Recent topics in complexity for nonconvex optimization problems

Philippe Toint

(with S. Bellavia, C. Cartis, N. Gould, G. Gurioli and B. Morini)

Namur Center for Complex Systems (naXys), University of Namur, Belgium
(philippe.toint@unamur.be)

SFO Seminar 2021, June 2021

The problem (again)

We consider the unconstrained nonlinear programming problem:

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

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

For now, focus on the

unconstrained case

but we are also interested in the case featuring

inexpensive constraints

Adaptive regularization

Adaptive regularization methods iteratively compute steps by minimizing

$$m(s) \stackrel{\text{def}}{=} f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3} \sigma_k \|s\|_2^3 = T_{f,2}(x,s) + \tfrac{1}{3} \sigma_k \|s\|_2^3$$

until an approximate first-order minimizer is obtained:

$$\|\nabla_s m(s)\| \leq \kappa_{\text{stop}} \|s\|^2$$

Note: no global optimization involved.

Second-order Adaptive Regularization (AR2)

Algorithm 1.1: The AR2 Algorithm

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

Step 1: Termination: If $\|g_k\| \leq \epsilon$, terminate.

Step 2: Step computation:

Compute s_k such that $m_k(s_k) \leq m_k(0)$ and $\|\nabla_s m(s_k)\| \leq \kappa_{\rm stop}\|s_k\|^2$.

Step 3: Step acceptance:

Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{f,2}(x_k, s_k)}$

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

Step 4: Update the regularization parameter:

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

Evaluation complexity: an important result

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

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

If H is globally Lipschitz and the s-rule is applied, the AR2 algorithm requires at most

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

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

"Nesterov & Polyak",

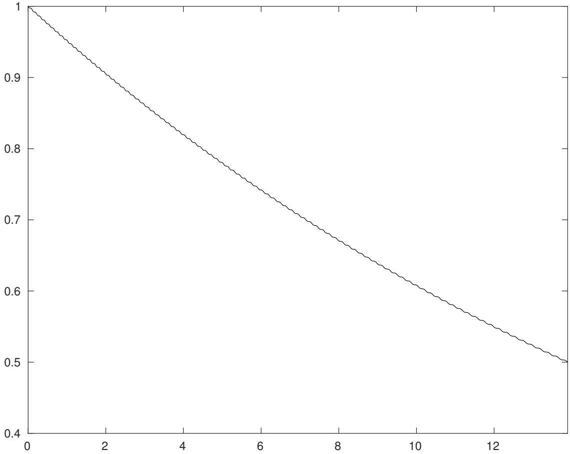
Cartis, Gould, T., 2011, Birgin, Gardenghi, Martinez, Santos, T., 2017

Note:

- The above result is sharp (in order of ϵ)!
- An $O(\epsilon^{-3})$ bound holds for convergence to second-order critical points.

