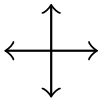
- M. Dodangeh, LNV, and Z. Zhang, On the optimal order of the worst case complexity of direct search, to appear in Optimization Letters.

Now,

$$\text{cm}(D) = \mathcal{O}(1/\sqrt{n}) \quad \longleftrightarrow \quad \kappa_{ef}, \kappa_{eg} = \mathcal{O}(\sqrt{n})$$

Optimal order of n in the WCC bounds

In Direct Search (DS) one knows that the n^2 factor in $\mathcal{O}(n^2\epsilon^{-2})$ is approximately optimal in the sense that $\min_{D \text{ PSS}} \frac{|D|}{\text{cm}(D)^2} \geq \mathcal{O}(n^2)$.

For instance, the PSS  gives us $\frac{2n}{(1/\sqrt{n})^2} = 2n^2$.

- M. Dodangeh, LNV, and Z. Zhang, On the optimal order of the worst case complexity of direct search, to appear in Optimization Letters.

Now,

$$\text{cm}(D) = \mathcal{O}(1/\sqrt{n}) \quad \longleftrightarrow \quad \kappa_{ef}, \kappa_{eg} = \mathcal{O}(\sqrt{n})$$

So, we expect that n^2 is also optimal for TRs.

WCC of DFTR (2nd order, smooth case)

The true criticality measure is $\sigma(x) = \max\left\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\right\}$.

WCC of DFTR (2nd order, smooth case)

The true criticality measure is $\sigma(x) = \max\left\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\right\}$.

LEMMA: If Δ_k is reduced, then $\sigma(x_k) \leq \mathcal{O}(\Delta_k)$.

WCC of DFTR (2nd order, smooth case)

The true criticality measure is $\sigma(x) = \max\left\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\right\}$.

LEMMA: If Δ_k is reduced, then $\sigma(x_k) \leq \mathcal{O}(\Delta_k)$.

The model criticality measure is $\sigma_k^m = \max\left\{\|g_k\|, -\lambda_{\min}(H_k)\right\}$.

WCC of DFTR (2nd order, smooth case)

The true criticality measure is $\sigma(x) = \max\left\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\right\}$.

LEMMA: If Δ_k is reduced, then $\sigma(x_k) \leq \mathcal{O}(\Delta_k)$.

The model criticality measure is $\sigma_k^m = \max\left\{\|g_k\|, -\lambda_{\min}(H_k)\right\}$.

Theorem

The DFTR method using FQ models and criticality measure σ_k^m takes at most

$$\mathcal{O}\left(n^3\,\epsilon^{-3}\right)$$

iterations to achieve $\sigma(x_k) \leq \epsilon$.

WCC of DFTR (2nd order, smooth case)

The true criticality measure is $\sigma(x) = \max\left\{\|\nabla f(x)\|, -\lambda_{\min}(\nabla^2 f(x))\right\}$.

LEMMA: If Δ_k is reduced, then $\sigma(x_k) \leq \mathcal{O}(\Delta_k)$.

The model criticality measure is $\sigma_k^m = \max\left\{\|g_k\|, -\lambda_{\min}(H_k)\right\}$.

Theorem

The DFTR method using FQ models and criticality measure σ_k^m takes at most

$$\mathcal{O}\left(n^3\,\epsilon^{-3}\right)$$

iterations to achieve $\sigma(x_k) \leq \epsilon$.

- The # of fevals must be multiplied by n^2: $\mathcal{O}\left(n^5\,\epsilon^{-3}\right)$.

Open issues

- Extension to constraints (closed and convex feasible regions).

Open issues

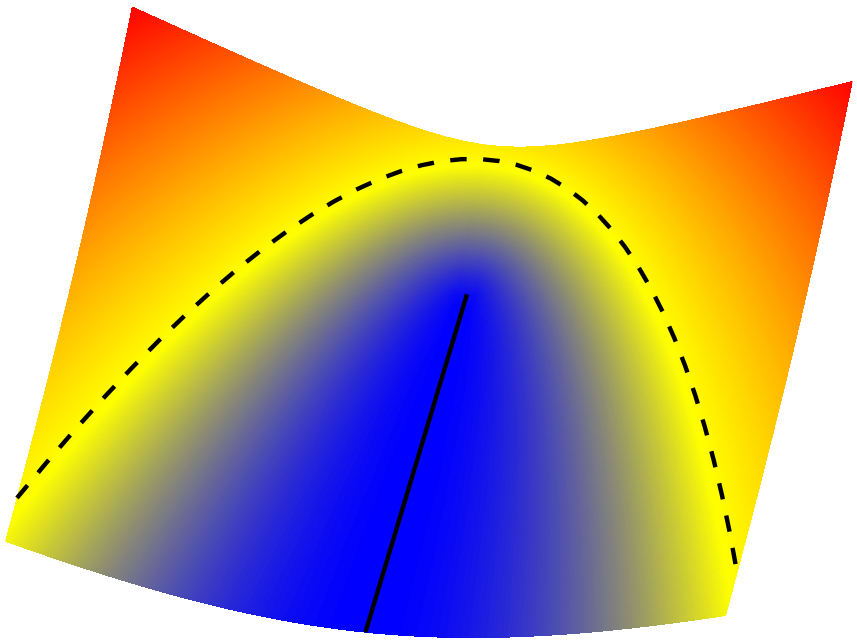
- Extension to constraints (closed and convex feasible regions).
- Particularization to convex (ϵ^{-1}) and strongly convex ($-\log(\epsilon)$).

Open issues

- Extension to constraints (closed and convex feasible regions).
- Particularization to convex (ϵ^{-1}) and strongly convex ($-\log(\epsilon)$).

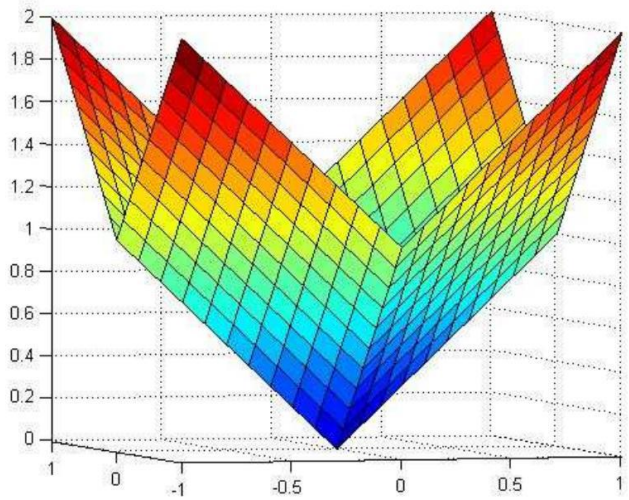
$\longrightarrow$ Do we have to exclude convex functions like this (as in DS)?

$$f(x,y) = \sqrt{x^2+y^2} - x$$



- M. Dodangeh and LNV, Worst case complexity of direct search under convexity, Mathematical Programming, 155 (2016) 307-332.

