Optimal Newton-type algorithms for nonconvex smooth optimization

Coralia Cartis
Mathematical Institute, University of Oxford

joint with
Nick Gould (RAL, UK) & Philippe Toint (Namur, Belgium)

EPSRC NA-HPC Workshop on New Directions in Nonlinear Optimization
University of Manchester, School of Mathematics, March 19, 2014
Unconstrained optimization — a “mature” area?

Nonconvex unconstrained optimization:

\[
\min_{x \in \mathbb{R}^n} f(x) \quad \text{where} \quad f \in C^1(\mathbb{R}^n) \text{ or } C^2(\mathbb{R}^n).
\]

Currently two main competing methodologies:

- **Linesearch** methods
- **Trust-region** methods

Much reliable, efficient software for (large-scale) problems.

- Is there anything more to say?...
Unconstrained optimization — a “mature” area?

Nonconvex unconstrained optimization:

\[
\text{minimize } f(x) \text{ where } f \in C^1(\mathbb{R}^n) \text{ or } C^2(\mathbb{R}^n).
\]

Currently two main competing methodologies:

- **Linesearch** methods
- **Trust-region** methods

Much reliable, efficient software for (large-scale) problems.

- Is there anything more to say?...

  - **Global rates of convergence** of optimization algorithms

  \[ \iff \text{Evaluation complexity} \text{ of methods (from any initial guess)} \]

  [well-studied for convex problems, but unprecedented for nonconvex until recently]
Evaluation complexity of unconstrained optimization

Relevant analyses of iterative optimization algorithms:

- **Global convergence** to first/second-order critical points (from any initial guess)
- **Local convergence** and **local rates** (sufficiently close initial guess, well-behaved minimizer)
  [Newton’s method: Q-quadratic; steepest descent: linear]
- **Global rates** of convergence (from any initial guess)
  ⇐⇒ **Worst-case function evaluation complexity**
  - evaluations are often expensive in practice (climate modelling, molecular simulations, etc)
  - black-box/oracle computational model (suitable for the different ‘shapes and sizes’ of nonlinear problems)
  [Nemirovskii & Yudin ('83); Vavasis ('92), Sikorski ('01), Nesterov ('04)]
Overview

- Evaluation complexity of standard methods
- Improved complexity for cubic regularization
- Optimality of cubic regularization
- Evaluation complexity of constrained optimization
- ‘Cubic regularization’ for global optimization
Global efficiency of steepest-descent methods

Steepest descent method (with linesearch or trust-region):

1. $f \in C^1(\mathbb{R}^n)$ with Lipschitz continuous gradient.
2. To generate gradient $\|g(x_k)\| \leq \epsilon$, requires at most
   \[ \lceil \kappa_{sd} \cdot \text{Lips}_g \cdot (f(x_0) - f_{\text{low}}) \cdot \epsilon^{-2} \rceil \] function evaluations.

[Nesterov ('04); Gratton, Sartenaer & Toint ('08)]
Global efficiency of steepest-descent methods

Steepest descent method (with linesearch or trust-region):

- $f \in C^1(\mathbb{R}^n)$ with Lipschitz continuous gradient.
- to generate gradient $\|g(x_k)\| \leq \epsilon$, requires at most
  \[\lceil \kappa_{sd} \cdot \text{Lips}_g \cdot (f(x_0) - f_{\text{low}}) \cdot \epsilon^{-2} \rceil\] function evaluations.

The worst-case bound is sharp for steepest descent: [CGT(‘10)]

For any $\epsilon > 0$ and $\tau > 0$, (inexact-linesearch) steepest descent applied to this $f$ takes precisely

\[\lceil \epsilon^{-2+\tau} \rceil\] function evaluations

to generate $|g(x_k)| \leq \epsilon$. 
Worst-case bound is sharp for steepest descent

Steepest descent method with exact linesearch

\[ x_{k+1} = x_k - \alpha_k g(x_k) \text{ with } \alpha_k = \arg \min_{\alpha \geq 0} f(x_k - \alpha g(x_k)) \]

- takes \( \lceil \epsilon^{-2+\tau} \rceil \) iterations to generate \( \|g(x_k)\| \leq \epsilon \)

Contour lines of \( f(x_1, x_2) \) and path of iterates.
Key ideas in the construction of examples

- If \( \|g(x_k)\| \overset{\text{def}}{=} \|g_k\| \geq \left( \frac{1}{k+1} \right)^{\frac{1}{2}} \), then \( \|g_k\| \leq \epsilon \) when \( k \geq \epsilon^{-2} \).
Key ideas in the construction of examples

- If $\|g(x_k)\| \overset{\text{def}}{=} \|g_k\| \geq \left( \frac{1}{k+1} \right)^{\frac{1}{2}}$, then $\|g_k\| \leq \epsilon$ when $k \geq \epsilon^{-2}$.

