Evaluation complexity in smooth constrained and unconstrained optimization

Philippe Toint (with Coralia Cartis and Nick Gould)

Namur Center for Complex Systems (naXys), University of Namur, Belgium

(philippe.toint@fundp.ac.be)

The problem

We consider the unconstrained nonlinear programming problem:

$$\text{minimize} \quad f(x)$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ smooth.

Important special case: the nonlinear least-squares problem

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2$$

for $x \in \mathbb{R}^n$ and $F : \mathbb{R}^n \to \mathbb{R}^m$ smooth.

A useful observation

Note the following: if

- f has gradient g and globally Lipschitz continuous Hessian H with constant $2L$

Taylor, Cauchy-Schwarz and Lipschitz imply

$$\begin{aligned} f(x+s) &= f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle \\ &\quad + \int_0^1 (1-\alpha)\langle s, [H(x+\alpha s) - H(x)]s\rangle \, d\alpha \\ &\leq \underbrace{f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}L\|s\|_2^3}_{m(s)} \end{aligned}$$

$\Longrightarrow$ reducing m from $s = 0$ improves f since $m(0) = f(x)$.

The cubic regularization

Change from trust-regions:

$$\min_s \quad f(x) + \langle s, g(x) \rangle + \tfrac{1}{2} \langle s, H(x) s \rangle \;\; \text{s.t.} \;\; \|s\| \leq \Delta$$

to cubic regularization:

$$\min_s \quad f(x) + \langle s, g(x) \rangle + \tfrac{1}{2} \langle s, H(x) s \rangle + \tfrac{1}{3} \sigma \|s\|^3$$

σ is the (adaptive) regularization parameter

(ideas from Griewank, Weiser/Deuflhard/Erdmann, Nesterov/Polyak, Cartis/Gould/T)

Cubic regularization highlights

$$f(x+s) \le m(s) \equiv f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3} L \|s\|_2^3$$

- Nesterov and Polyak minimize m globally and exactly
  - N.B. m may be non-convex!
  - efficient scheme to do so if H has sparse factors
- global (ultimately rapid) convergence to a 2nd-order critical point of f
- better worst-case function-evaluation complexity than previously known

Obvious questions:

- can we avoid the global Lipschitz requirement?
- can we approximately minimize m and retain good worst-case function-evaluation complexity?
- does this work well in practice?

Adaptive Regularization with Cubic (ARC)

Algorithm 1.1: The ARC Algorithm

Step 0: Initialization: x_0 and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Step computation: Compute s_k for which $m_k(s_k) \le m_k(s_k^C)$

Cauchy point: $s_k^C = -\alpha_k^C g_k$ & $\alpha_k^C = \arg\min_{\alpha \in \mathbf{R}_+} m_k(-\alpha g_k)$

Step 2: Step acceptance: Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}$

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

Step 3: Update the regularization parameter:

$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] = \frac{1}{2}\sigma_k & \text{if } \rho_k > 0.9 & \textit{very successful} \\ [\sigma_k, \gamma_1 \sigma_k] = \sigma_k & \text{if } 0.1 \le \rho_k \le 0.9 & \textit{successful} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] = 2\sigma_k & \text{otherwise} & \textit{unsuccessful} \end{cases}$$

Function-evaluation complexity (1)

How many function evaluations (iterations) are needed to ensure that

$$\|g_k\| \leq \epsilon?$$

So long as for very successful iterations $\sigma_{k+1} \leq \gamma_3 \sigma_k$ for $\gamma_3 < 1$

The basic ARC algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^2} \right\rceil \text{ function evaluations}$$

for some $\kappa_{\rm C}$ independent of ϵ

c.f. steepest descent

Function-evaluation complexity (2)

How many function evaluations (iterations) are needed to ensure that

$$\|g_k\| \leq \epsilon?$$

If H is globally Lipschitz, the s-rule is applied and additionally s_k is the global (line) minimizer of $m_k(\alpha s_k)$ as a function of α, the ARC algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm S}}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

for some $\kappa_{\rm S}$ independent of ϵ.

c.f. Nesterov & Polyak

Note: an $O(\epsilon^{-3})$ bound holds for convergence to second-order critical points.