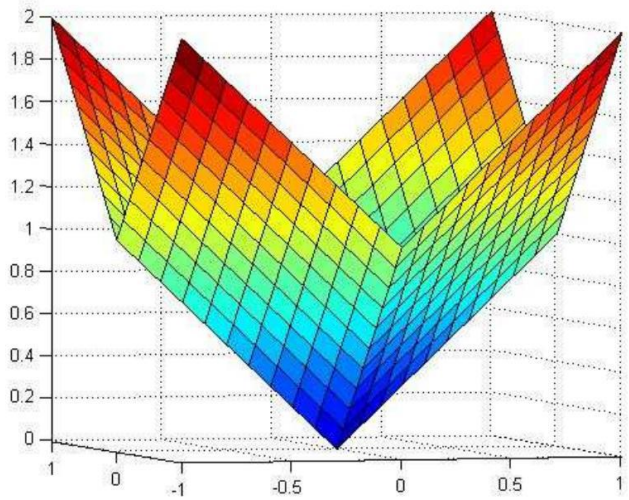
The non-smooth case



There are two avenues:

- Smoothing: apply smooth TR approach to smoothed instances.

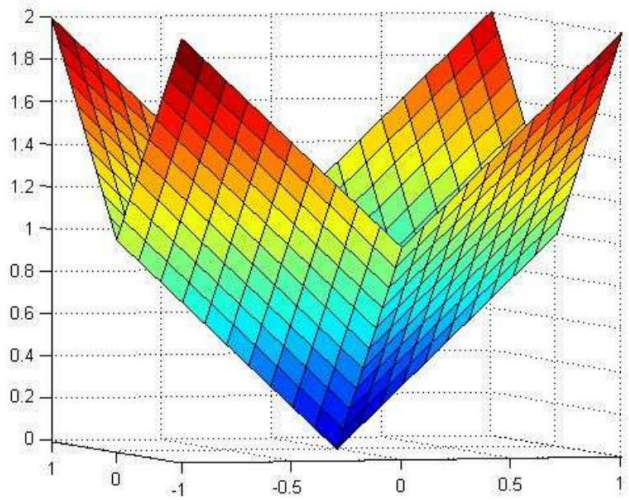
The non-smooth case



There are two avenues:

- Smoothing: apply smooth TR approach to smoothed instances.
- Composite: building non-smooth TR models.

The non-smooth case



There are two avenues:

- Smoothing: apply smooth TR approach to smoothed instances.
- Composite: building non-smooth TR models.

$\longrightarrow$ In both cases, some knowledge of the structure of the original non-smoothness is necessary

Smoothing functions

Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

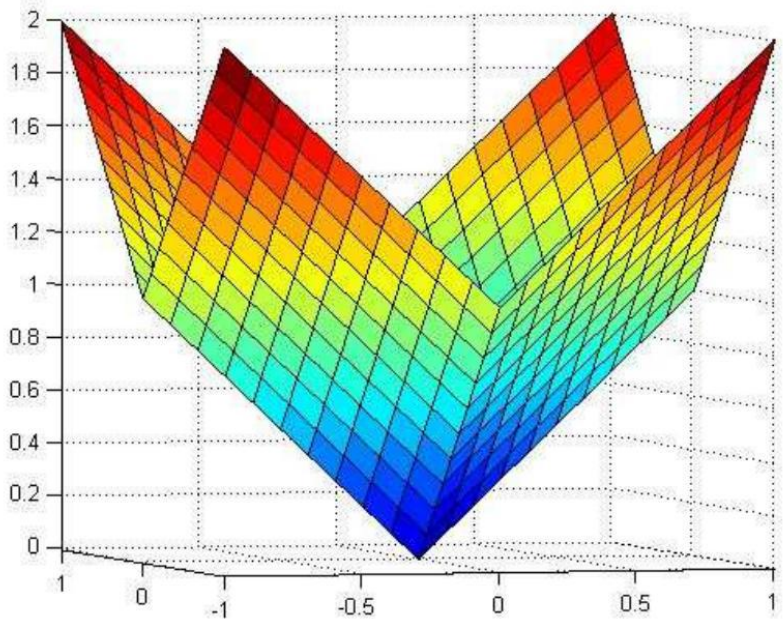
$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \; = \; f(x).$$

Smoothing functions

Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \;=\; f(x).$$

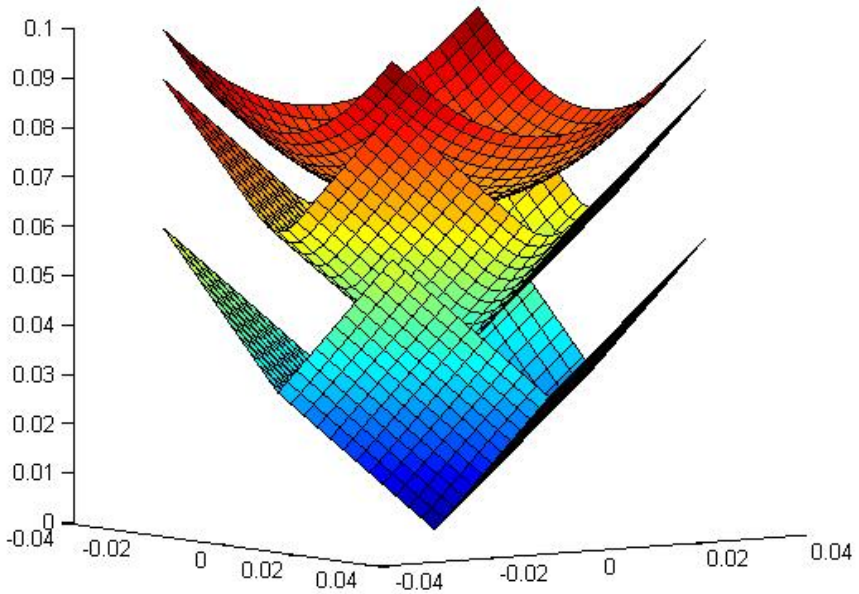


Smoothing functions

Definition

We call $\tilde{f} : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}$ *a smoothing function of* f *if,* $\forall \mu \in (0, \infty)$, $\tilde{f}(\cdot, \mu)$ *is* $\mathcal{C}^1$ *and,* $\forall x \in \mathbb{R}^n$,

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \; = \; f(x).$$



Smoothing functions (consistency & smoothness)

Definition

We say that x_* *is a stationary point associated with the smoothing function* $\tilde{f}$ *if* $0 \in G_{\tilde{f}}(x_*)$, *where*

$$G_{\tilde{f}}(x_*) \;=\; \{\textit{all limits of } \nabla \tilde{f}(x,\mu) \textit{ when } x \to x_* \textit{ and } \mu \to 0\}.$$

Smoothing functions (consistency & smoothness)

Definition

We say that x_* is a stationary point associated with the smoothing function $\tilde{f}$ if $0 \in G_{\tilde{f}}(x_*)$, where

$$G_{\tilde{f}}(x_*) \;=\; \{\textit{all limits of } \nabla \tilde{f}(x,\mu) \textit{ when } x \to x_* \textit{ and } \mu \to 0\}.$$

There are forms of building smoothing functions $\tilde{f}$ such that

- $\tilde{f}$ satisfies the gradient consistency property

$$\text{co } G_{\tilde{f}}(x_*) \;=\; \partial f(x_*).$$

Smoothing functions (consistency & smoothness)

Definition

We say that x_* is a stationary point associated with the smoothing function $\tilde{f}$ if $0 \in G_{\tilde{f}}(x_*)$, where

$$G_{\tilde{f}}(x_*) \;=\; \{\textit{all limits of } \nabla \tilde{f}(x,\mu) \textit{ when } x \to x_* \textit{ and } \mu \to 0\}.$$

There are forms of building smoothing functions $\tilde{f}$ such that

- $\tilde{f}$ satisfies the gradient consistency property

$$\text{co } G_{\tilde{f}}(x_*) \;=\; \partial f(x_*).$$

Thus, if $0 \in G_{\tilde{f}}(x_*) \subset \text{co}\, G_{\tilde{f}}(x_*)$, then $0 \in \partial f(x_*)$.

