A Trust Region Method with a Worst-Case Iteration Complexity of $O(\epsilon^{-3/2})$ for Nonconvex Smooth Optimization

Frank E. Curtis, Lehigh University

joint work with

Daniel P. Robinson, Johns Hopkins University
Mohammadreza Samadi, Lehigh University

International Symposium on Mathematical Programming
Pittsburgh, PA, USA

14 July 2015
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
Unconstrained (nonconvex) optimization

Given $f : \mathbb{R}^n \to \mathbb{R}$, consider the unconstrained optimization problem

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

In this talk, we are primarily interested in

- solving nonconvex instances
- ...to find first- or second-order critical points;
- employing second-order methods;
- attaining global and fast local (i.e., quadratic) convergence;
- attaining good worst-case iteration (evaluation, etc.) complexity bounds.
Methods of interest in this talk

Trust region methods

▶ Decades of algorithmic development
▶ Levenberg (1944); Marquardt (1963); Powell (1970); many more!

Cubic regularization methods

▶ Relatively recent algorithmic development; fewer variants
▶ Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)
Methods of interest in this talk

Trust region methods
- Decades of algorithmic development
- Levenberg (1944); Marquardt (1963); Powell (1970); many more!

Cubic regularization methods
- Relatively recent algorithmic development; fewer variants
- Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)

Theoretical guarantees to assess a nonconvex optimization algorithm:
- **Global convergence**, i.e., \( \nabla f(x_k) \to 0 \) and maybe \( \min(\text{eig}(\nabla^2 f(x_k))) \to \zeta > 0 \)
- **Local convergence rate**, i.e., \( \|\nabla f(x_{k+1})\|_2 / \|\nabla f(x_k)\|_2 \to 0 \) (or more)
- **Worst-case complexity**, i.e., upper bound on number of iterations\(^1\) to achieve

\[
\|\nabla f(x_k)\|_2 \leq \epsilon \ \text{and perhaps} \ \min(\text{eig}(\nabla^2 f(x_k))) \geq -\epsilon \ \text{for some} \ \epsilon > 0
\]

\(^1\) or function evaluations, subproblem solves, etc.
Methods of interest in this talk

Trust region methods
- Decades of algorithmic development
- Levenberg (1944); Marquardt (1963); Powell (1970); many more!
- Global convergence, local quadratic rate when $\nabla^2 f(x^*) > 0$
- $O(\epsilon^{-2})$ complexity to first-order $\epsilon$-criticality, $O(\epsilon^{-3})$ to second-order

Cubic regularization methods
- Relatively recent algorithmic development; fewer variants
- Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)
- Global convergence, local quadratic rate when $\nabla^2 f(x^*) > 0$
- $O(\epsilon^{-3/2})$ complexity to first-order $\epsilon$-criticality, $O(\epsilon^{-3})$ to second-order

Theoretical guarantees to assess a nonconvex optimization algorithm:
- Global convergence, i.e., $\nabla f(x_k) \to 0$ and maybe $\min(\text{eig}(\nabla^2 f(x_k))) \to \zeta > 0$
- Local convergence rate, i.e., $\|\nabla f(x_{k+1})\|_2 / \|\nabla f(x_k)\|_2 \to 0$ (or more)
- Worst-case complexity, i.e., upper bound on number of iterations\(^1\) to achieve

$$\|\nabla f(x_k)\|_2 \leq \epsilon \text{ and perhaps } \min(\text{eig}(\nabla^2 f(x_k))) \geq -\epsilon \text{ for some } \epsilon > 0$$

\(^1\) or function evaluations, subproblem solves, etc.
Goals and contributions

What are our goals in this work?

▶ **Question**: Can we design a TR method with improved complexity?
▶ ...and does this lead to improved performance?

What are our contributions? A TR method that has

▶ global and quadratic local convergence rate guarantees;
▶ a worst-case iteration complexity of $O(\epsilon^{-3/2})$ to first-order $\epsilon$-criticality;
▶ ...and of $O(\epsilon^{-3})$ to second-order $\epsilon$-criticality.