- Define the sequences (for the slow component)

$$g_k = - \left( \frac{1}{k+1} \right)^{\frac{1}{2}+\eta} \quad \text{and} \quad H_k = 1$$

$$x_0 = 0 \quad \text{and} \quad x_{k+1} = x_k - \alpha_k g_k$$

$$f_{k+1} = \text{Qmodel}_k(x_{k+1}) \quad \text{and so} \quad f_0 = \frac{1}{2} \zeta (1 + 2\eta).$$
Key ideas in the construction of examples

- If \( \| g(x_k) \| \overset{\text{def}}{=} \| g_k \| \geq \left( \frac{1}{k+1} \right)^{\frac{1}{2}} \), then \( \| g_k \| \leq \epsilon \) when \( k \geq \epsilon^{-2} \).

- Define the sequences (for the slow component)

\[
g_k = - \left( \frac{1}{k+1} \right)^{\frac{1}{2}+\eta} \quad \text{and} \quad H_k = 1
\]

\[
x_0 = 0 \quad \text{and} \quad x_{k+1} = x_k - \alpha_k g_k
\]

\[
f_{k+1} = \text{Qmodel}_k(x_{k+1}) \quad \text{and so} \quad f_0 = \frac{1}{2} \zeta (1 + 2\eta).
\]

- Use Hermite interpolation on \([x_k, x_{k+1}]\) to construct \( f \) s.t.

\[
f(x_k) = f_k, \quad g(x_k) = g_k \quad \text{and} \quad H(x_k) = H_k.
\]
Global efficiency of Newton’s method

- Newton’s method: \( x_{k+1} = x_k - H_k^{-1} g_k \) with \( H_k > 0 \).

Newton’s method: as slow as steepest descent

- may require \( \lceil \epsilon^{-2+\tau} \rceil \) evaluations/iterations, same as steepest descent method

Globally Lipschitz continuous gradient and Hessian
Worst-case bound for Newton’s method

- when globalized with trust-region or linesearch, Newton’s method will take at most
  \[\lceil \kappa_N \epsilon^{-2} \rceil\]
evaluations to generate \(\|g_k\| \leq \epsilon\)

- similar worst-case complexity for classical trust-region and linesearch methods

Is there any method with better evaluation complexity than steepest-descent?
Improved complexity for cubic regularization
Improved complexity for cubic regularization

A cubic model: [Griewank ('81, TR), Nesterov & Polyak ('06), Weiser et al ('07)]

\( H \) is globally Lipschitz continuous with Lipschitz constant \( 2\sigma \):

Taylor, Cauchy-Schwarz and Lipschitz \( \Rightarrow \)

\[
f(x_k + s) \leq f(x_k) + s^T g(x_k) + \frac{1}{2} s^T H(x_k) s + \frac{1}{3} \sigma \| s \|^3_2 + m_k(s)
\]

\( \Rightarrow \) reducing \( m_k \) from \( s = 0 \) decreases \( f \) since \( m_k(0) = f(x_k) \).

Cubic regularization method: [Nesterov & Polyak ('06)]

- \( x_{k+1} = x_k + s_k \)
- compute \( s_k \rightarrow \min_s m_k(s) \) globally: [possible, even if \( m_k \) nonconvex!]
Improved complexity for cubic regularization

A cubic model: [Griewank ('81, TR), Nesterov & Polyak ('06), Weiser et al ('07)]

\( H \) is globally Lipschitz continuous with Lipschitz constant \( 2\sigma \):
Taylor, Cauchy-Schwarz and Lipschitz \( \implies \)

\[
f(x_k + s) \leq f(x_k) + s^T g(x_k) + \frac{1}{2} s^T H(x_k) s + \frac{3}{2} \sigma \|s\|^3
\]

\( m_k(s) \)

\( \implies \) reducing \( m_k \) from \( s = 0 \) decreases \( f \) since \( m_k(0) = f(x_k) \).

