Economic inexact restoration for derivative-free expensive function minimization and applications*

E. G. Birgin[†] N. Krejić[‡] J. M. Martínez[§]

May 31, 2020

Abstract

bla bla bla.

Key words: Inexact Restoration, derivative-free, inexact evaluation, expensive function, global convergence, algorithms.

2010 Mathematics Subject Classification: 65K05, 65K10, 90C30, 90C90.

1 Introduction

For different reasons scientists and engineers may need to optimize problems in which the objective function is very expensive to evaluate. In these cases, partial, and obviously inexact, evaluations are useful. The idea is to decrease as much as possible functional values using partial evaluations in such a way that, when we have no chance except to evaluate the function with maximal accuracy, we are already close enough to a solution of the problem. Rational decisions about when to increase accuracy (and evaluation cost) or even when to appeal to more inexact evaluations are hard to make. Roughly speaking we need a compromise between accuracy of evaluation and functional decrease that is difficult to achieve on a mere heuristic basis.

In recent papers, we developed a methodology based on the analogy of the Inexact Restoration idea for continuous constrained optimization and the process of increasing accuracy of function evaluation. In our most recent paper [11], we developed an algorithm inspired in [10, 40] for which it is possible to prove, not only convergence but also complexity results. However, the algorithm proposed in [11], as well as the algorithms introduced in [10, 40], employs derivatives

*This work has been partially supported by the Brazilian agencies FAPESP (grants 2013/07375-0, 2016/01860-1, and 2018/24293-0) and CNPq (grants 302538/2019-4 and 302682/2019-8) and by the Serbian Ministry of Education, Science, and Technological Development (grant 174030).

†Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. e-mail: egbirgin@ime.usp.br

‡Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia. e-mail: natasak@uns.ac.rs

§Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing (IMECC), University of Campinas, 13083-859 Campinas SP, Brazil. e-mail: martinez@ime.unicamp.br

of the objective function, a feature that may be inadequate in many cases in which one does not have derivatives at all.

This motivates the present work, in which the algorithms of [10, 11, 40] are adapted to the case in which derivatives are not available and the main theoretical results are proved. Moreover, we concentrate ourselves in a practical problem related with our present work of predicting and mitigating the consequences of dam breaking disasters.

Notation. $\mathbb{N}_+$ denotes the non-negative integer numbers; while $\mathbb{R}_+$ denotes the non-negative real numbers. The symbol $\|\cdot\|$ denotes the Euclidean norm of vector and matrices.

2 Main algorithm

{mainalgo}

For defining the main problem we adopt the formulation of [11]. Assume that Y is a set, $h : Y \to \mathbb{R}_+$, $f : \mathbb{R}^n \times Y \to \mathbb{R}$, and $\Omega \subset \mathbb{R}^n$ is defined by the set of equations $c_E(x) = 0$ and inequations $c_I(x) \le 0$, where $c_E : \mathbb{R}^n \to \mathbb{R}^m$ and $c_I : \mathbb{R}^n \to \mathbb{R}^p$. The problem is given by

$$\text{Minimize (with respect to } x\text{)}\;\; f(x,y) \text{ subject to } h(y) = 0 \text{ and } x \in \Omega, \tag{1}$$ {theproblem}

where

$$\Omega = \{x \in \mathbb{R}^n \mid c_E(x) = 0 \text{ and } c_I(x) \le 0\}. \tag{2}$$ {omega}

The main algorithm for solving this problem is also the one suggested in [11]. The merit function $\Phi : \mathbb{R}^n \times Y \times (0,1) \to \mathbb{R}$ will be defined by

$$\Phi(x,y,\theta) = \theta f(x,y) + (1-\theta)h(y).$$

Algorithm 2.1. Let $x_0 \in \Omega$, $y_0 \in Y$, $\theta_0 \in (0,1)$, $\nu > 0$, $r \in (0,1)$, $\alpha > 0$, and $\beta > 0$ be given. Set $k \leftarrow 0$.