Smoothing functions (consistency & smoothness)

Definition

We say that x_* is a stationary point associated with the smoothing function $\tilde{f}$ if $0 \in G_{\tilde{f}}(x_*)$, where

$$G_{\tilde{f}}(x_*) \;=\; \{\textit{all limits of } \nabla \tilde{f}(x,\mu) \textit{ when } x \to x_* \textit{ and } \mu \to 0\}.$$

There are forms of building smoothing functions $\tilde{f}$ such that

- $\tilde{f}$ satisfies the gradient consistency property

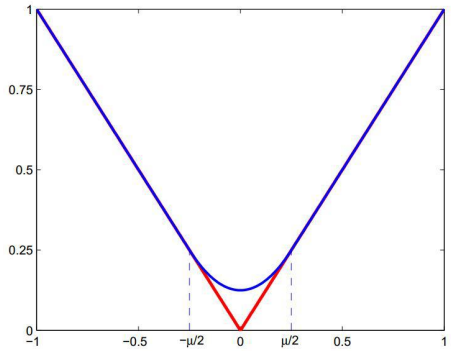
$$\text{co}\; G_{\tilde{f}}(x_*) \;=\; \partial f(x_*).$$

  Thus, if $0 \in G_{\tilde{f}}(x_*) \subset \text{co}\, G_{\tilde{f}}(x_*)$, then $0 \in \partial f(x_*)$.

- $L_{\nabla \tilde{f}} \;=\; \mathcal{O}\left(\frac{1}{\mu}\right)$.

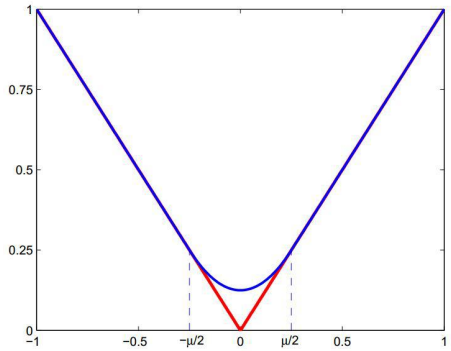
How to construct smoothing functions

Chen and Zhou introduced such a smoothing function $\tilde{s}(t, \mu)$ of $|t|$:



How to construct smoothing functions

Chen and Zhou introduced such a smoothing function $\tilde{s}(t, \mu)$ of $|t|$:

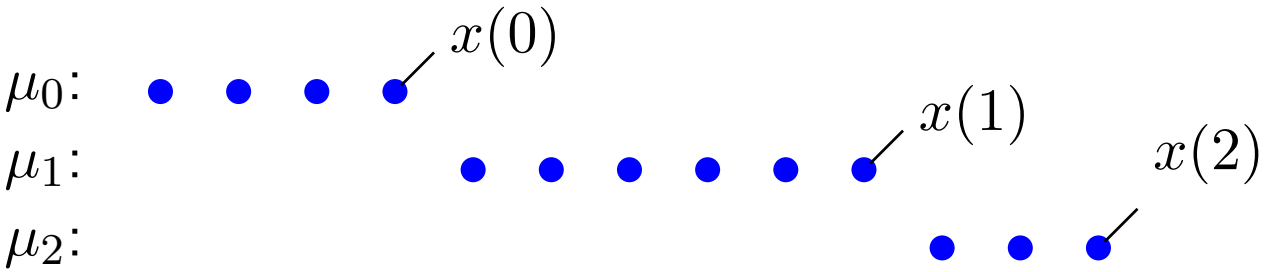


Similarly, for $\|F\|_1 = \sum_{i=1}^{m} |F_i|$,

$$\tilde{F}(x, \mu) = \sum_{i=1}^{m} \tilde{s}(F_i(x), \mu).$$

- R. Garmanjani and LNV, Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization, IMA Journal of Numerical Analysis, 33 (2013) 1008-1028.

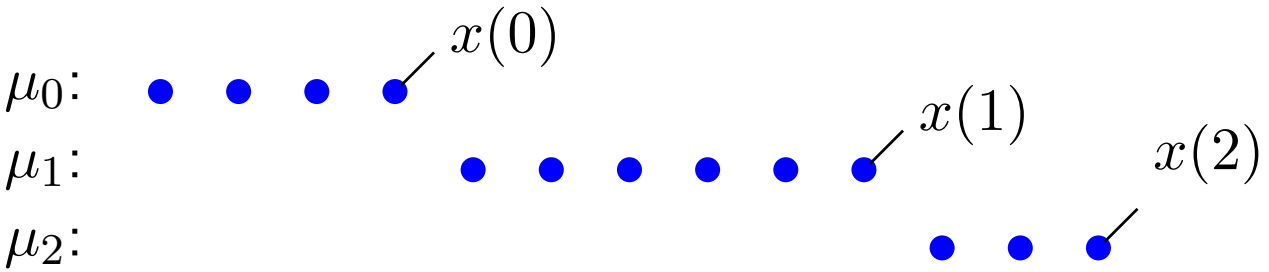
A class of smoothing TR methods



Initialization: Choose a function $r(\cdot)$ such that $\lim_{\mu \downarrow 0} r(\mu) = 0$.

Choose $\mu_0 > 0$, and $\sigma \in (0, 1)$

A class of smoothing TR methods



Initialization: Choose a function $r(\cdot)$ such that $\lim_{\mu \downarrow 0} r(\mu) = 0$.

Choose $\mu_0 > 0$, and $\sigma \in (0, 1)$

For $k = 0, 1, 2 \ldots$ (Until μ_k is suff. small)

- Apply DFTR to $\tilde{f}(\cdot, \mu_k)$ until trust-region radius $< r(\mu_k)$.
- Decrease the smoothing parameter: $\mu_{k+1} = \sigma \mu_k$.

Global convergence of smoothing DFTR (behavior of μ)

If we let DFTR run forever for a given k, then $\Delta \longrightarrow 0$. Thus

Global convergence of smoothing DFTR (behavior of μ)

If we let DFTR run forever for a given k, then $\Delta \longrightarrow 0$. Thus

Theorem

The smoothing parameter goes to zero: $\lim_{k \to \infty} \mu_k = 0.$

Global convergence of smoothing DFTR (behavior of μ)

If we let DFTR run forever for a given k, then $\Delta \longrightarrow 0$. Thus

Theorem

The smoothing parameter goes to zero: $\lim_{k\to\infty} \mu_k = 0.$

Theorem

1. $\lim_{k\to\infty} \Delta(k) = 0.$
2. $\exists x_*$ *and a subsequence* $K \subseteq \{(0), (1), \ldots\}$ *of unsucc. DFTR iterates such that* $x(k) \underset{K}{\longrightarrow} x_*$.