Function-evaluation complexity (3)

Is the bound in $O(\epsilon^{-3/2})$ sharp? YES!!!

Construct a unidimensional example with

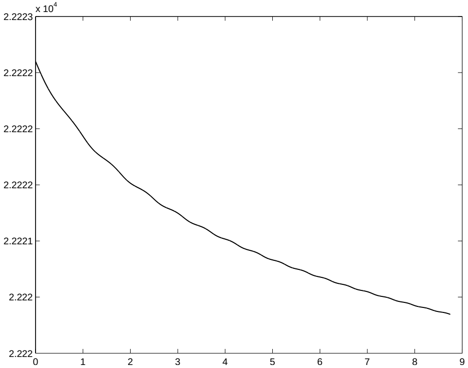
$$x_0 = 0, \quad x_{k+1} = x_k + \left(\frac{1}{k+1}\right)^{\frac{1}{3}+\eta},$$

$$f_0 = \frac{2}{3}\zeta(1+3\eta), \quad f_{k+1} = f_k - \frac{2}{3}\left(\frac{1}{k+1}\right)^{1+3\eta},$$

$$g_k = -\left(\frac{1}{k+1}\right)^{\frac{2}{3}+2\eta}, \quad H_k = 0 \text{ and } \sigma_k = 1,$$

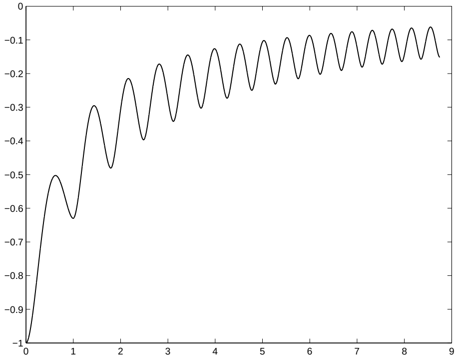
Use Hermite interpolation on $[x_K, x_{k+1}]$.

An example of slow ARC (1)



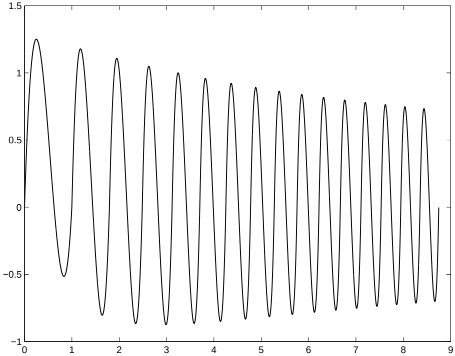
The objective function

An example of slow ARC (2)



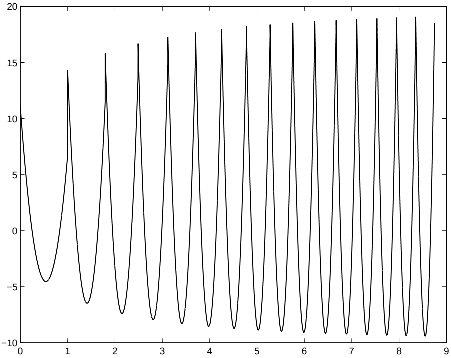
The first derivative

An example of slow ARC (3)



The second derivative

An example of slow ARC (4)



The third derivative

Minimizing the model

$$m(s) \equiv f + s^T g + \tfrac{1}{2} s^T B s + \tfrac{1}{3}\sigma \|s\|_2^3$$

- Small problems:

  use Moré-Sorensen-like method with modified secular equation (also OK as long as factorization is feasible)

- Large problems:

  an iterative Krylov space method

  approximate solution

Numerically sound procedures for computing exact/approximate steps

The main features of adaptive cubic regularization

And the result is. . .