Cubic regularization method: [Nesterov & Polyak ('06)]

\[ x_{k+1} = x_k + s_k \]

compute \( s_k \rightarrow \min_s m_k(s) \) globally: [possible, even if \( m_k \) nonconvex!]

Worst-case evaluation complexity: at most \( \lceil \kappa_{cr} \cdot \epsilon^{-3/2} \rceil \)
function evaluations to ensure \( \|g(x_k)\| \leq \epsilon \). [Nesterov & Polyak ('06)]

Can we make cubic regularization computationally efficient?
Minimizing the cubic model

- $f$ nonconvex $\rightarrow m_k(s)$ may be nonconvex!

$$m(s) \equiv f + s^T g + \frac{1}{2} s^T H s + \frac{1}{3} \sigma \|s\|^3$$

**Necessary and sufficient optimality:** any **global** minimizer $s_*$ of $m$ satisfies $(H + \lambda_* I)s_* = -g$ and $\lambda_* = \sigma \|s_*\|$, 

- $H + \lambda_* I$ is positive semidefinite

[Nesterov & Polyak ('06); C, Gould, & Toint ('11)]
Adaptive cubic regularization – a practical method

Use


- cubic regularization model at $x_k$

\[ m_k(s) \equiv f(x_k) + s^T g(x_k) + \frac{1}{2} s^T B_k s + \frac{1}{3} \sigma_k \|s\|^3 \]

- $\sigma_k > 0$ is the iteration-dependent regularization weight
- $B_k$ is an approximate Hessian
Adaptive cubic regularization – a practical method

Use

- cubic regularization model at $x_k$

$$m_k(s) \equiv f(x_k) + s^T g(x_k) + \frac{1}{2} s^T B_k s + \frac{1}{3} \sigma_k \|s\|^3$$

- $\sigma_k > 0$ is the iteration-dependent regularization weight
- $B_k$ is an approximate Hessian

- compute $s_k \approx \arg \min_s m_k(s)$ [details to follow]

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

- set $x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > 0.1 \\ x_k & \text{otherwise} \end{cases}$

- $\sigma_{k+1} = 2\sigma_k$ whenever $\rho_k < 0.1$
**Adaptive Regularization with Cubics (ARC)**

**ARC:** $s_k =$ global min of $m_k(s)$ over $s \in S \subseteq \mathbb{R}^n$, with $g \in S$

$\rightarrow$ increase subspaces to satisfy termination criteria:

$$\|\nabla_s m_k(s_k)\| \leq \min(1, \|s_k\|)\|g_k\|$$
Adaptive Regularization with Cubics (ARC)

**ARC**: $s_k$ = global min of $m_k(s)$ over $s \in \mathbb{S} \leq \mathbb{R}^n$, with $g \in \mathbb{S}$

→ increase subspaces to satisfy termination criteria:

$$\|\nabla_s m_k(s_k)\| \leq \min(1, \|s_k\|)\|g_k\|$$

**ARC** has excellent convergence properties: globally, to second-order critical points and locally, Q-quadratically.
Adaptive Regularization with Cubics (ARC)

ARC: \( s_k = \text{global min of } m_k(s) \) over \( s \in S \subseteq \mathbb{R}^n \), with \( g \in S \)

\[ \rightarrow \text{increase subspaces to satisfy termination criteria:} \]

\[ \| \nabla_s m_k(s_k) \| \leq \min(1, \| s_k \|) \| g_k \| \]

ARC has excellent convergence properties: globally, to second-order critical points and locally, Q-quadratically.

‘Average-case’ performance of ARC variants (preliminary numerics)
Worst-case performance of ARC

If $g$ and $H$ are Lipschitz continuous on iterates’ path and $\| (B_k - H_k) s_k \| = O(\|s_k\|^2)^{(\ast)}$, then ARC requires at most

$$\left[ \kappa_{arc} \cdot (L_g L_H)^{\frac{3}{2}} \cdot (f(x_0) - f_{low}) \cdot \epsilon^{-\frac{3}{2}} \right]$$

function evaluations to ensure $\|g_k\| \leq \epsilon$. [cf. Nesterov & Polyak]

(\ast) achievable when $B_k = H_k$ or when $B_k$ is computed by gradient finite differences
Worst-case performance of ARC