How is this achieved?

▶ new step acceptance criteria;
▶ new mechanism for rejecting a step while expanding the TR radius;
▶ new updates that may involve sublinear TR radius decrease.
Goals and contributions

What are our goals in this work?

▶ **Question**: Can we design a TR method with improved complexity?
▶ ... and does this lead to improved performance?

What are our contributions? A TR method that has

▶ global and quadratic local convergence rate guarantees;
▶ a worst-case iteration complexity of $O(\epsilon^{-3/2})$ to first-order $\epsilon$-criticality;
▶ ... and of $O(\epsilon^{-3})$ to second-order $\epsilon$-criticality.

How is this achieved?

▶ new step acceptance criteria;
▶ new mechanism for rejecting a step while expanding the TR radius;
▶ new updates that may involve sublinear TR radius decrease.

We discuss three algorithms:

▶ **TTR**: “Traditional” Trust Region algorithm
▶ **ARC**: Adaptive Regularisation algorithm using Cubics
  ▶ Cartis, Gould, & Toint (2011)
▶ **TRACE**: Trust Region Algorithm with Contractions and Expansions
  ▶ Curtis, Robinson, & Samadi (2014)
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
## Algorithm basics

<table>
<thead>
<tr>
<th><strong>TTR</strong></th>
<th><strong>ARC</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Solve to compute $s_k$:</td>
<td>1: Solve to compute $s_k$:</td>
</tr>
<tr>
<td>$\min_{s \in \mathbb{R}^n} q_k(s)$</td>
<td>$\min_{s \in \mathbb{R}^n} c_k(s)$</td>
</tr>
<tr>
<td>$:= f_k + g_k^T s + \frac{1}{2} s^T H_k s$</td>
<td>$:= f_k + g_k^T s + \frac{1}{2} s^T H_k s$</td>
</tr>
<tr>
<td>s.t. $|s|_2 \leq \delta_k$ (dual: $\lambda_k$)</td>
<td>+ $\frac{1}{3} \sigma_k |s|_2^3$</td>
</tr>
<tr>
<td>2: Compute ratio:</td>
<td>2: Compute ratio:</td>
</tr>
<tr>
<td>$\rho_k^q \leftarrow \frac{f_k - f(x_k + s_k)}{f_k - q_k(s_k)}$</td>
<td>$\rho_k^c \leftarrow \frac{f_k - f(x_k + s_k)}{f_k - c_k(s_k)}$</td>
</tr>
<tr>
<td>3: Update radius:</td>
<td>3: Update regularization:</td>
</tr>
<tr>
<td>$\rho_k^q \geq \eta$: accept and $\delta_k \uparrow$</td>
<td>$\rho_k^c \geq \eta$: accept and $\sigma_k \downarrow$</td>
</tr>
<tr>
<td>$\rho_k^q &lt; \eta$: reject and $\delta_k \downarrow$</td>
<td>$\rho_k^c &lt; \eta$: reject and $\sigma_k \uparrow$</td>
</tr>
</tbody>
</table>
Algorithm basics: Subproblem solution correspondence

**TTR**

1: Solve to compute $s_k$:

$$\min_{s \in \mathbb{R}^n} q_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s$$

s.t. $\|s\|_2 \leq \delta_k$ (dual: $\lambda_k$)

2: Compute ratio:

$$\rho_k^q \leftarrow \frac{f_k - f(x_k + s_k)}{f_k - q_k(s_k)}$$

3: Update radius:

$\rho_k^q \geq \eta$: accept and $\delta_k \uparrow$

$\rho_k^q < \eta$: reject and $\delta_k \downarrow$

**ARC**

1: Solve to compute $s_k$:

$$\min_{s \in \mathbb{R}^n} c_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{3} \sigma_k \|s\|_2^3$$

2: Compute ratio:

$$\rho_k^c \leftarrow \frac{f_k - f(x_k + s_k)}{f_k - c_k(s_k)}$$

3: Update regularization:

$\rho_k^c \geq \eta$: accept and $\sigma_k \downarrow$

$\rho_k^c < \eta$: reject and $\sigma_k \uparrow$
Discussion

What are the similarities?