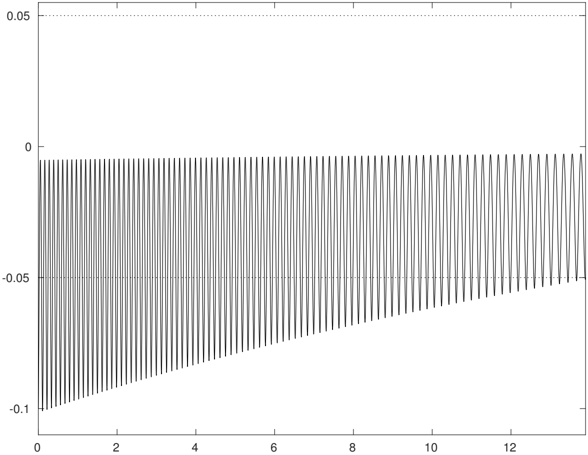
Evaluation complexity: sharpness

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



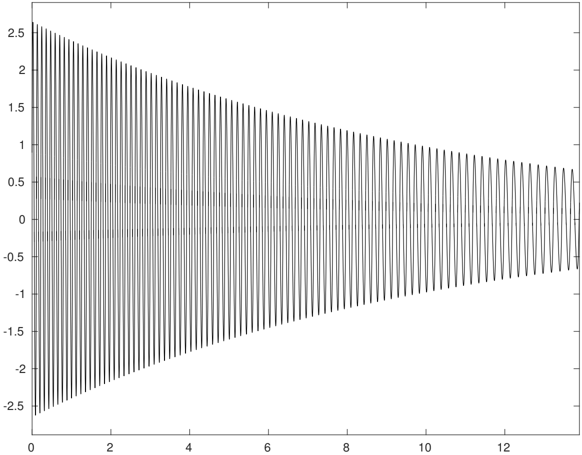
The objective function

An example of slow AR2 (2)



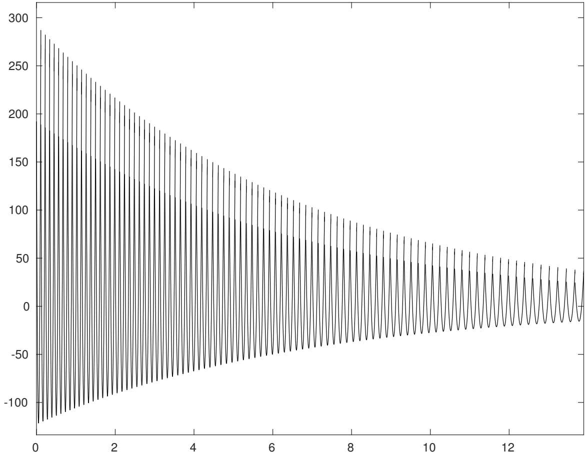
The first derivative

An example of slow AR2 (3)



The second derivative

An example of slow AR2 (4)



The third derivative

Slow steepest descent (1)

The steepest descent method with requires at most

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

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

Nesterov

Sharp??? YES

Newton's method (when convergent) requires at most

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

for obtaining $\|g_k\| \leq \epsilon$!!!!

High-order models for first-order points (1)

What happens if one considers the model

$$m_k(s) = T_{f,p}(x_k, s) + \frac{\sigma_k}{p!}\|s\|_2^{p+1}$$

where

$$T_{f,p}(x,s) = f(x) + \sum_{j=1}^{p} \frac{1}{j!}\nabla_x^j f(x)[s]^j$$

terminating the step computation when

$$\|\nabla_s m(s_k)\| \le \kappa_{\text{stop}}\|s_k\|^p$$

High-order models for first-order points (2)

unconstrained ϵ-approximate 1rst-order-necessary minimizer after at most

$$\frac{f(x_0) - f_{\rm low}}{\kappa} \, \epsilon^{-\frac{p+1}{p}}$$

function and gradient evaluations

Birgin, Gardhenghi, Martinez, Santos, T., 2017

One then wonders. . .

If one uses a model of degree p ($T_{f,p}(x,s)$), why be satisfied with first- or second-order critical points???

What do we mean by critical points of order larger than 2 ???

What are necessary optimality conditions for order larger than 2 ???

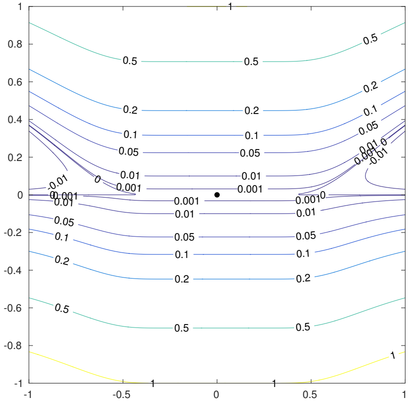
Not an obvious question!