If $g$ and $H$ are Lipschitz continuous on iterates’ path and $\|(B_k - H_k)s_k\| = O(\|s_k\|^2)(*)$, then ARC requires at most

$$\left[ \kappa_{arc} \cdot (L_g L_H)^{\frac{3}{2}} \cdot (f(x_0) - f_{low}) \cdot \epsilon^{-\frac{3}{2}} \right]$$

function evaluations to ensure $\|g_k\| \leq \epsilon$. [cf. Nesterov & Polyak]

(*) achievable when $B_k = H_k$ or when $B_k$ is computed by gradient finite differences

Key ingredients:

- **sufficient function decrease:** $f(x_k) - f(x_{k+1}) \geq \frac{m}{6} \sigma_k \|s_k\|^3$
- **long successful steps:** $\|s_k\| \geq C\|g_{k+1}\|^{\frac{1}{2}}$ (and $\sigma_k \geq \sigma_{\text{min}} > 0$)
Worst-case performance of ARC

If $g$ and $H$ are Lipschitz continuous on iterates’ path and $\|(B_k - H_k)s_k\| = O(\|s_k\|^2)$\textsuperscript{(*)}, then ARC requires at most

$$\left[ \kappa_{arc} \cdot (L_g L_H)^{\frac{3}{2}} \cdot (f(x_0) - f_{low}) \cdot \epsilon^{-\frac{3}{2}} \right] \text{ function evaluations}$$

to ensure $\|g_k\| \leq \epsilon$. \textsuperscript{[cf. Nesterov & Polyak]}

\textsuperscript{(*)} achievable when $B_k = H_k$ or when $B_k$ is computed by gradient finite differences

Key ingredients:

- **sufficient function decrease:** $f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{6} \sigma_k \|s_k\|^3$

- **long successful steps:** $\|s_k\| \geq C \|g_{k+1}\|^{\frac{1}{2}}$ (and $\sigma_k \geq \sigma_{\min} > 0$)

$\implies$ while $\|g_k\| \geq \epsilon$ and $k$ successful,

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{6} \sigma_{\min} C \cdot \epsilon^{\frac{3}{2}}$$

summing up over $k$ successful:

$$f(x_0) - f_{low} \geq k_{S} \frac{\eta_1 \sigma_{\min} C}{6} \epsilon^{\frac{3}{2}}$$
Cubic regularization: worst-case bound is sharp

For any $\epsilon > 0$ and $\tau > 0$, cubic regularization/ARC applied to this $f$ takes precisely

$$\left\lceil \epsilon^{-\frac{3}{2}} + \tau \right\rceil$$

function evaluations
to generate $|g(x_k)| \leq \epsilon$.

$$g_k = -\left(\frac{1}{k+1}\right)^{\frac{2}{3} + \eta}, \quad H_k = 0, \quad \sigma_k = \text{Lips}_H$$
Optimality of cubic regularization
A general class of methods and objectives

Class of methods $M.\alpha$: $x_{k+1} = x_k + s_k, \ k \geq 0$;

- $(H_k + \lambda_k I)s_k = -g_k$ with $\lambda_k \geq 0$ and $H_k + \lambda_k I \succeq 0$
- $\|s_k\| \leq \kappa_s$ and $\lambda_k \leq \kappa_\lambda \|s_k\|^\alpha$, for some $\alpha \in [0, 1]$. 
A general class of methods and objectives

Class of methods $M.\alpha$: $x_{k+1} = x_k + s_k, \quad k \geq 0$;

- $(H_k + \lambda_k I)s_k = -g_k$ with $\lambda_k \geq 0$ and $H_k + \lambda_k I \succeq 0$
- $\|s_k\| \leq \kappa_s$ and $\lambda_k \leq \kappa_\lambda \|s_k\|^\alpha$, for some $\alpha \in [0, 1]$.

Class of objectives $A.\alpha$: $f \in C^2$ bounded below;
$g$ globally Lipschitz continuous and $H \alpha$-Hölder continuous on the path of the iterates.
A general class of methods and objectives

Class of methods $M.\alpha$: $x_{k+1} = x_k + s_k$, $k \geq 0$;

- $(H_k + \lambda_k I)s_k = -g_k$ with $\lambda_k \geq 0$ and $H_k + \lambda_k I \succeq 0$
- $\|s_k\| \leq \kappa_s$ and $\lambda_k \leq \kappa \lambda \|s_k\|^\alpha$, for some $\alpha \in [0, 1]$.