Global convergence of smoothing DFTR

Now,

$$\begin{aligned} \|\nabla \tilde{f}(x(k), \mu_k)\| &\leq \mathcal{O}(L_{\nabla \tilde{f}} \Delta(k)) + \mathcal{O}(\Delta(k)^{p-1}) \\ &\leq \mathcal{O}(L_{\nabla \tilde{f}} r(\mu_k)) + + \mathcal{O}(\Delta(k)^{p-1}). \end{aligned}$$

Global convergence of smoothing DFTR

Now,

$$\begin{aligned} \|\nabla \tilde{f}(x(k), \mu_k)\| &\leq \mathcal{O}(L_{\nabla \tilde{f}} \Delta(k)) + \mathcal{O}(\Delta(k)^{p-1}) \\ &\leq \mathcal{O}(L_{\nabla \tilde{f}} r(\mu_k)) + +\mathcal{O}(\Delta(k)^{p-1}). \end{aligned}$$

Thus, choosing $r(\cdot)$ appropriately, i.e., $r(\mu) = \mu^2$ when $L_{\nabla \tilde{f}} = \mathcal{O}\left(\frac{1}{\mu}\right)$:

Theorem

$$\lim_{k \in K} \|\nabla \tilde{f}(x(k), \mu_k)\| = 0$$

and x_* is stationary point associated with the smoothing function $\tilde{f}$.

WCC of smoothing DFTR

Theorem

Smoothing DFTR (with $c_1\Delta^p$ and $r(\mu) = \mu^q$) takes at most

$$\mathcal{O}\left(|\log(\xi)|\xi^{-pq}\right)$$

inner iterations to reduce the smoothing parameter μ below $\xi \in (0, 1)$.

WCC of smoothing DFTR

Theorem

Smoothing DFTR (with $c_1\Delta^p$ and $r(\mu) = \mu^q$) takes at most

$$\mathcal{O}\left(|\log(\xi)|\xi^{-pq}\right)$$

inner iterations to reduce the smoothing parameter μ below $\xi \in (0, 1)$.

WCC bounds come from this, and from

$$\begin{aligned}
\|\nabla \tilde{f}(x(k), \mu_k)\| &\leq \mathcal{O}(L_{\nabla \tilde{f}}\Delta(k)) + \mathcal{O}(\Delta(k)^{p-1}) \\
&\leq \mathcal{O}(\mu_k^{-1}\mu_k^q) + +\mathcal{O}((\mu_k^q)^{p-1}) \\
&= \mathcal{O}(\xi) \quad \text{when} \quad p = \frac{3}{2} \text{ and } q = 2.
\end{aligned}$$

WCC of smoothing DFTR

The optimal choices are: $c_1\Delta^p = c_1\Delta^{\frac{3}{2}}$ and $r(\mu) = \mu^2$.

WCC of smoothing DFTR

The optimal choices are: $c_1\Delta^p = c_1\Delta^{\frac{3}{2}}$ and $r(\mu) = \mu^2$.

Theorem

Smoothing DFTR (with such choices) takes at most

$$\mathcal{O}\left(n^{\frac{3}{2}}[-\log(\epsilon) + \log(n)]\epsilon^{-3}\right)$$

iterations to compute a point where μ and $\nabla \tilde{f}$ are $\mathcal{O}(\epsilon)$.

WCC of smoothing DFTR

The optimal choices are: $c_1\Delta^p = c_1\Delta^{\frac{3}{2}}$ and $r(\mu) = \mu^2$.

Theorem

Smoothing DFTR (with such choices) takes at most

$$\mathcal{O}\left(n^{\frac{3}{2}}[-\log(\epsilon) + \log(n)]\epsilon^{-3}\right)$$

iterations to compute a point where μ and $\nabla \tilde{f}$ are $\mathcal{O}(\epsilon)$.

- The # of function evaluations must be multiplied by n.

Composite DFTR

Consider

$$f = h(F),$$

with $h : \mathbb{R}^\ell \to \mathbb{R}$ non-smooth convex, and F smooth.

Composite DFTR

Consider

$$f = h(F),$$

with $h : \mathbb{R}^\ell \to \mathbb{R}$ non-smooth convex, and F smooth.

When there are derivatives (Cartis, Gould, and Toint, 2011),

$$l(x, s) \;=\; h(F(x) + J(x)s),$$

$$\Psi(x, \Delta) \;=\; l(x, 0) - \min_{\|s\| \leq \Delta} l(x, s), \quad \Psi(x, 1) \text{ is a criticality measure.}$$

Composite DFTR

Consider

$$f = h(F),$$

with $h : \mathbb{R}^\ell \to \mathbb{R}$ non-smooth convex, and F smooth.

When there are derivatives (Cartis, Gould, and Toint, 2011),

$$l(x, s) \;=\; h(F(x) + J(x)s),$$

$$\Psi(x, \Delta) \;=\; l(x, 0) - \min_{\|s\| \leq \Delta} l(x, s), \quad \Psi(x, 1) \text{ is a criticality measure.}$$

Without derivatives,

$$l^m(x, s) = h(m(x + s)), \text{ with } m(x + s) = F(x) + J^m(x)s$$

$$\Psi^m(x, \Delta) \;=\; l^m(x, 0) - \min_{\|s\| \leq \Delta} l^m(x, s).$$

Composite DFTR algorithm

Then, $\Psi^m(x,1)$ is a FL model of criticality measure:

$$|\Psi(x+t,1) - \Psi^m(x+t,1)| = \mathcal{O}(\Delta), \quad \forall t \in B(0;\Delta).$$

Composite DFTR algorithm

Then, $\Psi^m(x,1)$ is a FL model of criticality measure:

$$|\Psi(x+t,1) - \Psi^m(x+t,1)| = \mathcal{O}(\Delta), \quad \forall t \in B(0;\Delta).$$

Changes to DFTR algorithm:

- The TR subproblem is now $\min_{\|s\| \leq \Delta_k} l^m(x_k, s)$.

Composite DFTR algorithm

Then, $\Psi^m(x, 1)$ is a FL model of criticality measure:

$$|\Psi(x + t, 1) - \Psi^m(x + t, 1)| = \mathcal{O}(\Delta), \quad \forall t \in B(0; \Delta).$$

Changes to DFTR algorithm:

- The TR subproblem is now $\min_{\|s\| \leq \Delta_k} l^m(x_k, s)$.
- The predicted reduction $m_k(x_k) - m_k(x_k + s_k)$ becomes $\Psi^m(x_k, \Delta_k)$.

Composite DFTR algorithm

Then, $\Psi^m(x, 1)$ is a FL model of criticality measure:

$$|\Psi(x + t, 1) - \Psi^m(x + t, 1)| = \mathcal{O}(\Delta), \quad \forall t \in B(0; \Delta).$$

Changes to DFTR algorithm:

- The TR subproblem is now $\min_{\|s\| \leq \Delta_k} l^m(x_k, s)$.
- The predicted reduction $m_k(x_k) - m_k(x_k + s_k)$ becomes $\Psi^m(x_k, \Delta_k)$.
- The model criticality measure $\|g_k\|$ is replaced by $\Psi^m(x_k, 1)$.