▶ algorithmic frameworks are almost identical
▶ one-to-one correspondence (except $\lambda_k = 0$) between subproblem solutions

What are the key differences?

▶ step acceptance criteria
▶ trust region vs. regularization coefficient updates
Discussion

What are the similarities?
- algorithmic frameworks are almost identical
- one-to-one correspondence (except $\lambda_k = 0$) between subproblem solutions

What are the key differences?
- step acceptance criteria
- trust region vs. regularization coefficient updates

Recall that a solution $s_k$ of the TR subproblem is also a solution of

$$\min_{s \in \mathbb{R}^n} f_k + g_k^T s + \frac{1}{2} s^T (H_k + \lambda_k I) s,$$

so the dual variable $\lambda_k$ can be viewed as a quadratic regularization coefficient.
Regularization/stepszie trade-off: TTR

At a given iterate $x_k$, curve illustrates dual variable (i.e., quadratic regularization) and norm of corresponding step as a function of TR radius.

\[ \| s_k(\delta) \|_2 (= \delta) \]

\[ \max\{0, -\min(\text{eig}(H_k))\} \]

\[ \lambda_k(\delta_k), \| s_k(\delta_k) \|_2 \]
Regularization/stepsizeshape trade-off: TTR

After a rejected step (i.e., with $x_{k+1} = x_k$) we set $\delta_{k+1} \leftarrow \gamma \delta_k$ (linear rate of decrease) while $\lambda_{k+1} > \lambda_k$
Regularization/steps-size trade-off: TTR

\[ \|s_k(\delta)\|_2 = \delta \]

In fact, the increase in the dual can be quite severe in some cases! (We have no direct control over this.)

\[ \max\{0, -\min(\text{eig}(H_k))\} \]

\[ (\lambda_k(\delta_k), \|s_k(\delta_k)\|_2) \]

\[ (\lambda_{k+1}(\delta_{k+1}), \|s_{k+1}(\delta_{k+1})\|_2) \]
Intuition, please!

Intuitively, what is so important about $\frac{\lambda_k}{\|s_k\|_2} = \frac{\lambda_k}{\delta_k}$?

- Large $\delta_k$ implies $s_k$ may not yield objective decrease.
- Small $\delta_k$ prohibits long steps.
- Small $\lambda_k$ suggests the TR is not restricting us too much.
- Large $\lambda_k$ suggests more objective decrease is possible.

So what is so bad (for complexity’s sake) with the following?

$$\frac{\lambda_k}{\delta_k} \approx 0 \text{ and } \frac{\lambda_{k+1}}{\delta_{k+1}} \gg 0.$$ 

It’s that we may go from a

- large, but unproductive step to a
- productive, but (too) short step!
ARC magic

So what’s the magic of ARC?

- It’s not the types of steps you compute (since TR subproblem gives the same).
- It’s that a simple update for $\sigma_k$ gives a good regularization/stepszie balance.
So what’s the magic of ARC?

▶ It’s not the types of steps you compute (since TR subproblem gives the same).
▶ It’s that a simple update for $\sigma_k$ gives a good regularization/stepsise balance.

In ARC, restricting $\sigma_k \geq \sigma_{\text{min}}$ for all $k$ and proving that $\sigma_k \leq \sigma_{\text{max}}$ for all $k$ ensures that all accepted steps satisfy

$$f_k - f_{k+1} \geq c_1 \sigma_{\text{min}} \|s_k\|_2^3 \quad \text{and} \quad \|s_k\|_2 \geq \left(\frac{c_2}{\sigma_{\text{max}} + c_3}\right)^2 \|g_{k+1}\|_2^{1/2}.$$

One can also show that, at any point, the number of rejected steps that can occur consecutively is bounded above by a constant (independent of $k$ and $\epsilon$).