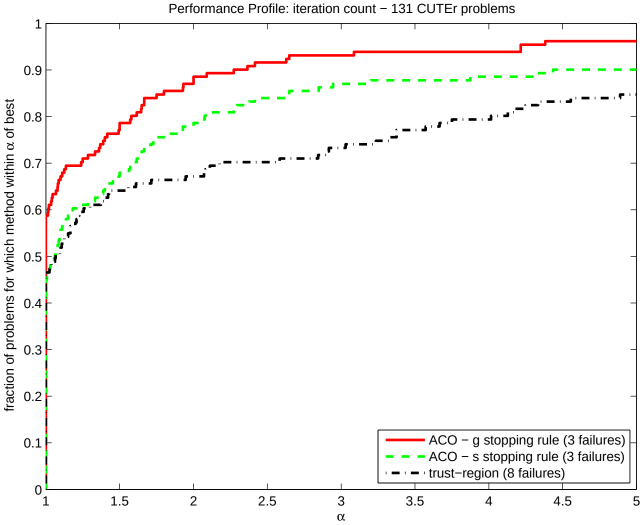
longer steps on ill-conditioned problems

(very satisfactory convergence analysis)

best function-evaluation complexity for nonconvex problems

good performance and reliability

Numerical experience — small problems using Matlab



Without regularization ?

What is known for unregularized (standard) methods?

The steepest descent method requires at most

$$\left\lceil \frac{\kappa_C}{\epsilon^2} \right\rceil \text{ function evaluations}$$

for obtaining $\|g_k\| \le \epsilon$.

Sharp???

Newton's method (when convergent) requires at most

??? function evaluations

for obtaining $\|g_k\| \le \epsilon$.

Slow steepest descent (1)

For steepest descent, the bound of

$$\left\lceil \frac{\kappa_C}{\epsilon^2} \right\rceil \text{ function evaluations}$$

is sharp on functions with Lipschitz continuous gradients.

As before, construct a unidimensional example with

$$x_0 = 0, \quad x_{k+1} = x_k + \alpha_k \left(\frac{1}{k+1} \right)^{\frac{1}{2}+\eta},$$

for some steplength $\alpha_k > 0$ such that

$$0 < \underline{\alpha} \le \alpha_k \le \overline{\alpha} < 2,$$

giving the step

$$s_k \stackrel{\text{def}}{=} x_{k+1} - x_k = \alpha_k \left(\frac{1}{k+1} \right)^{\frac{1}{2}+\eta}.$$

Slow steepest descent (1)

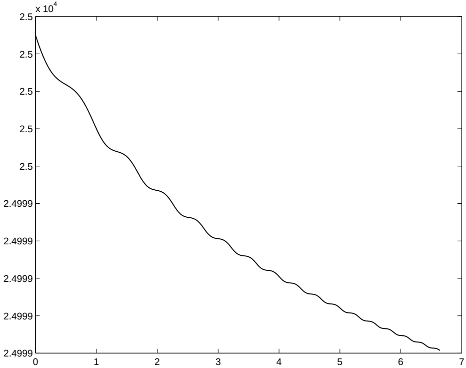
Also set

$$f_0 = \frac{1}{2}\zeta(1+2\eta), \qquad f_{k+1} = f_k - \alpha_k(1 - \tfrac{1}{2}\alpha_k)\left(\frac{1}{k+1}\right)^{1+2\eta},$$

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

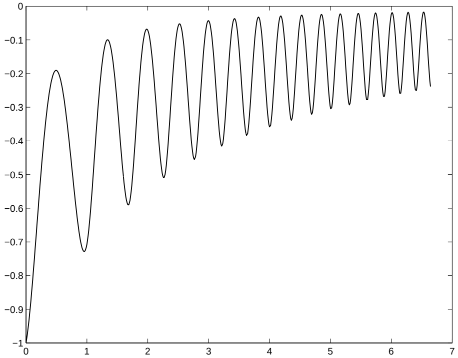
Use Hermite interpolation on $[x_K, x_{k+1}]$.