Global convergence of composite DFTR

Theorem

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

Global convergence of composite DFTR

Theorem

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

Lemma

If Δ_k is reduced, then

$$\Delta_k \;\geq\; \mathcal{O}(\min\{\sqrt{\Psi_k}, \Psi_k\}).$$

Global convergence of composite DFTR

Theorem

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

Lemma

If Δ_k is reduced, then

$$\Delta_k \;\geq\; \mathcal{O}(\min\{\sqrt{\Psi_k}, \Psi_k\}).$$

Theorem

$$\liminf_{k \to \infty} \Psi_k \;=\; 0.$$

WCC of composite DFTR

Theorem

To drive Ψ *below* $\epsilon \in (0, 1)$, *composite DFTR needs*

$$|\mathcal{S}| \leq \mathcal{O}(n\epsilon^{-2}) \text{ sucessful iterations},$$

$$|\mathcal{N}| \leq \mathcal{O}(|\mathcal{S}| + \epsilon^{-1}) \text{ other iterations}.$$

Thus, a total of $\mathcal{O}(n\epsilon^{-2})$.

WCC of composite DFTR

Theorem

To drive Ψ *below* $\epsilon \in (0,1)$, *composite DFTR needs*

$$|\mathcal{S}| \leq \mathcal{O}(n\epsilon^{-2}) \text{ sucessful iterations},$$

$$|\mathcal{N}| \leq \mathcal{O}(|\mathcal{S}| + \epsilon^{-1}) \text{ other iterations}.$$

Thus, a total of $\mathcal{O}(n\epsilon^{-2})$.

- The # of fevals must be multiplied by $\ell\, n$: $\mathcal{O}\left(\ell\, n^2\, \epsilon^{-2}\right)$.

WCC of composite DFTR

Theorem

To drive Ψ *below* $\epsilon \in (0, 1)$, *composite DFTR needs*

$$|\mathcal{S}| \leq \mathcal{O}(n\epsilon^{-2}) \text{ sucessful iterations},$$

$$|\mathcal{N}| \leq \mathcal{O}(|\mathcal{S}| + \epsilon^{-1}) \text{ other iterations}.$$

Thus, a total of $\mathcal{O}(n\epsilon^{-2})$.

- The # of fevals must be multiplied by $\ell\, n$: $\mathcal{O}\left(\ell\, n^2\, \epsilon^{-2}\right)$.

$\longrightarrow$ The # of function evaluations derived here are better by a factor of $|\log \epsilon|$ than the $\mathcal{O}(|\log \epsilon|\epsilon^{-2})$ type bound derived by Grapiglia, Yuan, and Yuan (2014).

Smoothing DFTR code

For the smoothing code (Sdfo-tr), the smooth TR code (dfo-tr) is taken from Bandeira, Scheinberg, and LNV (2012):

- New sample points are only defined by the trust-region step $x + s$ (no model management iterations), as in Fasano, Morales, and Nocedal (2009).

Smoothing DFTR code

For the smoothing code (Sdfo-tr), the smooth TR code (dfo-tr) is taken from Bandeira, Scheinberg, and LNV (2012):

- New sample points are only defined by the trust-region step $x + s$ (no model management iterations), as in Fasano, Morales, and Nocedal (2009).
- Quadratic underdetermined models are built by minimum Frobenius norm minimization.

Smoothing DFTR code

For the smoothing code (Sdfo-tr), the smooth TR code (dfo-tr) is taken from Bandeira, Scheinberg, and LNV (2012):

- New sample points are only defined by the trust-region step $x + s$ (no model management iterations), as in Fasano, Morales, and Nocedal (2009).
- Quadratic underdetermined models are built by minimum Frobenius norm minimization.
- Points too far from the current iterate are thrown away (sort of a criticality step).

Smoothing DFTR code

For the smoothing code (Sdfo-tr), the smooth TR code (dfo-tr) is taken from Bandeira, Scheinberg, and LNV (2012):

- New sample points are only defined by the trust-region step $x + s$ (no model management iterations), as in Fasano, Morales, and Nocedal (2009).
- Quadratic underdetermined models are built by minimum Frobenius norm minimization.
- Points too far from the current iterate are thrown away (sort of a criticality step).
- Trust-region radius is not reduced when the sample set has less than $n + 1$ points.

Composite DFTR code

The composite code (Cdfo-tr) is an adaptation of the smooth one:

- In the TR subproblem

$$m_k(x_k + s) = F(x_k) + J^m(x_k)s,$$

where the rows of $J^m(x_k)$ are regression simplex gradients computed using the $2n$ points $x_k \pm \mathrm{e}_i \min(10^{-2}, \Delta_k)$.

Composite DFTR code

The composite code (Cdfo-tr) is an adaptation of the smooth one:

- In the TR subproblem

$$m_k(x_k + s) = F(x_k) + J^m(x_k)s,$$

  where the rows of $J^m(x_k)$ are regression simplex gradients computed using the $2n$ points $x_k \pm \mathrm{e}_i \min(10^{-2}, \Delta_k)$.

- The TR subproblem is an LP when using ℓ_∞ for defining the trust-region ball.

Composite DFTR code

The composite code (Cdfo-tr) is an adaptation of the smooth one:

- In the TR subproblem

$$m_k(x_k + s) = F(x_k) + J^m(x_k)s,$$

  where the rows of $J^m(x_k)$ are regression simplex gradients computed using the $2n$ points $x_k \pm \mathrm{e}_i \min(10^{-2}, \Delta_k)$.

- The TR subproblem is an LP when using ℓ_∞ for defining the trust-region ball.

- Due to fully linearity of the models, no critical or model-improvement iterations were considered.

Problems and testing

We used a set of 53 problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) = \|F(x)\|_1,$$

where F varies among 22 nonlinear vector functions of CUTEr with $2 \leq n \leq 12$ and different initial points, Moré and Wild (2009).

Problems and testing

We used a set of 53 problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) = \|F(x)\|_1,$$

where F varies among 22 nonlinear vector functions of CUTEr with $2 \leq n \leq 12$ and different initial points, Moré and Wild (2009).

Results are shown using data profiles (1500 fevals).

Problems and testing

We used a set of 53 problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) = \|F(x)\|_1,$$

where F varies among 22 nonlinear vector functions of CUTEr with $2 \leq n \leq 12$ and different initial points, Moré and Wild (2009).

Results are shown using data profiles (1500 fevals).

The smoothing version Sdfo-tr is compared with smoothing Direct Search (Ssid-psm), Custódio and LNV (2009).

Problems and testing

We used a set of 53 problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) = \|F(x)\|_1,$$

where F varies among 22 nonlinear vector functions of CUTEr with $2 \leq n \leq 12$ and different initial points, Moré and Wild (2009).