A sobering example (1)

Consider the unconstrained minimization of

$$f(x_1, x_2) = \begin{cases} x_2\left(x_2 - e^{-1/x_1^2}\right) & \text{if } \; x_1 \neq 0, \\ x_2^2 & \text{if } \; x_1 = 0, \end{cases}$$

Peano (1884), Hancock (1917)



A sobering example (2)

Conclusions:

- looking at optimality along straight lines is not enough
- depending on Taylor's expansion for necessary conditions is not always possible

Even worse:

$$f(x_1, x_2) = \begin{cases} x_2 \left(x_2 - \sin(1/x_1) e^{-1/x_1^2} \right) & \text{if } \; x_1 \neq 0, \\ x_2^2 & \text{if } \; x_1 = 0, \end{cases}$$

(no continuous descent path from 0, although not a local minimizer!!!)

Hopeless?

A new (approximate) optimality measure

Define, for some small $\delta > 0$, $(\mathcal{F} = \mathbb{R}^n)$

$$\phi_{f,j}^{\delta}(x) \stackrel{\rm def}{=} f(x) - \text{globmin}_{\substack{x+d\in\mathcal{F}\\ \|d\|\leq\delta}} T_{f,j}(x,d),$$

x is a (strong) (ϵ,δ)-approximate qth-order-necessary minimizer

$$\Leftrightarrow$$

$$\phi_{f,j}^{\delta_j}(x) \leq \epsilon_j \frac{\delta_j^j}{j!} \quad \text{for} \quad j \in \{1,\ldots,q\}$$

for some $\delta \in (0,1]^q$.

- $\phi_{f,j}^{\delta}(x)$ is continuous as a function of x for all j.
- $\phi_{f,j}^{\delta}(x) = o\left(\frac{\delta^j}{j!}\right)$ is a necessary optimality condition

Approximate unconstrained optimality

Familiar results for low orders: when $q = 1$

$$\phi^{\delta}_{f,1}(x) = \|\nabla_x f(x)\|\,\delta \Rightarrow \|\nabla_x f(x)\| \leq \epsilon_1$$

while, for $q = 2$,

$$\left.\begin{array}{c} \|\nabla_x f(x)\| = 0 \\ \lambda_{\min}(\nabla_x^2 f(x)) \geq -\epsilon \end{array}\right\} \Rightarrow \phi^{\delta}_{f,2}(x) \leq \epsilon_2 \frac{\delta^2}{2}$$

Introducing inexpensive constraints

Constraints are inexpensive

$\Leftrightarrow$

their evaluation/enforcement has negligible cost
(compared with that of evaluating f)

- evaluation complexity for the constrained problem well measured in counting evaluations of f and its derivatives
- many well-known and important examples
  - bound constraints
  - convex constraints with cheap projections
  - parametric constraints
  - ...
- the global minimization defining $\phi_{f,j}^{\delta}(x)$ must be conducted in $\mathcal{F}$!

From now on: $\mathcal{F} \stackrel{\text{def}}{=}$ (inexpensive) feasible set

A very general optimization problem

Our aim:

Compute an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

where

- $p \geq q \geq 1$,
- $\nabla_x^p f(x)$ is Lipschitz continuous
- $\mathcal{F}$ is an inexpensive feasible set

Note:

1. no convexity assumption of f
2. no convexity assumption on $\mathcal{F}$
3. Lipschitz can be extended to Hölder

A (theoretical) regularization algorithm

Algorithm 3.1: The ARqp algorithm for qth-order optimality

Step 0: Initialization: x_0, $\delta_{-1} \in (0,1]^q$ and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Stop?: If $\phi_{f,j}^{\delta_{k-1,j}}(x_k) \leq \epsilon_j \delta_{k-1,j}^j / j!$ for $j \in \{1,\ldots,q\}$, stop.

Step 2: Step computation:

Compute* s_k such that $x_k + s_k \in \mathcal{F}$, $m_k(s_k) \leq m_k(0)$ and

$$\phi_{m_k,j}^{\delta_{k,j}}(x_k + s_k) \leq \theta \epsilon_j \frac{\delta_{k,j}{}^j}{j!} \quad (j \in \{1,\ldots,q\})$$

Step 3: Step acceptance:

Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{f,p}(x_k, s_k)}$

and set $x_{k+1} = x_k + s_k$ if $\rho_k > 0.1$ or $x_{k+1} = x_k$ otherwise.