An example of slow steepest descent (1)



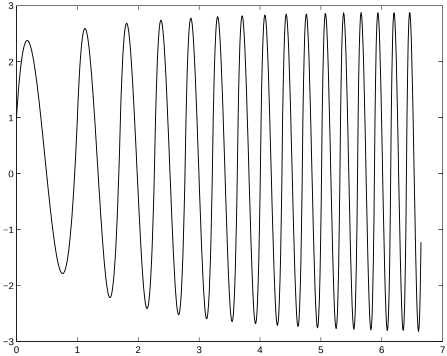
The objective function

An example of slow steepest-descent (2)



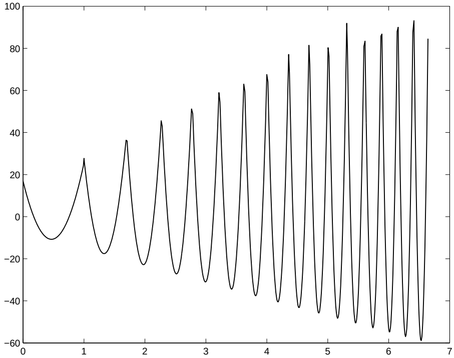
The first derivative

An example of slow steepest-descent (3)



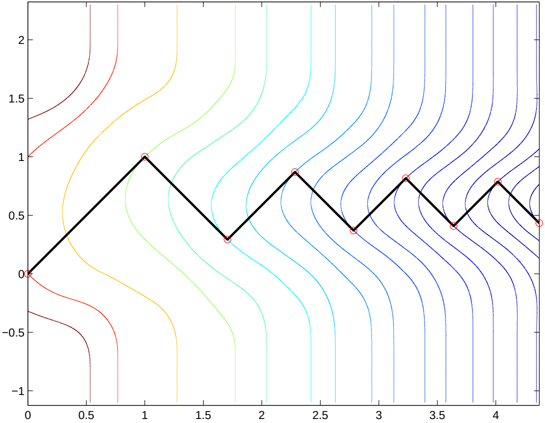
The second derivative

An example of slow steepest descent (4)



The third derivative

Slow steepest descent with exact linesearch



True also if one considers exact linesearch

Slow Newton (1)

A big surprise:

Newton's method may require as much as

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^2} \right\rceil \text{ function evaluations}$$

to obtain $\|g_k\| \leq \epsilon$ on functions with bounded and (segment-wise) Lipschitz continuous Hessians.

Example now bi-dimensional

Slow Newton (2)

The conditions are now:

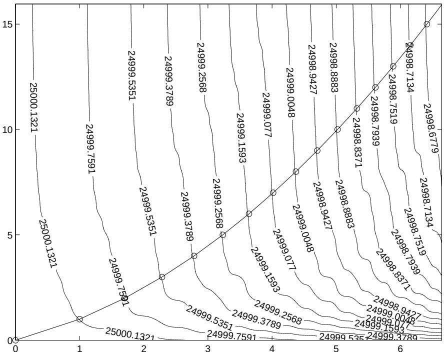
$$x_0 = (0,0)^T, \quad x_{k+1} = x_k + \begin{pmatrix} \left(\frac{1}{k+1}\right)^{\frac{1}{2}+\eta} \\ 1 \end{pmatrix},$$

$$f_0 = \frac{1}{2}\left[\zeta(1+2\eta) + \zeta(2)\right], \quad f_{k+1} = f_k - \frac{1}{2}\left[\left(\frac{1}{k+1}\right)^{1+2\eta} + \left(\frac{1}{k+1}\right)^2\right],$$

$$g_k = -\begin{pmatrix} \left(\frac{1}{k+1}\right)^{\frac{1}{2}+\eta} \\ \left(\frac{1}{k+1}\right)^2 \end{pmatrix}, \text{ and } H_k = \begin{pmatrix} 1 & 0 \\ 0 & \left(\frac{1}{k+1}\right)^2 \end{pmatrix}$$

Use previous example for x_1 and Hermite interpolation on $[x_K, x_{k+1}]$ for x_2.

An example of slow Newton



The path of iterates on the objective's contours

More general second-order methods

Assume that, for $\beta \in (0,1]$, the step is computed by

$$(H_k + \lambda_k I)s_k = -g_k \ \text{ and } \ 0 \le \lambda_k \le \kappa_s \|s_k\|^\beta$$

(ex: Newton, ARC, (TR), ...)

The corresponding method may require as much as

$$\left\lceil \frac{\kappa_C}{\epsilon^{-(\beta+2)/(\beta+1)}} \right\rceil \text{ function evaluations}$$

to obtain $\|g_k\| \le \epsilon$ on functions with bounded and (segment-wise) β-Hölder continuous Hessians.