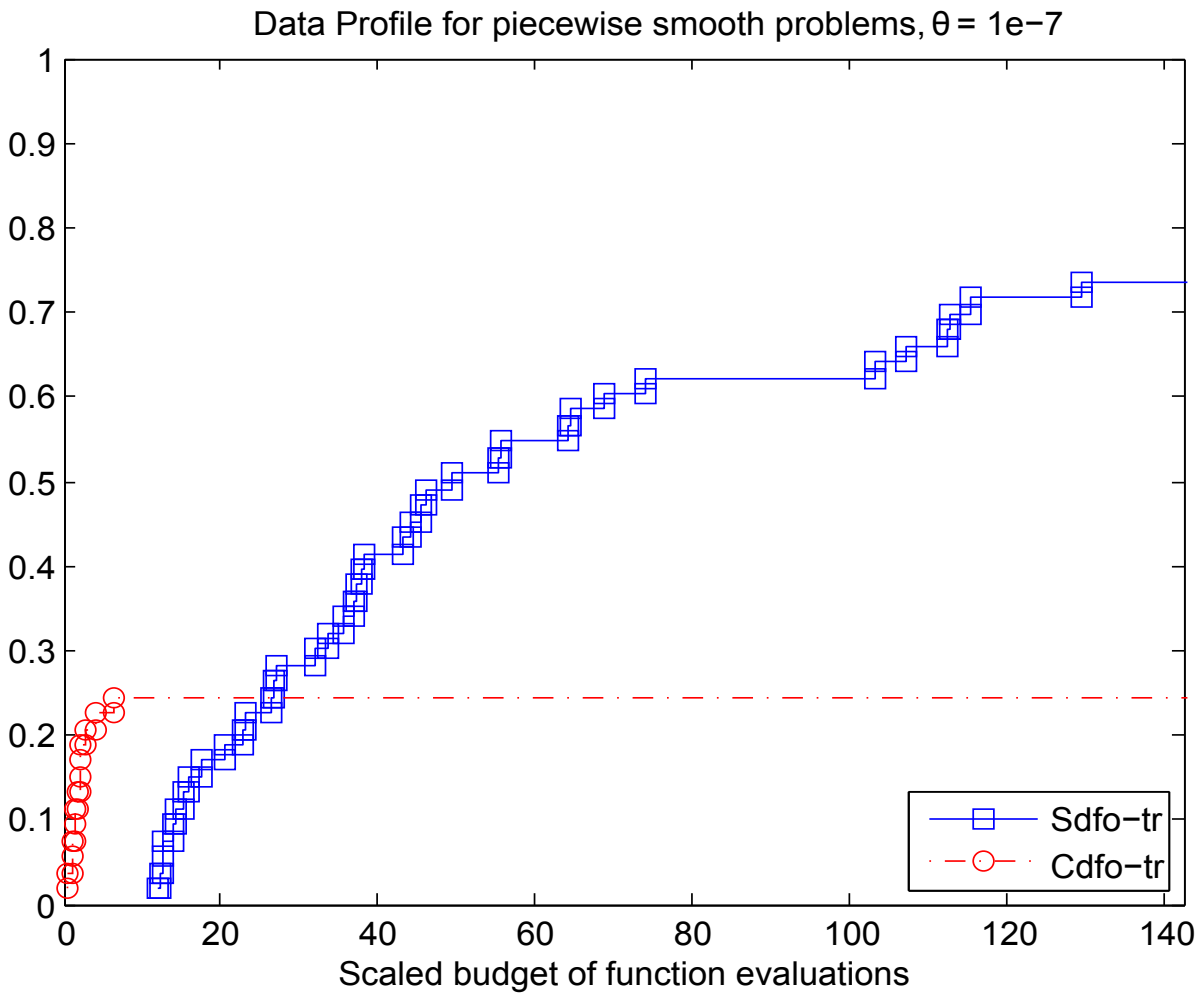
Results are shown using data profiles (1500 fevals).

The smoothing version Sdfo-tr is compared with smoothing Direct Search (Ssid-psm), Custódio and LNV (2009).

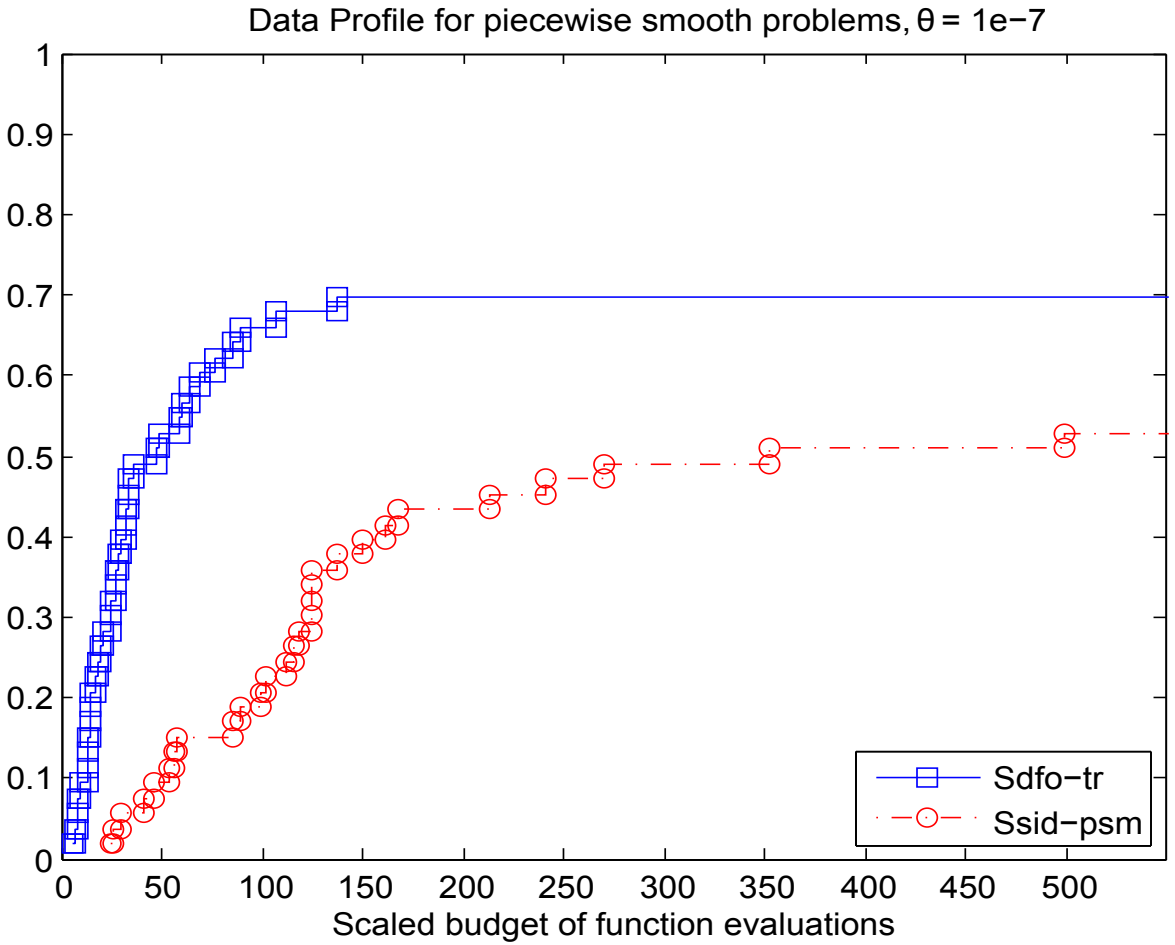
$\longrightarrow$ TR parameters are the same for both Sdfo-tr and Cdfo-tr.

$\longrightarrow$ Smoothing parameters are the same for both Sdfo-tr and Ssid-psm.

Data profiles (Sdfo-tr vs. Cdfo-tr)



Data profiles (Sdfo-tr vs. Ssid-psm)



Open issues

- A TR smoothing approach for the determination of second-order stationary points of non-smooth functions (global convergence and WCC).

Open issues

- A TR smoothing approach for the determination of second-order stationary points of non-smooth functions (global convergence and WCC).
- A general TR methodology & theory for the totally black-box non-smooth case.

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.