▶ Important to note that ARC always has the regularization “on.”
Regularization/stepsizes trade-off: ARC

\[ \|s_k(\sigma)\|_2 \quad \text{slope} = \frac{1}{\sigma_k} \]

All points on the dashed line yield the same ratio \( \sigma = \lambda / \|s\|_2 \)
so, given \( \sigma_k \), the properties of \( s_k \) are determined by the intersection of the dashed line and the curve

\[ \max\{0, -\min(\text{eig}(H_k))\} \]

\[ \lambda_k(\sigma) \]
**Regularization/stepszie trade-off: ARC**

\[ \|s_k(\sigma)\|_2 \]

**slope = 1/\sigma_k**

A sequence of rejected steps follow the curve much differently than TTR;

\[ \max\{0, -\min(\text{eig}(H_k))\} \]

\[ \lambda_k(\sigma) \]
Regularization/stepsizes trade-off: ARC

\[ \|s_k(\sigma)\|_2 \]

A sequence of rejected steps follow the curve much differently than TTR; in particular, for sufficiently large \( \sigma \), the rate of decrease in \( \|s\| \) is sublinear.

\[ \max\{0, -\min(\text{eig}(H_k))\} \]

\[ \lambda_k(\sigma) \]
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
From TTR to TRACE

TRACE involves three key modifications of TTR.

1: Different step acceptance ratio
2: New expansion step: May reject step while increasing TR radius
3: New contraction procedure: Explicit or implicit (through update of \( \lambda \))
Step acceptance ratio

1: Different step acceptance ratio

\[
TTR: \; \rho_k^q = \frac{f_k - f(x_k + s_k)}{f_k - q_k(s_k)} \quad \Rightarrow \quad \text{TRACE:} \; \rho_k = \frac{f_k - f(x_k + s_k)}{\|s_k\|^3_2}
\]

Motivations:
- With second-order model, error is third-order.
- Recall the first guarantee of accepted steps in ARC:
  \[
f_k - f_{k+1} \geq c_1 \sigma_{\text{min}} \|s_k\|^3_2.
\]
Expansion steps

2: New expansion step: May reject step while increasing TR radius

- We define a monotonically increasing sequence \( \{\sigma_k\} \).
- (Plays a similar theoretical role as the regularization coefficients in ARC.)
- If objective decrease is good, but dual suggests more decrease is possible, i.e.,

\[ \rho_k \geq \eta \quad \text{but} \quad \lambda_k > \sigma_k \| s_k \|_2, \]

then reject the step and increase the TR radius to allow more decrease.
- With \( \delta_{k+1} \leftarrow \lambda_k / \sigma_k \), need at most one expansion between accepted steps.
Regularization/stepsziue trade-off: “Off” to “on”

How to go from “off” to “on” in terms of regularization?
Easy to undershoot or overshoot!
(Recall that, in ARC, regularization is never “off.”)
**Contraction steps**

3: New contraction procedure: Explicit or implicit (through update of $\lambda$)

\[ \lambda_k < \sigma \| s_k \|_2 \]

- set $\lambda_{k+1} \leftarrow \lambda_k + (\sigma \| g_k \|_2)^{1/2}$, or
- set $\lambda_{k+1} \in (\lambda_k, \lambda_k + (\sigma \| g_k \|_2)^{1/2})$ so $\sigma \leq \lambda_{k+1}/\| s_{k+1} \|_2 \leq \bar{\sigma}$

\[ \lambda_k \geq \sigma \| s_k \|_2 \]

- set $\lambda_{k+1} \leftarrow \gamma \lambda_k$ (with $\gamma > 1$), or
- set $\delta_{k+1} \leftarrow \gamma_c \delta_k$ (with $\gamma_c \in (0, 1)$)