Note: ranges form ϵ^{-2} to $\epsilon^{-3/2}$

ARC is optimal within this class

The constrained case

Can we apply regularization to the constrained case?

Consider the constrained nonlinear programming problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ & x \in \mathcal{F} \end{array}$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ smooth, and where

$$\mathcal{F} \text{ is convex.}$$

Main ideas:

- exploit (cheap) projections on convex sets
- define using the generalized Cauchy point idea
- prove global convergence + function-evaluation complexity

Constrained step computation (1)

$$\min_s \quad f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}\sigma\|s\|^3$$

subject to

$$x + s \in \mathcal{F}$$

σ is the (adaptive) regularization parameter

Criticality measure: (as before)

$$\chi(x) \stackrel{\rm def}{=} \left|\min_{x+d\in\mathcal{F},\|d\|\leq 1} \langle \nabla_x f(x), d\rangle\right|,$$

The generalized Cauchy point for ARC

Cauchy step: Goldstein-like piecewise linear seach on m_k along the gradient path projected onto $\mathcal{F}$

Find

$$x_k^{\rm GC} = P_{\mathcal{F}}[x_k - t_k^{\rm GC} g_k] \stackrel{\rm def}{=} x_k + s_k^{\rm GC} \quad (t_k^{\rm GC} > 0)$$

such that

$$m_k(x_k^{\rm GC}) \leq f(x_k) + \kappa_{\rm ubs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(below linear approximation)}$$

and either

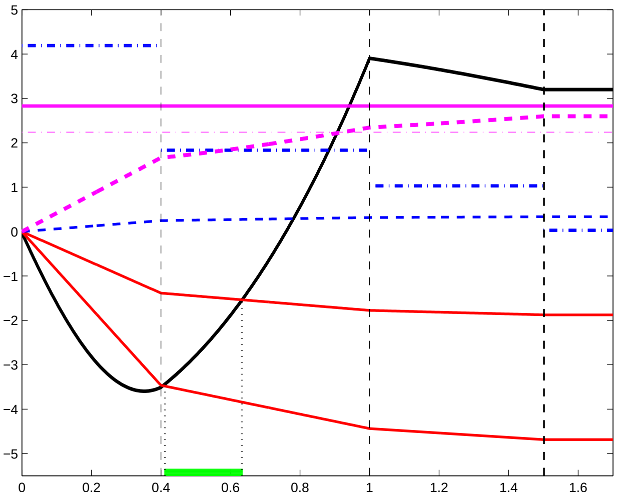
$$m_k(x_k^{\rm GC}) \geq f(x_k) + \kappa_{\rm lbs}\langle g_k, s_k^{\rm GC}\rangle \qquad \text{(above linear approximation)}$$

or

$$\|P_{T(x_k^{\rm GC})}[-g_k]\| \leq \kappa_{\rm epp}|\langle g_k, s_k^{\rm GC}\rangle| \qquad \text{(close to path's end)}$$

no trust-region condition!

Searching for the ARC-GCP



$$m_k(0+s) = -3.57s_1 - 1.5s_2 - s_3 + s_1 s_2 + 3s_2^2 + s_2 s_3 - 2s_3^2 + \frac{1}{3}\|s\|^3 \quad \text{such that} \quad s \leq 1.5$$

A constrained regularized algorithm

Algorithm 3.1: ARC for Convex Constraints (COCARC)

Step 0: Initialization. $x_0 \in \mathcal{F}$, σ_0 given. Compute $f(x_0)$, set $k = 0$.

Step 1: Generalized Cauchy point. If x_k not critical, find the generalized Cauchy point $x_k^{\rm GC}$ by piecewise linear search on the regularized cubic model.

Step 2: Step calculation. Compute s_k and $x_k^+ \stackrel{\rm def}{=} x_k + s_k \in \mathcal{F}$ such that $m_k(x_k^+) \le m_k(x_k^{\rm GC})$.

Step 3: Acceptance of the trial point. Compute $f(x_k^+)$ and ρ_k. If $\rho_k \ge \eta_1$, then $x_{k+1} = x_k + s_k$; otherwise $x_{k+1} = x_k$.