$\longrightarrow$ Hessian sparse but sparsity structure unknown.

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.
$\longrightarrow$ Hessian sparse but sparsity structure unknown.

Random models may give an advantage in a parallel environment without full synchronization.
$\longrightarrow$ Time to compute function values is random and a budget is imposed.

TR based on probabilistic models

Random models (random sample sets) may maintain a higher quality by using fewer sample points.
$\longrightarrow$ Hessian sparse but sparsity structure unknown.

Random models may give an advantage in a parallel environment without full synchronization.
$\longrightarrow$ Time to compute function values is random and a budget is imposed.

So, now, models are built iteratively in some random fashion.

$\longrightarrow$ M_k for random models, and $m_k = M_k(\omega_k)$ for their realizations.

The key assumption for convergence will be then that these models exhibit good accuracy with sufficiently high probability.

TR based on probabilistic models

Fix three positive parameters $\eta_1, \eta_2, \gamma, \beta$, with $\beta < 1 < \gamma$.

Algorithm (Iteration k)

Approximate the function f in $B(x_k; \delta_k)$ with m_k.

Compute a step s_k by solving $\min_{s \in B(0;\delta_k)} m_k(x_k + s)$.

Let

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2 \delta_k$, set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \gamma \delta_k$.

Otherwise, set $x_{k+1} = x_k$ and $\delta_{k+1} = \beta \delta_k$.

Probabilistic fully linear/quadratic models

Assumption

We say that a sequence of random models $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear for a corresponding sequence* $\{B(X_k, \Delta_k)\}$ *if the events*

$$S_k \;=\; \{M_k \text{ is a } (\kappa_{eg}, \kappa_{ef})\text{-fully linear model of } f \text{ on } B(X_k, \Delta_k)\}$$

satisfy the following submartingale-like condition

$$P(S_k | \sigma(M_0, \ldots, M_{k-1})) \;\geq\; p.$$

Probabilistic fully linear/quadratic models

Assumption

We say that a sequence of random models $\{M_k\}$ is p-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear for a corresponding sequence $\{B(X_k, \Delta_k)\}$ if the events

$$S_k \;=\; \{M_k \text{ is a } (\kappa_{eg}, \kappa_{ef})\text{-fully linear model of } f \text{ on } B(X_k, \Delta_k)\}$$

satisfy the following submartingale-like condition

$$P(S_k | \sigma(M_0, \ldots, M_{k-1})) \;\geq\; p.$$

p-probabilistically $(\kappa_{eh}, \kappa_{eg}, \kappa_{ef})$-fully quadratic are defined accordingly.

Getting ready for global convergence

Let Z_k be the indicator function of S_k.

Getting ready for global convergence

Let Z_k be the indicator function of S_k.

Let

$$p_0 = \frac{\ln \beta}{\ln(\gamma^{-1}\beta)} = \frac{1}{2} \quad \text{when} \quad \beta = 1/2,\ \gamma = 2.$$

Getting ready for global convergence

Let Z_k be the indicator function of S_k.

Let

$$p_0 \;=\; \frac{\ln \beta}{\ln(\gamma^{-1}\beta)} \;=\; \frac{1}{2} \quad \text{when} \quad \beta = 1/2,\ \gamma = 2.$$

If $\{M_k\}$ is p_0-probabilistically fully linear/quadratic, then (due to a submartingale argument)

$$P\left[\sum_{\ell=0}^{\infty} (Z_\ell - p_0) \;=\; -\infty\right] \;=\; 0.$$

Getting ready for global convergence

Let Z_k be the indicator function of S_k.

Let

$$p_0 \;=\; \frac{\ln \beta}{\ln(\gamma^{-1}\beta)} \;=\; \frac{1}{2} \quad \text{when} \quad \beta = 1/2,\ \gamma = 2.$$

If $\{M_k\}$ is p_0-probabilistically fully linear/quadratic, then (due to a submartingale argument)

$$P\left[\sum_{\ell=0}^{\infty} (Z_\ell - p_0) \;=\; -\infty\right] \;=\; 0.$$

This is a key observation for global convergence, established in:

- A. S. Bandeira, K. Scheinberg, and LNV, Convergence of trust-region methods based on probabilistic models, SIAM J. on Optimization, 24 (2014) 1238-1264.

Global convergence of TR based on probabilistic models

Theorem

Let $\{X_k\}$ be a sequence of random iterates generated by the algorithm.

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear for some $\kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1.$$

Global convergence of TR based on probabilistic models

Theorem

Let $\{X_k\}$ be a sequence of random iterates generated by the algorithm.

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear for some $\kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1.$$

Suppose that the model sequence $\{M_k\}$ is p_0-probabilistically $(\kappa_{eh}, \kappa_{eg}, \kappa_{ef})$-fully quadratic for some $\kappa_{eh}, \kappa_{eg}, \kappa_{ef} > 0$. Then,

$$P\left[\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}[\nabla^2 f(x_k)]\right\} = 0\right] = 1.$$

Global rate and WCC

Global rates and WCC bounds were developed for DS based on probabilistic descent in:

- S. Gratton, C. W. Royer, LNV, and Z. Zhang, Direct search based on probabilistic descent, SIAM J. on Optimization, 25 (2015) 1249-1716.

Global rate and WCC

Global rates and WCC bounds were developed for DS based on probabilistic descent in:

- S. Gratton, C. W. Royer, LNV, and Z. Zhang, Direct search based on probabilistic descent, SIAM J. on Optimization, 25 (2015) 1249-1716.

The same theory applies for TR methods based on probabilistic models.

Global rate: What is desirable?

For each realization of the TR method, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,

Global rate: What is desirable?

For each realization of the TR method, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Global rate: What is desirable?

For each realization of the TR method, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

Global rate: What is desirable?

For each realization of the TR method, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

We are interested in the probabilities

Global rate

$$P\big(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k})\big)$$

and

Global rate: What is desirable?

For each realization of the TR method, define

- $\tilde{g}_k$: the gradient with minimum norm among $\nabla f(x_0), \ldots, \nabla f(x_k)$,
- k_ϵ: the smallest integer such that $\|\nabla f(x_k)\| \leq \epsilon$.

Denote the corresponding random variables by $\tilde{G}_k$ and K_ϵ.

We are interested in the probabilities

Global rate

$$P\big(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k})\big)$$

and

Worst case complexity

$$P\big(K_\epsilon \leq \mathcal{O}(\epsilon^{-2})\big).$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \;\leq\; \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \;\leq\; \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

It then results,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \;\subset\; \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right] k\right\}.$$

Global rate: Counting descent

Let z_ℓ denote the realization of Z_ℓ $(\ell \geq 0)$.

Intuition: If $\|\tilde{g}_k\|$ is 'big', then $\sum_{\ell=0}^{k-1} z_\ell$ is probably 'small'.

In fact, one can prove

$$\sum_{\ell=0}^{k-1} z_\ell \;\leq\; \mathcal{O}\left(\frac{1}{\|\tilde{g}_k\|^2}\right) + p_0 k.$$

It then results,

$$\left\{\|\tilde{G}_k\| > \epsilon\right\} \;\subset\; \left\{\sum_{\ell=0}^{k-1} Z_\ell \leq \underbrace{\left[\mathcal{O}\left(\frac{1}{k\epsilon^2}\right) + p_0\right]}_{\lambda} k\right\}.$$

A universal result

Hence $P\big(\|\tilde{G}_k\| \leq \epsilon\big) \;=\; 1 - P\big(\|\tilde{G}_k\| > \epsilon\big) \;\geq\; 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)$.