Step 4: Update the regularization parameter:

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

Comments on the algorithm

1. for $q = 1$ and $q = 2$, computing $\phi_{f,j}^{\delta_{k-1,j}}(x_k)$ is easy
   - $q = 1$: analytic solution
   - $q = 2$: trust-region subproblem with unit radius

   $\Rightarrow$ practical algorithm
2. for $q > 2$: hard problem in general

   $\Rightarrow$ conceptual algorithm

Define

easy case: $\left[\ q \leq 2 \ \text{ and } \ \mathcal{F} = \mathbb{R}^n \ \right]$ or $\left[\ q = 1 \ \text{ and } \ \mathcal{F} \ \text{ is convex } \ \right]$

hard case: all other cases.

The main result

The ARqp algorithm is well-defined and

The ARqp algorithm finds an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

in at most

$$\begin{cases} O\left(\epsilon^{-\frac{p+1}{p-q+1}}\right) & \text{if easy} \\ O\left(\epsilon^{-q\,\frac{p+1}{p}}\right) & \text{if hard} \end{cases}$$

iterations and evaluations of the objective function and its p first derivatives. Moreover, this bound is sharp.

What this theorem does

1. generalizes ALL known complexity results for regularization methods to

   arbitrary degree p, arbitrary order q and arbitrary smoothness $p+1$

2. applies to very general constrained problems
3. generalizes the lower complexity bound of Carmon at al., 2018, to arbitrary dimension, arbitrary order and to constrained problems
4. provides a considerably better complexity order than the bound

   $$O\left(\epsilon^{-(q+1)}\right)$$

   known for unconstrained trust-region algorithms (Cartis, Gould, T., 2017)
   Note: linesearch methods all fail for $q > 3$!
5. is provably optimal within a wide class of algorithms (Cartis, Gould, T., 2018 for $p \leq 2$)

Further extensions

Recent advances:

- in smooth Banach spaces (for $q = 1$), using a new method to minimize polynomials using a Hölder regularization (Gratton, Jerad, T., 2021)
- when using a regularization in general possibly non-smooth norms (for $q \leq p \leq 2$), despite the non-smoothness of the model m_k
  - step termination tests not on m_k but on $T_{f,2}(x_k, s_k)$ ($\Rightarrow$ allows Newton steps)
  - even more compact complexity analysis!
  - a specialized method for finding "second-order" points when minimizing quadratic polynomials regularized with a non-smooth norm (and its complexity)

(Gratton, T., 2021)

Moving on: allowing inexact evaluations

A common observation:

In many applications, it is necessary/useful to evaluate $f(x)$ and/or $\nabla_x^j f(x)$ inexactly

1. complicated computations involving truncated iterative processes
2. variable accuracy schemes
3. sampling techniques (machine learning)
4. finite-differences,
5. ...

Focus on the case where f and/or all its derivatives are inexact

The implicit dynamic accuracy (IDA) framework

Suppose that

- f is exact
- the absolute accuracies of the i-th derivative satisfy a bound

$$\|\overline{\nabla_x^i f}(x_k) - \nabla_x^i f(x_k)\| \leq \kappa_{\nabla,i}\, h_{k,i} \quad (i \in \{1, \ldots, j\})$$

for some accuracy goal $h_{k,i}$ specified by the algorithm before their computation and some unknown constant $\kappa_{\nabla,i}$.

Implicit Dynamic Accuracy (IDA)

Examples:

- finite-difference estimations
- multivariate polynomial interpolation/regression (DFO)
- ...

Inexactness consequences and accuracy enforcement

Denote inexact quantities with overbars.