Step 4: Regularisation parameter update. Set

$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] & \text{if } \rho_k \ge \eta_2, \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Function-Evaluation Complexity for COCARC (1)

But

What about function-evaluation complexity?

If, for very successful iterations, $\sigma_{k+1} \leq \gamma_3 \sigma_k$ for $\gamma_3 < 1$, the COCARC algorithm requires at most

$$\left\lceil \frac{\kappa_C}{\epsilon^2} \right\rceil \text{ function evaluations}$$

(for some κ_C independent of ϵ) to achieve $\chi_k \leq \epsilon$

c.f. steepest descent

Do the nicer bounds for unconstrained optimization extend to the constrained case?

Function-evaluation complexity for COCARC (2)

As for unconstrained, impose a termination rule on the subproblem solution:

- Do not terminate solving $\min_{x_k+s\in\mathcal{F}} m_k(x_k+s)$ before

$$\chi_k^{\mathrm{m}}(x_k^+) \le \min(\kappa_{\mathrm{stop}}, \|s_k\|)\,\chi_k$$

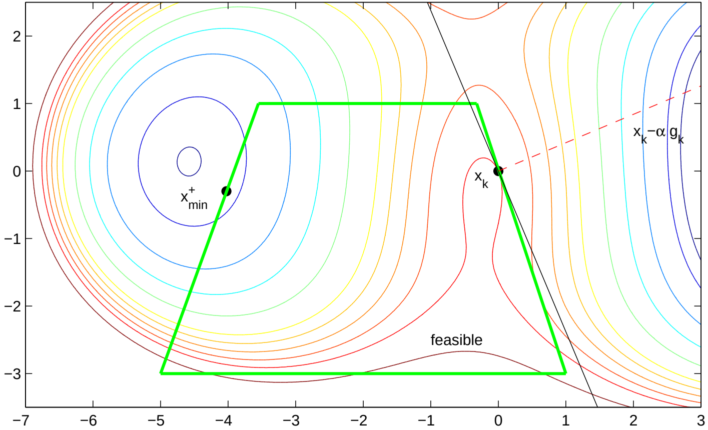
  where

$$\chi_k^{\mathrm{m}}(x) \stackrel{\mathrm{def}}{=} \left| \min_{x+d\in\mathcal{F}, \|d\|\le 1} \langle \nabla_x m_k(x), d\rangle \right|$$

c.f. the “s-rule” for unconstrained

Note: OK at local constrained model minimizers

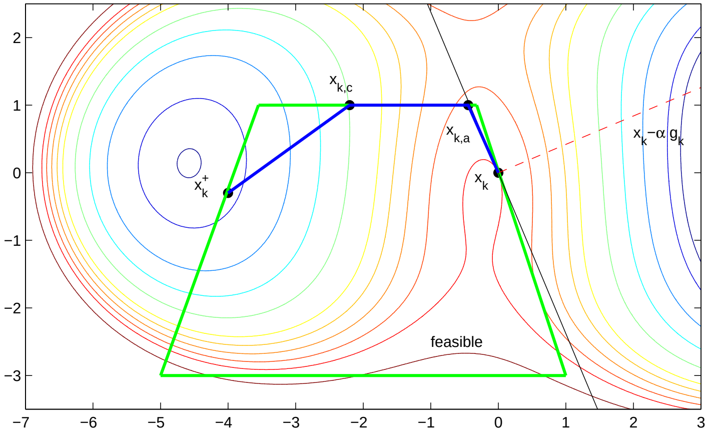
Walking through the pass...



A "beyond the pass" constrained problem with

$$m(x,y) = -x - \tfrac{42}{100}y - \tfrac{3}{10}x^2 - \tfrac{1}{10}y^3 + \tfrac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Walking through the pass...with a sherpa



A piecewise descent path from x_k to x_k^+ on

$$m(x,y) = -x - \tfrac{42}{100}y - \tfrac{3}{10}x^2 - \tfrac{1}{10}y^3 + \tfrac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Function-Evaluation Complexity for COCARC (2)

Assume also

- $x_k \leftarrow x_k^+$ in a bounded number of feasible descent substeps
- $\|H_k - \nabla_{xx} f(x_k)\| \leq \kappa \|s_k\|^2$
- $\nabla_{xx} f(\cdot)$ is globally Lipschitz continuous
- $\{x_k\}$ bounded

The COCARC algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

(for some $\kappa_{\rm C}$ independent of ϵ) to achieve $\chi_k \leq \epsilon$

Caveat: cost of solving the subproblem! c.f. unconstrained case!!!