A universal result

Hence $P\big(\|\tilde{G}_k\| \le \epsilon\big) \;=\; 1 - P\big(\|\tilde{G}_k\| > \epsilon\big) \;\ge\; 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \le \lambda\, k\right)$.

Denote

$$\pi_k(\lambda) = P\left(\sum_{\ell=0}^{k-1} Z_\ell \le \lambda\, k\right).$$

A universal result

Hence $P\big(\|\tilde{G}_k\| \leq \epsilon\big) \;=\; 1 - P\big(\|\tilde{G}_k\| > \epsilon\big) \;\geq\; 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right)$.

Denote

$$\pi_k(\lambda) = P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

If $\{M_k\}$ is a probabilistic model, then π_k obeys a Chernoff type bound:

Lemma

Suppose that $\{M_k\}$ is p-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear and $\lambda \in (0, p)$. Then

$$\pi_k(\lambda) \;\leq\; \exp\left[-\frac{(p-\lambda)^2}{2p} k\right].$$

Global rate

Now we plug the Chernoff type bound into

$$P\left(\|\tilde{G}_k\| \leq \epsilon\right) \;\geq\; 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Global rate

Now we plug the Chernoff type bound into

$$P\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Theorem

Suppose that $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear with* $p > p_0$ *and*

$$k \geq \mathcal{O}\left(\frac{1}{\epsilon^2}\right).$$

Global rate

Now we plug the Chernoff type bound into

$$P\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - P\left(\sum_{\ell=0}^{k-1} Z_\ell \leq \lambda\, k\right).$$

Theorem

Suppose that $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear with* $p > p_0$ *and*

$$k \geq \mathcal{O}\left(\frac{1}{\epsilon^2}\right).$$

Then

$$P\left(\|\tilde{G}_k\| \leq \epsilon\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

Global rate

Theorem

Suppose that $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear with* $p > p_0$*. Then*

$$P\left(\|\tilde{G}_k\| \leq \frac{1}{\sqrt{k}}\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

Global rate

Theorem

Suppose that $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear with* $p > p_0$*. Then*

$$P\left(\|\tilde{G}_k\| \leq \frac{1}{\sqrt{k}}\right) \geq 1 - \exp\left[-\mathcal{O}(k)\right].$$

$\longrightarrow$ $\mathcal{O}(1/\sqrt{k})$ decaying sublinear rate for gradient holds with overwhelmingly high probability, matching the deterministic case.

Worst case complexity

Since $P(K_\epsilon \leq k) = P(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem

Suppose that $\{M_k\}$ *is* p*-probabilistically* $(\kappa_{eg}, \kappa_{ef})$*-fully linear with* $p > p_0$. *Then*

$$P\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right],$$

Worst case complexity

Since $P(K_\epsilon \leq k) = P(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem

Suppose that $\{M_k\}$ is p-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear with $p > p_0$. Then

$$P\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right],$$

where the n comes, as before, from squaring $\kappa = \mathcal{O}(\sqrt{n}L_{\nabla f})$ in bound for FL models.

Worst case complexity

Since $P(K_\epsilon \leq k) = P(\|\tilde{G}_k\| \leq \epsilon)$, we also get:

Theorem

Suppose that $\{M_k\}$ is p-probabilistically $(\kappa_{eg}, \kappa_{ef})$-fully linear with $p > p_0$. Then

$$P\left(K_\epsilon \leq \left\lceil \mathcal{O}\left(n\epsilon^{-2}\right)\right\rceil\right) \geq 1 - \exp\left[-\mathcal{O}(\epsilon^{-2})\right],$$

where the n comes, as before, from squaring $\kappa = \mathcal{O}(\sqrt{n}L_{\nabla f})$ in bound for FL models.

$\longrightarrow \mathcal{O}(n\epsilon^{-2})$ complexity bound for # of iterations holds with overwhelmingly high probability, matching the deterministic case.

Open issues

- In DS based on probabilistic descent, the WCC bound for **function evaluations** is

$$\mathcal{O}(m\, n\epsilon^{-2}) \;=\; \mathcal{O}(n\epsilon^{-2}),$$

where $m > 1$ is the number of directions used (say $m = 2$).

Open issues

- In DS based on probabilistic descent, the WCC bound for **function evaluations** is

$$\mathcal{O}(m\, n\epsilon^{-2}) \;=\; \mathcal{O}(n\epsilon^{-2}),$$

where $m > 1$ is the number of directions used (say $m = 2$).

Can one do something similar (in the sense of getting $\mathcal{O}(n)$) for TR methods based on probabilistic models?

Open issues

- In DS based on probabilistic descent, the WCC bound for **function evaluations** is

$$\mathcal{O}(m\, n\epsilon^{-2}) \;=\; \mathcal{O}(n\epsilon^{-2}),$$

  where $m > 1$ is the number of directions used (say $m = 2$).

  Can one do something similar (in the sense of getting $\mathcal{O}(n)$) for TR methods based on probabilistic models?

- Particularization to the convex and strongly convex cases.

Open issues

- In DS based on probabilistic descent, the WCC bound for **function evaluations** is

$$\mathcal{O}(m\, n\epsilon^{-2}) \;=\; \mathcal{O}(n\epsilon^{-2}),$$

  where $m > 1$ is the number of directions used (say $m = 2$).

  Can one do something similar (in the sense of getting $\mathcal{O}(n)$) for TR methods based on probabilistic models?

- Particularization to the convex and strongly convex cases.

- Extension to constraints and non-smooth obj. functions.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

The number of observations in each Monte Carlo oracle may be up to $\mathcal{O}(\delta^{-4})$.

Stochastic optimization

What is observable is $\tilde{f}(x, \varepsilon(\omega))$, where ε is a random variable. The objective function $f(x)$ may be given by $E(\tilde{f}(x, \varepsilon))$.

One possible approach is to extend the **CSV framework** using SAA (Sample-Average Approximation).

First-order global convergence wp1 was derived in:

- ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization, S. Shashaani, F. S. Hashemi, and R. Pasuparhy, 2015.

The number of observations in each Monte Carlo oracle may be up to $\mathcal{O}(\delta^{-4})$.

The proof seems correct but... for algorithmic parameters that depend on unknown constants. To follow...

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

In the non biased case $f(x) = E(\tilde{f}(x, \varepsilon))$, the probabilistic assumptions can be ensured by SAA (with $\mathcal{O}(\delta^{-4})$ observations).

Stochastic optimization

Another avenue is to extend the **TR based on probabilistic models** to cover also probabilistic estimates of the obj. function:

- Stochastic optimization using a trust-region method and random models, R. Chen, M. Menickelly, and K. Scheinberg, 2015.

First-order global convergence wp1 has also been derived, but also for algorithmic parameters that depend on unknown constants. To follow...

In the non biased case $f(x) = E(\tilde{f}(x, \varepsilon))$, the probabilistic assumptions can be ensured by SAA (with $\mathcal{O}(\delta^{-4})$ observations).

This approach can handle biased cases like failures in function evaluations or even processor failures (thus accommodating gradient failures when using f.d.).