Because only inexact derivatives are available:

$$\nabla_x^i f(x_k) \to \overline{\nabla_x^i f}(x_k), \quad T_{f,j}(x_k,s) \to \overline{T}_{f,j}(x_k,s) \quad \phi_{f,j}^{\delta_{k,j}}(x_k) \to \overline{\phi}_{f,j}^{\delta_{k,j}}(x_k)$$

Accuracy goal management: require

$$h_{k,i} \le \kappa_s \|s_k\|^{p-i+1} \quad (i \in \{1,\ldots,p\})$$

$\Rightarrow$ more accuracy for low-order derivatives

Consequences:

$$|T_{f,j}(x_k,s) - \overline{T}_{f,j}(x_k,s)| \le 2\kappa_{\nabla,\max}\|s\|^{p+1}$$

$$\phi_{f,j}^{\delta_{k,j}}(x_k) \le \overline{\phi}_{f,j}^{\delta_{k,j}}(x_k) + 6\kappa_{\nabla,\max} h_{k,\max}$$

An IDA regularization algorithm

Algorithm 4.1: The ARqpIDA algorithm for qth-order optimality

Step 0: Initialization: x_0, $\delta_{-1} \in (0,1]^q$ and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Approx. optimal? Set $\delta_k = \delta_{s_{k-1}}$. If

$$\overline{\phi}_{f,j}^{\delta_{k,j}}(x_k) \leq \tfrac{1}{2}\epsilon_j \delta_{k,j}^j / j! \;\text{ for }\; j \in \{1, \ldots, q\},$$

go to Step 5. Else, ensure that $\Delta m_k(d_{k,j}) \geq \frac{1}{4}\epsilon_j \delta_{k,j}^j / j!$ by possibly reducing δ_k and returning to Step 1.

Step 2: Step computation:

Compute s_k such that $x_k + s_k \in \mathcal{F}$, $\Delta m_k(s_k) \geq \Delta m_k(d_{k,j})$ and

$$\phi_{m_k,i}^{\delta_{s_k,i}}(s_k) \leq \theta \epsilon_i \, \delta_{s_k,i}^i / i! \quad (i \in \{1, \ldots, q\})$$

If accuracy test fails, go to Step 5.

Step 3: Step acceptance: [As before.]

Step 4: Update the regularization parameter: [As before.]

Step 5: Improve accuracy: $h_{k+1,i} = \frac{1}{2} h_{k,i}$ $(i \in \{1, \ldots, p\})$.

An IDA regularization algorithm: comments

Notes:

- no termination rule, but optimality reached...
- $d_{k,j}$ plays the role of a generalized Cauchy point
- some hidden (unimportant) details
- approx. optimality test can be organized in a loop over successive orders $j = 1, \ldots, q$
- no need to check the condition on $\phi_{m_k,i}^{\delta_{s_k},i}(s_k)$ if the step is large.
- A trust-region variant (TRqIDA) exists

An IDA regularization algorithm: complexity

The ARqpIDA algorithm finds an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

in at most

$$\begin{cases} O\left(\epsilon^{-\frac{p+1}{p-q+1}} + |\log(\epsilon)|\right) & \text{if easy} \\ O\left(\epsilon^{-q\,\frac{p+1}{p}} + |\log(\epsilon)|\right) & \text{if hard} \end{cases}$$

iterations and evaluations of the objective function and its p first derivatives.

Complexity for TRqIDA: $O\left(\epsilon^{-(q+1)} + |\log(\epsilon)|\right)$

The explicit dynamic accuracy (EDA) framework

Suppose now that

- the absolute accuracy of f
- the absolute accuracies of the i-th derivative satisfy a bound

$$\|\overline{\nabla_x^i f}(x_k) - \nabla_x^i f(x_k)\| \leq \zeta_{k,i} \quad (i \in \{1, \ldots, j\})$$

for some accuracy requests $\zeta_{k,i}$ specified by the algorithm before their computation

Explicit Dynamic Accuracy (EDA)

Examples:

- truncated iterative processes
- variable accuracy computations
- ...