The general constrained case

Consider the general constrained nonlinear programming problem:

$$\begin{array}{ll} \text{minimize}_x & f(x) \\ \text{such that} & c(x) \begin{pmatrix} \geq \\ = \\ \leq \end{pmatrix} 0 \end{array}$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ smooth.

Complexity of computing an (approximate) first-order critical point?

Question not restricted to cubic regularization algorithms!

A detour: minimizing non-smooth composite functions

A useful tool (and an interesting question in itself): consider the unconstrained problem:

$$\text{minimize}_{\,x} \quad f(x) + h(c(x))$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ smooth and nonconvex, and $h : \mathbb{R}^m \to \mathbb{R}$ non-smooth but convex (ex: $h(\cdot) = \|\cdot\|$).
First-order method: compute a step by solving the (convex) problem

$$\text{minimize}_{\,\|s\|\leq\Delta} \quad \ell(x,s) \stackrel{\text{def}}{=} f(x) + \langle g(x), s\rangle + h(c(x) + J(x)s)$$

for some trust-region radius Δ (also possible using quadratic regularization)
(considered by Nesterov (2007, 2007), Cartis/Gould/T)

Minimizing non-smooth composite functions (2)

Main result:

Assume f, c and h are globally Lipschitz continuous. Then the "algorithm" takes at most

$$O(\epsilon^{-2}) \text{ function evaluations}$$

to achieve

$$\psi(x_k) \leq \epsilon$$

where $\psi(x)$ is a first-order criticality measure defined by

$$\psi(x) \stackrel{\text{def}}{=} \ell(x, 0) - \min_{\|s\| \leq 1} \ell(x, s).$$

A first-order algorithm for EC-NLO

Consider now

$$\begin{array}{ll} \text{minimize}_x & f(x) \\ \text{such that} & c(x) = 0 \end{array}$$

Idea for a first-order algorithm:

1. get feasible (if possible) by minimizing $\|c(x)\|$
2. track the trajectory

$$\mathcal{T}(t) \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n \mid c(x) = 0 \quad \text{and} \quad f(x) = t\}$$

for values of t decreasing from f(first feasible iterate)

A first-order algorithm for EC-NLO (2)

How to do that? A short-step steepest-descent (SSSD) algorithm:

feasibility: apply nonsmooth composite minimization to

$$\min_x \|c(x)\|$$

at most $O(\epsilon^{-2})$ function evaluations

tracking: successively

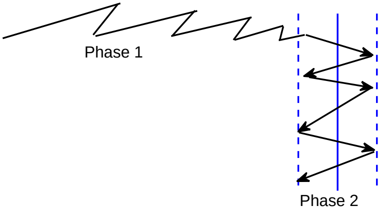
- apply one (successful) step of nonsmooth composite minimization to

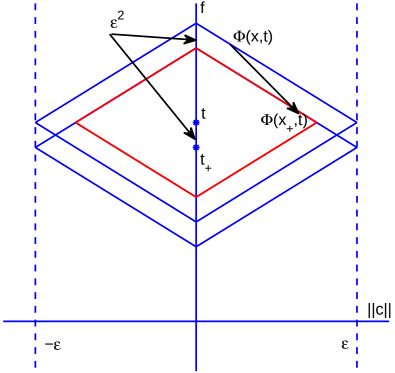
$$\min_x \phi(x) \stackrel{\text{def}}{=} \|c(x)\| + |f(x) - t|$$

- decrease t (proportionally to the decrease in $\phi(x)$)

at most $O(\epsilon^{-2})$ function evaluations !