Class of objectives $A.\alpha$: $f \in C^2$ bounded below; $g$ globally Lipschitz continuous and $H \alpha$-Hölder continuous on the path of the iterates.

Properties of class $M.\alpha$:

- $f \in A.\alpha$ and $M \in M.\alpha \implies \|s_k\| \geq C\|g_{k+1}\|^{\frac{1}{1+\alpha}}$.
- $\|g_{k+1}\| \leq c\|g_k\|^{1+\alpha} \implies$ lower bound on the step.
Examples of methods in $M.\alpha$

Class of methods $M.\alpha$: $x_{k+1} = x_k + (\theta_k)s_k$, $k \geq 0$;

- $(H_k + \lambda_k I)s_k = -g_k$ with $\lambda_k \geq 0$ and $H_k + \lambda_k I \succeq 0$
- $\|s_k\| \leq \kappa_s$ and $\lambda_k \leq \kappa \lambda \|s_k\|^\alpha$.

Examples of methods in $M.\alpha$ (applied to functions in $A.\alpha$):

- Newton’s method: $\lambda_k = 0$ and $\alpha \in [0, 1]$.
- Regularization methods: $\lambda_k = \sigma_k \|s_k\|^\alpha$

  $\implies \alpha = 1$: cubic regularization.

- Linesearch methods, with inexact linesearch $\theta \leq \theta_k \leq \theta$.
- Trust-region methods, when $\lambda_k$ is at least bounded above.
- Goldfeld-Quandt-Trotter: $\lambda_k = -\lambda_{\min}(H_k) + R_k \|g_k\|^\frac{\alpha}{1+\alpha}$.
(Order) optimality of regularization methods

Theorem: Let $\mathcal{M} \in M.\alpha$. Then there exists a function $f^\mathcal{M} \in A.\alpha$ such that $\mathcal{M}$ takes (at least)

$$\epsilon^{-\frac{2+\alpha}{1+\alpha}+\tau} \in [\epsilon^{-\frac{3}{2}+\tau}, \epsilon^{-2+\tau}]$$

iterations/function-evaluations to generate $\|g_k\| \leq \epsilon$, for any $\tau > 0$ arbitrarily small.

$\implies (2 + \alpha)$-regularization method is optimal for the class $M.\alpha$ when applied to functions in $A.\alpha$, as its complexity upper bound coincides in order to the lower bound.

[Extension to examples with finite minimizers: possible.]
Evaluation complexity of constrained optimization
Evaluation complexity of constrained problems

Consider the general NLO problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \quad \text{subject to} \quad c_E(x) = 0; \quad c_I(x) \geq 0,
\end{align*}
\]

where \( f \) and \( c_{E,I} : \mathbb{R}^n \rightarrow \mathbb{R}^{m,p} \) are smooth and nonconvex.

Evaluation complexity of generating a first-order critical (KKT) point to \( \epsilon \) accuracy?
Evaluation complexity of constrained problems

Consider the general NLO problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \quad \text{subject to} \quad c_E(x) = 0; \; c_I(x) \geq 0,
\end{align*}
\]

where \( f \) and \( c_{E,I} : \mathbb{R}^n \rightarrow \mathbb{R}^{m,p} \) are smooth and nonconvex.

Evaluation complexity of generating a first-order critical (KKT) point to \( \epsilon \) accuracy?

- Penalty methods (exact, quadratic), augmented Lagragian: best known bounds are worse than \( O(\epsilon^{-2}) \)

- Phase 1-Phase 2 path-following first-order method: complexity of order \( O(\epsilon^{-2}) \), same as for steepest-descent for unconstrained optimization \[\text{CGT, Math Programming, 2013}\]

Can we devise methods with better evaluation complexity than \( O(\epsilon^{-2}) \) for constrained problems?...
A slight detour: ARC for nonlinear least-squares

Apply ARC to

\[
\minimize_{x \in \mathbb{R}^n} \frac{1}{2} \| r(x) \|^2,
\]

with new Termination Condition (TC): for given \( \epsilon_p, \epsilon_d > 0 \),

\[
\| r(x_k) \| \leq \epsilon_p \quad \text{or} \quad \| g_r(x_k) \| \leq \epsilon_d,
\]

where

\[
g_r(x) \overset{\text{def}}{=} \begin{cases} 
\frac{A(x)^T r(x)}{\| r(x) \|}, & \text{whenever } r(x) \neq 0; \\
0, & \text{otherwise.}
\end{cases}
\]