Inexactness consequences and accuracy enforcement

Again using

$$\overline{\nabla_x^i f}(x_k), \quad \overline{T}_{f,j}(x_k, s) \quad \text{and} \quad \overline{\phi}_{f,j}^{\delta_{k,j}}(x_k)$$

because of inexact derivatives, but also now

$$\overline{f}(x_k) \quad \text{and} \quad \overline{f}(x_k + s_k)$$

Control both

- the relative error of $\overline{\Delta T}_{f,j}(x_k, s_k)$
- the absolute error of $\overline{f}(x_k)$ and $\overline{f}(x_k + s_k)$

by suitably adapting the requests $\zeta_{k,i}$.

Inexactness consequences and accuracy enforcement (2)

- need of a VERIFY algorithm to check if the $\zeta_{k,i}$ are small enough to ensure that

$$|\Delta T_{f,j}(x_k, s_k) - \overline{\Delta T}_{f,j}(x_k, s_k)| \leq \omega|\overline{\Delta T}_{f,j}(x_k, s_k)|$$

VERIFY for

$$\begin{array}{rcl} \overline{\phi}_{f,j}^{\delta_{k,j}}(x_k) & \rightarrow & V[\overline{\phi}_{f,j}^{\delta_{k,j}}(x_k)] \\ \overline{\Delta T}_{f,j}(x_k, d_{k,j}) & \rightarrow & V[\overline{\Delta T}_{f,j}(x_k, d_{k,j})] \\ \overline{\phi}_{m_k,i}^{\delta_{s_k,i}}(s_k) & \rightarrow & V[\overline{\phi}_{m_k,i}^{\delta_{s_k,i}}(s_k)] \end{array}$$

- need to ensure that $\zeta_{k,i}$ are small enough to ensure that

$$|f(x_k + s_k) - \overline{f}(x_k + s_k)| \leq \omega|\overline{\Delta T}_{f,j}(x_k, s_k)|$$

$$|f(x_k) - \overline{f}(x_k)| \leq \omega|\overline{\Delta T}_{f,j}(x_k, s_k)|$$

An EDA regularization algorithm

Algorithm 4.2: The ARqpEDA algorithm for qth-order optimality

Step 0: Initialization: x_0, $\delta_{-1} \in (0,1]^q$ and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Terminate? Set $\delta_k = \delta_{s_{k-1}}$. Terminate if

$$V[\overline{\phi}_{f,j}^{\delta_{k,j}}(x_k)] \leq (\epsilon_j/(1+\omega)\delta_{k,j}^j/j! \text{ for } j \in \{1,\ldots,q\}.$$

If VERIFY fails for $\overline{\phi}_{f,j}^{\delta_{k,j}}(x_k)$, go to Step 5. Else, reduce δ_k to ensure that $\Delta m_k(d_{k,j}) \geq (\epsilon_j/2(1+\omega))\delta_{k,j}^j/j!$ and go to Step 1.

Step 2: Step computation:

Compute s_k such that $x_k + s_k \in \mathcal{F}$, $V[\Delta m_k(s_k)] \geq \Delta m_k(d_{k,j})$ and

$$V[\phi_{m_k,i}^{\delta_{s_k,i}}(s_k)] \leq (\theta(1-\omega)/(1+\omega))\epsilon_i\, \delta_{s_k,i}^i/i! \quad (i \in \{1,\ldots,q\})$$

If one of the two calls to VERIFY fails, go to Step 5.

Step 3: Step acceptance: [As before using $\overline{f}(x_k)$ and $\overline{f}(x_k+s_k)$.]

Step 4: Update the regularization parameter: [As before.]

Step 5: Improve accuracy: $\zeta_{k+1,i} = \frac{1}{2}\zeta_{k,i}$ $(i \in \{1,\ldots,p\})$.