Update based on dual variable only requires a linear system solve!

\[ (H_{k+1} + \lambda_{k+1} I) s = -g_{k+1} \]
Main algorithm

**Algorithm 1** Trust Region Algorithm with Contraction and Expansion (TRACE)

**Require:** an acceptance constant $\eta \in \mathbb{R}^{++}$ with $0 < \eta < 1/2$

**Require:** update constants $\{\gamma_c, \gamma_e, \gamma_\lambda\} \subset \mathbb{R}^{++}$ with $0 < \gamma_c < 1 < \gamma_e$ and $\gamma_\lambda > 1$

**Require:** bound constants $\{\sigma, \bar{\sigma}\} \subset \mathbb{R}^{++}$ with $0 < \sigma \leq \sigma_0 \leq \bar{\sigma}$

1: procedure TRACE
2: choose $x_0 \in \mathbb{R}^n$, $\{\delta_0, \Delta_0\} \subset \mathbb{R}^{++}$ with $\delta_0 \leq \Delta_0$, and $\sigma_0 \in \mathbb{R}^{++}$ with $\sigma_0 \geq \sigma$
3: compute $(s_0, \lambda_0)$ by TR subproblem, then compute $\rho_0$
4: for $k = 0, 1, 2, \ldots$ do
5: \hspace{1em} if $\rho_k \geq \eta$ and either $\lambda_k \leq \sigma_k \|s_k\|_2$ or $\|s_k\|_2 = \Delta_k$ then
6: \hspace{2em} set $x_{k+1} \leftarrow x_k + s_k$
7: \hspace{2em} set $\Delta_{k+1} \leftarrow \max\{\Delta_k, \gamma_e \|s_k\|_2\}$
8: \hspace{2em} set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \max\{\delta_k, \gamma_e \|s_k\|_2\}\}$
9: \hspace{2em} set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_k / \|s_k\|_2\}$
10: else if $\rho_k < \eta$ then
11: \hspace{2em} set $x_{k+1} \leftarrow x_k$
12: \hspace{2em} set $\Delta_{k+1} \leftarrow \Delta_k$
13: \hspace{2em} set $\delta_{k+1} \leftarrow \text{contract}(x_k, \delta_k, \sigma_k, s_k, \lambda_k)$
14: else (i.e., if $\rho_k \geq \eta$, $\lambda_k > \sigma_k \|s_k\|_2$, and $\|s_k\|_2 < \Delta_k$)
15: \hspace{2em} set $x_{k+1} \leftarrow x_k$
16: \hspace{2em} set $\Delta_{k+1} \leftarrow \Delta_k$
17: \hspace{2em} set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \lambda_k / \sigma_k\}$
18: \hspace{2em} set $\sigma_{k+1} \leftarrow \sigma_k$
19: compute $(s_{k+1}, \lambda_{k+1})$ by TR subproblem, then compute $\rho_{k+1}$
20: if $\rho_k < \eta$ then
21: \hspace{2em} set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_{k+1} / \|s_{k+1}\|_2\}$
Contraction subroutine