Main result: Under same conditions as ‘before’ (Lipschitz continuity of Hessian on path of iterates etc.), ARC (with exact or approximate model solution step) will take at most

\[
\left\lceil \kappa_{nls} \epsilon^{-\frac{3}{2}} \right\rceil
\]

residual evaluations to satisfy the new TC, where \( \epsilon = \min(\epsilon_p, \epsilon_d) \).
A two-phase algorithm for constrained problems

Consider now the equality-constrained problem

\[
\begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & c(x) = 0.
\end{align*}
\]
A two-phase algorithm for constrained problems

Consider now the equality-constrained problem

\[
\minimize_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad c(x) = 0.
\]

Idea for a two-phase algorithm:

- get feasible (if possible) by minimizing \( \frac{1}{2} \| c(x) \|^2 \).
- get ‘close’ to the manifold

\[
\mathcal{T}(t) = \{ x \in \mathbb{R}^n : c(x) = 0 \text{ and } f(x) = t \},
\]

for decreasing values of \( t \) from some \( t_1 \) (corresponding to the first feasible iterate).
A second-order algorithm for constrained problems

A Short-Step ARC (ShS - ARC) algorithm:

- **Phase 1 (Feasibility):** apply ARC with newTC to minimize
  \[
  \min_{x \in \mathbb{R}^n} \|c(x)\|^2 \\
  \implies \text{at most } O(\min\{\epsilon_p, \epsilon_d\}^{-\frac{3}{2}}) \text{ evaluations.}
  \]
  If \(\|c(x_p)\| \leq \epsilon_p\), do:

- **Phase 2 (Target-following):**
A second-order algorithm for constrained problems

A Short-Step ARC (ShS - ARC) algorithm:

- **Phase 1 (Feasibility):** apply ARC with newTC to

  \[
  \text{minimize } \|c(x)\|^2 \quad x \in \mathbb{R}^n
  \]

  \[\implies \text{at most } \mathcal{O}(\min\{\epsilon_p, \epsilon_d\}^{-\frac{3}{2}}) \text{ evaluations.}\]

  If \(\|c(x_p)\| \leq \epsilon_p\), do:

- **Phase 2 (Target-following):** successively, take one (successful) ARC step from \(x\) to \(x_+\) to minimize

  \[
  \Phi(x, t)^2 = \|r(x, t)\|^2 = \|c(x)\|^2 + (f(x) - t)^2
  \]

  while \(\|g_r(x_+, t)\| > \epsilon_d\).

  Update \(t \downarrow t_+: \|r(x_+, t_+)\| = \epsilon_p\).
Improved evaluation complexity for EC problems

\[ \| r(x_k, t_k) \| = \epsilon_p \implies \| c(x_k) \| \leq \epsilon_p \text{ for all } k. \]

while \[ \| g_r(x_{k+1}, t_k) \| > \epsilon_d \text{ and } k \text{ successful,} \]

\[ t_k - t_{k+1} \geq \| r(x_k, t_k) \| - \| r(x_{k+1}, t_k) \| \geq \kappa r \frac{3}{2} \epsilon_d \epsilon_p. \]

Assume that \( f, g, H, c \text{ and } J \) are globally Lipschitz continuous; \( f \) bounded below and above in a neighbourhood of feasibility.
Then the ShS - ARC algorithm takes at most

\[ \left[ \kappa_{ec} \cdot \epsilon_d^{-\frac{3}{2}} \epsilon_p^{-\frac{1}{2}} \right] \] (\text{\#}) problem evaluations

to find an iterate \( x_k \) with either

\[ \frac{\| J(x_k)^T c(x_k) \|}{\| c(x_k) \|} \leq \epsilon_d \quad \text{or} \]

\[ \frac{\| g(x_k) + J(x_k)^T y_k \|}{\| (y_k, 1) \|} \leq \epsilon_d \quad \text{and} \quad \| c(x_k) \| \leq \epsilon_p. \]