An EDA regularization algorithm: comments

Notes:

- uses a proper termination rule!
- as before, $d_{k,j}$ plays the role of a generalized Cauchy point
- lots of hidden details
- approx. optimality test can be organized in a loop over successive orders $j = 1, \ldots, q$
- no need to check the condition on $\phi_{m_k,i}^{\delta_{s_k,i}}(s_k)$ if the step is large.
- A trust-region variant (TRqEDA) exists

An EDA regularization algorithm: (tentative) complexity

The ARqpEDA algorithm finds an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

in at most

$$\begin{cases} O\left(\epsilon^{-\frac{p+1}{p-q+1}} + |\log(\epsilon)|\right) & \text{if easy} \\ O\left(\epsilon^{-q\,\frac{p+1}{p}} + |\log(\epsilon)|\right) & \text{if hard} \end{cases}$$

iterations and evaluations of the objective function and its p first derivatives.

Complexity for TRqEDA: $O\left(\epsilon^{-(q+1)} + |\log(\epsilon)|\right)$

A semi-stochastic context

Suppose now that the inequalities

$$|f(x_k+s_k)-\overline{f}(x_k+s_k)| \le \omega|\overline{\Delta T}_{f,j}(x_k,s_k)|$$

$$|f(x_k)-\overline{f}(x_k)| \le \omega|\overline{\Delta T}_{f,j}(x_k,s_k)|$$

are enforceable, but that derivatives values are affected by random noise.
$\Longrightarrow$ no way to ensure any of the two above accuracy models (IDA, EDA)!

Semi-stochastic framework

Example: DFO using a smoothed objective function value and random finite-differences for derivatives.

Question: Can ARqp still be applied?

A semi-stochastic regularization algorithm

Algorithm 5.1: The SARqp algorithm for qth-order optimality

Step 0: Initialization: x_0, $\delta_{-1} \in (0,1]^q$ and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Step computation:

Compute s_k such that $x_k + s_k \in \mathcal{F}$, $\overline{m}_k(s_k) \le \overline{m}_k(0)$ and

$$\overline{\phi}_{m_k,j}^{\delta_{k,j}}(x_k + s_k) \le \theta \epsilon_j \frac{\delta_{k,j}^j}{j!} \quad (j \in \{1,\ldots,q\})$$

Step 2: Step acceptance:

Compute "ω-accurate $\overline{f}(x_k + s_k)$ and (if necessary) $\overline{f}(x_k)$. Set

$$\rho_k = \begin{cases} \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{f,p}(x_k, s_k)} & \text{if} \quad f(x_k) > T_{f,p}(x_k, s_k) \\ +\infty & \text{otherwise.} \end{cases}$$

and set $x_{k+1} = x_k + s_k$ if $\rho_k > 0.1$ or $x_{k+1} = x_k$ otherwise.

Step 3: Update the regularization parameter: [As usual.]

An semi-stochastic regularization algorithm: comments

Notes:

- no termination at all!
- no need to check the condition on $\overline{\phi}_{m_k,i}^{\delta_{s_k,i}}(s_k)$ if the step is large.
- A trust-region variant (STRq) is being developped

An semi-stochastic regularization algorithm: complexity

A *informal* statement of our assumptions:

Consider the events

$$\{|\overline{\Delta T}_{f,p}(X_k, S_k) - \Delta T_{f,p}(X_k, S_k)| \le \omega \overline{\Delta T}_{f,p}(X_k, S_k)\}$$
$$\{|\overline{\Delta T}_{m_k,j}(S_k, D_{k,j}) - \Delta T_{m_k,j}(S_k, D_{k,j})| \le \omega \overline{\Delta T}_{m_k,j}(S_k, D_{k,j})\}$$
$$\{|\overline{\Delta T}_{m_k,j}(S_k, \overline{D}_{k,j}) - \Delta T_{m_k,j}(S_k, D_{k,j})| \le \omega \overline{\Delta T}_{m_k,j}(S_k, D_{k,j})\}$$
$$\{\max_{\ell \in \{2,\dots,p\}} \|\overline{\nabla_x^\ell f}(X_k)\| \le \Theta\}.$$