A view of Algorithm SSSD





A complexity result for EC-NLO

Assume f, and c are globally Lipschitz continuous and f bounded below and above in an ϵ-neighbourhood of feasibility. Then the SSSD algorithm takes at most

$$O(\epsilon^{-2}) \text{ function evaluations}$$

to find an iterate x_k with either

$$\|c(x_k)\| \leq \epsilon \quad \text{and} \quad \|J(x_k)y + g_k\| \leq \epsilon$$

for some y, or

$$\|c(x_k)\| > \kappa_f \epsilon \quad \text{and} \quad \|J(x_k)z\| \leq \epsilon$$

for some z.

($\kappa_f \in (0,1)$, user defined).

Extensions to the general case

Also applies to inequality constrained problems

by replacing

$$\|c(x)\| \quad \text{by} \quad \|\min(c(x),0)\|.$$

A detour via nonlinear least-squares (1)

Consider

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2$$

Apply ARC to obtain $O(\epsilon^{-3/2})$ complexity?

- only yields $\|J(x_k)F(x_k)\| \leq \epsilon$!
- requires unpalatably strong conditions on $J(x)$!

Turn to the "scaled residual"

$$r(x_k) \stackrel{\text{def}}{=} \begin{cases} \dfrac{\|J(x_k)F(x_k)\|}{\|F(x_k)\|} & \text{if } \|F(x_k)\| > 0 \\ 0 & \text{otherwise} \end{cases}$$

Copes with both zero and nonzero residuals !

A detour via nonlinear least-squares (2)

Assume f has Lipschitz Hessian. Then the ARC algorithm takes at most

$$O(\epsilon^{-3/2}) \text{ function evaluations}$$

to find an iterate x_k with either $\|r(x_k)\| \leq \epsilon$ or $\|F(x_k)\| \leq \epsilon$.

- No requierement on regularity for $J(x)$!
- Applicable in phase 1 of an algorithm for EC-NLO ?

Second-order compexity for EC-NLO (1)

A short-step ARC (ShS-ARC) algorithm

feasibility: apply ARC (with $\|r(x_k)\|$ stopping rule) to

$$\min_x \|c(x)\|^2$$

> at most $O(\epsilon^{-3/2})$ function evaluations

tracking: successively

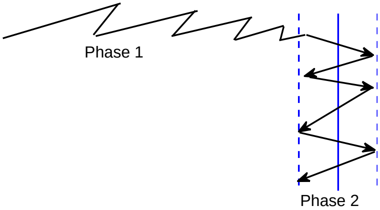
- apply one (successful) step of ARC (with $\|r(x_k)\|$ stopping rule) to

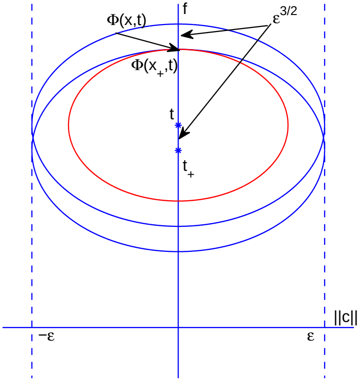
$$\min_x \phi(x) \stackrel{\text{def}}{=} \|c(x)\|^2 + (f(x) - t)^2$$

- decrease t (proportionally to the decrease in $\phi(x)$)

> at most $O(\epsilon^{-3/2})$ function evaluations !

A view of Algorithm ShS-ARC





Second-order complexity for EC-NLO (2)

Assume f, and c are globally Lipschitz continuous and f bounded below and above in an ϵ-neighbourhood of feasibility. Then the ShS-ARC algorithm takes at most

$$O(\epsilon^{-3/2}) \text{ function evaluations}$$

to find an iterate x_k with either

$$\|c(x_k)\| \leq \epsilon \quad \text{and} \quad \|J(x_k)y + g_k\| \leq \epsilon^{2/3}$$

for some y, or

$$\|c(x_k)\| > \kappa_f \epsilon \quad \text{and} \quad \|J(x_k)z\| \leq \epsilon$$

for some z.

Conclusions

- Many open questions . . . but very interesting
- Analysis for unconstrained second-order optimality also possible
- Jarre's example $\Rightarrow$ global optimization much harder
- Algorithm design profits from complexity analysis
- Many issues regarding regularizations still unresolved
- ARC is optimal amongst second-order method

Many thanks for your attention!