\text{(\#)} \text{ same as ARC’s evaluation complexity for unconstrained problems when } \epsilon_p := \epsilon \text{ and } \epsilon_d := \epsilon^2/3.
General nonlinearly constrained problems

\[
\text{minimize } f(x) \quad \text{subject to } c(x) = 0 \quad \text{and} \quad x \geq 0.
\]

Improved complexity for problems with general constraints
⇒ ‘merge’ ShS - ARC with ARC for problems with bounds
⇒ same complexity of at most \( O(\epsilon^{-\frac{3}{2}}) \) evaluations

[when lower accuracy required for dual first-order criticality]
‘Cubic regularization’ for global optimization
Overlapping branch & bound for global optimization

Globally minimize [also with J Fowkes (Edinburgh)]

\[ f : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R} \]

where \( \mathcal{D} \) convex, \( f \) nonconvex.  [NP-hard]

\[ L(\mathcal{B}) \leq \min_{x \in \mathcal{B}} f(x) \leq U(\mathcal{B}) \]
Tractable bounds for Lipschitz optimization

Lower bound $L(B)$ obtained by solving the nonconvex cubic regularization subproblem

$$\begin{align*}
\text{minimize} & \quad m_B(x) \\
\text{subject to} & \quad \|x - x_B\|_2 \leq r_B
\end{align*}$$

where $m_B(x) \leq f(x)$ for all $x \in B$, is the lower bound

$$f(x_B) + (x - x_B)^T g(x_B) + \frac{1}{2}(x - x_B)^T H(x_B)(x - x_B) - \frac{L_H(B)}{6} \|x - x_B\|_2^3$$

and where $L_H(B)$ is the Lipschitz constant for Hessian over $B$.

Tractable only over balls $B$ (cubic regularization extension)

[Fowkes, Gould & Farmer ('12); Evtushenko & Posypkin ('12); C, Fowkes & Gould ('13)]
Overlapping branch & bound for global optimization

Globally optimize RBF approximations to:

**Dixon-Szégo test set (serial)**

<table>
<thead>
<tr>
<th>Function</th>
<th>$\omega BB - L_H$</th>
<th>$BB - L_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branin</td>
<td>5s(3s)</td>
<td>63s</td>
</tr>
<tr>
<td>Six-hump Camel</td>
<td>7s(3s)</td>
<td><em>0.05</em></td>
</tr>
<tr>
<td>Goldstein-Price</td>
<td>13s(12s)</td>
<td><em>4000</em></td>
</tr>
<tr>
<td>Shubert</td>
<td>12s(10s)</td>
<td><em>200</em></td>
</tr>
<tr>
<td>Hartman 3</td>
<td>271s(39s)</td>
<td><em>5</em></td>
</tr>
<tr>
<td>Shekel 5</td>
<td><em>0.3</em>(1080s)</td>
<td><em>2</em></td>
</tr>
<tr>
<td>Shekel 7</td>
<td><em>0.4</em>(671s)</td>
<td><em>4</em></td>
</tr>
<tr>
<td>Shekel 10</td>
<td><em>0.5</em>(682s)</td>
<td><em>4</em></td>
</tr>
<tr>
<td>Hartman 6</td>
<td><em>600</em>(50)*</td>
<td><em>100</em></td>
</tr>
</tbody>
</table>

*Algorithm did not achieve $10^{-2}$ tolerance in 50 min, but achieved listed gap.

**Coconut test set (parallel)**

---

New Directions in Nonlinear Optimization: Manchester, 2014 – p. 31/33
Conclusions

Algorithm design profits from complexity analysis.

- Even when convergent, Newton’s method may be as slow as steepest descent.

- ARC has optimal worst-case evaluation complexity amongst a large class of second-order methods.

- Evaluation complexity of constrained optimization, same as of unconstrained optimization for carefully devised methods that stay close to constraints

Cubic regularization: the next generation of optimization software?

(work in progress)
Topics/results not covered

- Problem-dimension dependence of complexity bounds → Jarre/Nesterov example
- Sharp bounds for computing second-order critical points using ARC and trust-region
- Complexity bounds for midly nonsmooth composite objectives, of same order as in the smooth case
- Complexity bounds for derivative-free ARC (CGT), direct-search methods (Vicente et al)
- Complexity bounds for nonsmooth nonconvex problems (Chen et al; Vicente et al)
- Cubic regularization for convex problems (Nesterov; CGT)