We assume that

$$\mathbb{P}\mathrm{r}\Big[\text{these events occur}\Big|\text{conditioned by the past}\Big] > \tfrac{1}{2}$$

$+$ f bounded below and Lipschitz continuity of $\{\nabla_x^i f\}_{i=1}^p$

Let

$$N_\epsilon = \inf\left\{k \ge 0 \mid \phi_{f,j}^{\Delta_{k-1,j}}(X_k) \le \epsilon_j \frac{\Delta_{k-1,j}^j}{j!} \text{ for } j \in \{1,\dots,q\}\right\}.$$

An semi-stochastic regularization algorithm: complexity (2)

If the SARqp algorithm is applied to the problem

$$\min_{x \in \mathcal{F}} f(x)$$

then, under the stated assumptions,

$$\mathbb{E}\left[N_\epsilon\right] = \begin{cases} O\left(\epsilon^{-\frac{p+1}{p-q+1}}\right) & \text{if easy} \\ O\left(\epsilon^{-q\,\frac{p+1}{p}}\right) & \text{if hard} \end{cases}$$

$\Longrightarrow$ the complexity of ARqp is unaffected provided the model is "ω-accurate" sufficiently often

Conclusions

A more global view (ignoring $|\log(\epsilon)|$ terms)

		weak minimizers		strong minimizers				
	inexpensive constraints	non-composite ($h = 0$)		non-composite ($h = 0$)		composite: h convex		composite: h non-convex
$q = 1$	none	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-2}\right)$
	convex	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-2}\right)$
	non-convex	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-2}\right)$		$\mathcal{O}\left(\epsilon^{-2}\right)$
$q = 2$	none	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-3}\right)$		$\mathcal{O}\left(\epsilon^{-3}\right)$
	convex	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-3}\right)$		$\mathcal{O}\left(\epsilon^{-3}\right)$
	non-convex	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{2(p+1)}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-3}\right)$		$\mathcal{O}\left(\epsilon^{-3}\right)$
$q > 2$	none, or general	$\mathcal{O}\left(\epsilon^{-\frac{p+1}{p-q+1}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-\frac{q(p+1)}{p}}\right)$	sharp	$\mathcal{O}\left(\epsilon^{-(q+1)}\right)$		$\mathcal{O}\left(\epsilon^{-(q+1)}\right)$

Inexact evaluations (deterministic or stochastic)
do not (significantly) affect the complexity

Perspectives

Complexity for expensive constraints for $q > 1$?

A "completely" stochastic approach of inexact evaluation

Optimization in variable arithmetic precision

etc., etc., etc.

Thank you for your attention!

Some references

C. Cartis, N. Gould and Ph. L. Toint,
"Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints", SIOPT, vol. 30(1), pp. 513-541, 2020

C. Cartis and N. I. M. Gould and Ph. L. Toint,
"Strong Evaluation Complexity Bounds for Arbitrary-Order Optimization of Nonconvex Nonsmooth Composite Functions", arXiv:2001.10802, 2020.

S. Bellavia, G. Gurioli, B. Morini and Ph. L. Toint,
"Deterministic and stochastic inexact regularization algorithms for nonconvex optimization with optimal complexity", SIOPT, vol. 29(4), pp. 2881-2915, 2019.

S. Bellavia, G. Gurioli, B. Morini and Ph. L. Toint,
"High-order Evaluation Complexity of a Stochastic Adaptive Regularization Algorithm for Nonconvex Optimization Using Inexact Function Evaluations and Randomly Perturbed Derivatives", arXiv:2005.04639, 2020.

C. Cartis, N. Gould and Ph. L. Toint,
"Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization", FoCM, vol. 18(5), pp. 1083-1107, 2018.

Also see http://perso.fundp.ac.be/˜phtoint/toint.html