Algorithm 2 Trust Region Contraction Subroutine

1: procedure CONTRACT($x_k, \delta_k, \sigma_k, s_k, \lambda_k$)
2:     if $\lambda_k < \sigma \|s_k\|_2$ then
3:         set $\lambda \leftarrow \lambda_k + (\sigma \|g_k\|_2)^{1/2}$
4:         set $s$ as the solution of $(H_k + \lambda I)s = -g_k$
5:         set $\delta \leftarrow \|s\|_2$
6:         if $\lambda/\delta \leq \bar{\sigma}$ then
7:             return $\delta_{k+1} \leftarrow \delta$
8:         else
9:             compute $\hat{\lambda} \in (\lambda_k, \lambda)$ so $(H_k + \hat{\lambda} I)\hat{s} = -g_k$ yields $\sigma \leq \hat{\lambda}/\|\hat{s}\|_2 \leq \bar{\sigma}$
10:            set $\hat{\delta} \leftarrow \|\hat{s}\|_2$
11:            return $\delta_{k+1} \leftarrow \hat{\delta}$
12:     else (i.e., if $\lambda_k \geq \sigma \|s_k\|_2$)
13:         set $\lambda \leftarrow \gamma \lambda_k$
14:         set $s$ as the solution of $(H_k + \lambda I)s = -g_k$
15:         set $\delta \leftarrow \|s\|_2$
16:         if $\delta \geq \gamma_c \|s_k\|_2$ then
17:             return $\delta_{k+1} \leftarrow \delta$
18:         else
19:             return $\delta_{k+1} \leftarrow \gamma_c \|s_k\|_2$
Global and local quadratic convergence

**Assumption 1**
- \( f \) twice continuously differentiable and bounded below by \( f_{\text{min}} \)
- \( g \) Lipschitz continuous in open convex set containing \( \{x_k\} \) and \( \{x_k + s_k\} \)
- \( \{g_k\} \) has nonzero elements and bounded above
- \( \{H_k\} \) bounded above

**Theorem 2**
\[
\|g_k\|_2 \to 0
\]

**Assumption 3 (in addition to Assumption 1)**
\( \{x_k\} \xrightarrow{S} x^* \) around which \( H \) is positive definite and locally Lipschitz

**Theorem 4**
\( \{x_k\} \to x^* \) with \( g(x^*) = 0 \) and, for sufficiently large \( k \),
\[
\|g_{k+1}\|_2 = O(\|g_k\|_2^2) \quad \text{and} \quad \|x_{k+1} - x^*\|_2 = O(\|x_k - x^*\|_2^2)
\]
Worst-case iteration complexity to first-order $\epsilon$-criticality

Assumption 5 (in addition to Assumption 1)

$H$ Lipschitz continuous in open convex set containing $\{x_k\}$ and $\{x_k + s_k\}$

Lemma 6

- $f_k - f_{k+1} \geq \eta\|s_k\|_2^3$ for all accepted steps
- $\{\sigma_k\}$ bounded by $\sigma_{\text{max}} > 0$
- $\|s_k\|_2 \geq (H_{\text{Lip}} + \sigma_{\text{max}})^{-1/2}\|g_{k+1}\|_2^{1/2}$

Theorem 7

Total number of iterations with $\|g_k\|_2 > \epsilon$ is

$$O\left(\left[\frac{f_0 - f_{\text{min}}}{\eta \Delta_0^3}\right] + \left[\left(\frac{f_0 - f_{\text{min}}}{\eta (H_{\text{Lip}} + \sigma_{\text{max}})^{-3/2}}\right)^{\epsilon^{-3/2}}\right]\right)$$
Worst-case iteration complexity to second-order $\epsilon$-criticality

Under the same assumptions...

**Lemma 8**

$$\liminf_{k \to \infty} \min(eig(H_k)) \geq 0$$

**Theorem 9**

*Total number of iterations with*

$$\|g_k\|_2 > \epsilon \text{ or } \min(eig(H_k)) < -\epsilon$$

*is*

$$\mathcal{O} \left( \left[ \frac{f_0 - f_{\min}}{\eta \Delta_0^3} \right] + \left[ \frac{f_0 - f_{\min}}{\eta (H_{\text{Lip}} + \sigma_{\text{max}})^{-3/2}} \right] \epsilon^{-3/2} \right) + \mathcal{O} \left( \left[ \frac{f_0 - f_{\min}}{\eta \sigma_{\text{max}}^{-3}} \right] \epsilon^{-3} \right)$$
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
Implementation

Implemented TRACE, TTR, and ARC together in MATLAB

- “Same” subproblem solver for all algorithms; Conn, Gould, Toint (2000)
- MATLAB’s eigs for leftmost eigenvalues
- Radius and regularization updates:

\[
\begin{align*}
\text{(TTR) } \delta_{k+1} & \leftarrow \begin{cases}
\max\{\delta_k, 2\|s_k\|_2\} & \text{if } \rho_{qk}^q \geq \eta_2 \\
\delta_k & \text{if } \rho_{qk}^q \in [\eta_1, \eta_2) \\
\delta_k/2 & \text{if } \rho_{qk}^q < \eta_1
\end{cases} \\
\text{(ARC) } \sigma_{k+1} & \leftarrow \begin{cases}
\sigma_k/2 & \text{if } \rho_{ck}^c \geq \eta_2 \\
\sigma_k & \text{if } \rho_{ck}^c \in [\eta_1, \eta_2) \\
2\sigma_k & \text{if } \rho_{ck}^c < \eta_1
\end{cases}
\end{align*}
\]

- Termination criterion:

\[\|g_k\|_\infty \leq 10^{-6} \cdot \max\{\|g_0\|_\infty, 1\}\]
TRACE implementation details

- Reduction ratio:
  \[ \rho_k = \frac{f_k - f(x_k + s_k)}{\min\{\|s_k\|^3_2, f_k - c_k(s_k; \sigma)\}} \]

- Radius and regularization updates:
  \[ (\text{TRACE}) \quad \delta_{k+1} \left\{ \begin{array}{ll}
  \max\{\delta_k, 2\|s_k\|_2\} & \text{if } \rho_k \geq \eta_2 \\
  \delta_k & \text{if } \rho_k \in [\eta_1, \eta_2) \\
  \text{CONTRACT} & \text{if } \rho_k < \eta_1
  \end{array} \right. \]

  where CONTRACT uses
  \[ \sigma = 10^{-10}, \quad \overline{\sigma} = 10^{10}, \quad \gamma_\lambda = 2, \quad \gamma_c = 10^{-2} \]
Test set

Unconstrained problems from the CUTEr collection

- Removed 9 problems due to memory or decoding errors
- Removed 21 problems on which all algorithms failed
- Remaining set includes 130 problems

Step types taken (normalized by iterations per problem):

<table>
<thead>
<tr>
<th></th>
<th>Accepted</th>
<th>Contraction</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>63.73%</td>
<td>35.26%</td>
<td>1.01%</td>
</tr>
</tbody>
</table>

Contraction types taken (normalized by contractions per problem):

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_k + (\sigma |g_k|_2)^{1/2}$</th>
<th>$\sigma \leq \lambda / |s|_2 \leq \bar{\sigma}$</th>
<th>$\gamma \lambda \lambda_k$</th>
<th>$\delta \leftarrow \gamma_c |s_k|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.70%</td>
<td>0.00%</td>
<td>88.09%</td>
<td>9.21%</td>
</tr>
</tbody>
</table>
Performance profiles: Iterations
Performance profiles: Function evaluations
Performance profiles: Gradient evaluations
Performance profiles: Matrix factorizations

A Trust Region Method with Complexity $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Smooth Optimization
Outline

Motivation

TTR and ARC

TRACE

Numerical Experiments

Summary
Contributions

**Question:** Can we design a TR method with improved complexity?

- Yes, TRACE achieves the same convergence/complexity guarantees as ARC
- New step acceptance criteria
- New mechanism for **rejecting a step while expanding the TR radius**
- New updates that may involve **sublinear TR radius decrease**

Numerical experiments show algorithm is at least competitive with TTR and ARC

Future work

Next questions: Does TRACE offer new insights for improved performance?
  ▶ Competitive performance is not surprising, but can it be better?
  ▶ Note that an iteration of TRACE may only need a linear system solve!
  ▶ One may imagine algorithms like TRACE and ARC that achieve the same convergence/complexity guarantees and never fully solve a subproblem
  ▶ ... worst-case (approximate) linear system solve complexity?
  ▶ Does TRACE offer new insights for constrained optimization?