Step 1. *Restoration phase*

Define $y_k^{\rm re} \in Y$ in such a way that

$$h(y_k^{\rm re}) \le r h(y_k) \tag{3}$$ {erre}

and

$$f(x_k, y_k^{\rm re}) \le f(x_k, y_k) + \beta h(y_k). \tag{4}$$ {beta}

Step 2. *Updating the penalty parameter*

If

$$\Phi(x_k, y_k^{\rm re}, \theta_k) \le \Phi(x_k, y_k, \theta_k) + \frac{1-r}{2}\left(h(y_k^{\rm re}) - h(y_k)\right), \tag{5}$$ {cinco}

set $\theta_{k+1} = \theta_k$. Otherwise, set

$$\theta_{k+1} = \frac{(1+r)\left(h(y_k) - h(y_k^{\rm re})\right)}{2\left(f(x_k, y_k^{\rm re}) - f(x_k, y_k) + h(y_k) - h(y_k^{\rm re})\right)}. \tag{6}$$ {seis}

Step 3. *Optimization phase*

Compute $y_{k+1} \in Y$ and $s_k \in \mathbb{R}^n$ such that $x_k + s_k \in \Omega$,

$$f(x_k + s_k, y_{k+1}) \le f(x_k, y_k^{\rm re}) - \alpha \|s_k\|^\nu \tag{7}$$ {armijo1}

and

$$\Phi(x_k + s_k, y_{k+1}, \theta_{k+1}) \le \Phi(x_k, y_k, \theta_{k+1}) + \frac{1-r}{2}\left(h(y_k^{\rm re}) - h(y_k)\right). \tag{8}$$ {dos}

Define $x_{k+1} = x_k + s_k$, update $k \leftarrow k+1$, and go to Step 1.

The main results related to this algorithm have been proved in [11] and may be summarized as follows.

{a1}

Assumption A1 *At Step 1 of Algorithm 2.1, for all $k \in \mathbb{N}_+$ it is possible to compute, in finite time, $y_k^{\rm re}$ satisfying (3) and (4).*

{lemaboludo}

Lemma 2.1 *Suppose that Assumption A1 holds. Then, Algorithm 2.1 is well defined.*

{a2}

Assumption A2 *There exist $h_{\max} > 0$ and $f_{\min} \in \mathbb{R}$ such that, for all $y \in Y$ and $x \in \Omega$ we have that $h(y) \le h_{\max}$ and $f(x, y) \ge f_{\min}$.*

{corofeas}

Theorem 2.1 *Suppose that Assumptions A1 and A2 hold. Given $\varepsilon_{\rm feas} > 0$, the number of indices k such that $h(y_k) > \varepsilon_{\rm feas}$ is bounded above by*

$$\frac{c_{\rm feas}}{\varepsilon_{\rm feas}}, \tag{9}$$ {lavarlosdiente

where $c_{\rm feas}$ only depends on x_0, y_0, r, θ_0, β, $h_{\max}$, and $f_{\min}$.

{corosk}

Theorem 2.2 *Suppose that Assumptions A1 and A2 hold. Then, the series $\sum_{k=0}^{\infty} \|s_k\|^\nu$ is convergent. Moreover, given $\varepsilon_{\rm opt} > 0$, the number of iterates k at which $\|s_k\| > \varepsilon_{\rm opt}$ is not bigger than*

$$\frac{c_{\rm opt}}{\varepsilon_{\rm opt}^\nu}, \tag{10}$$ {buenoeso}

where $c_{\rm opt}$ only depends on α, x_0, y_0, r, θ_0, β, $h_{\max}$, and $f_{\min}$.

3 Solving the subproblem without derivatives

{solvinsub}

Step 3 of Algorithm 2.1 will be implemented as in [11]. Namely, initially we test an arbitrary $y_{\rm trial} \in Y$ and $s_{\rm trial} \in \mathbb{R}^n$ such that $x_k + s_{\rm trial} \in \Omega$, and we accept $y_{k+1} = y_{\rm trial}$ and $s_k = s_{\rm trial}$ if conditions (7) and (8) are fulfilled. If this is not the case we set $y_{k+1} = y_k^{\rm re}$. In that case it is trivial to see that (7) implies (8). So, the implementability of the algorithm depends on the introduction of some method that, for arbitrary x, be able to compute s such that (7) holds with $y_{k+1} = y_k^{\rm re}$ and such that "$\|s\|$ small" imply some type of optimality for f.

In Section 2 we did not use continuity let alone differentiability of the functions f, c_E, c_I. Nevertheless we are able to prove that $h(y_k)$ and $\|s_k\|$ tend to zero. Algorithm 2.1 will be useful only if the fact of $\|s_k\|$ being small implies some kind of optimality with respect to f.

A general idea for solving the implementability problem consists of choosing an arbitrary derivative-free minimization algorithm and applying it to the minimization, with respect to s, of $f(x+s,y)+\alpha\|s\|^\nu$. Since the value of this function for $s=0$ is $f(x,y)$, it turns out that, except in the case that $s=0$ is satisfactory for the algorithm, the algorithm will find $s \neq 0$ such that $f(x+s,y)+\alpha\|s\|^\nu < f(x,y)$, so that the condition (7) will take place. Moreover, if we run the derivative-free minimization algorithm up to the fulfillment of some reasonable approximate convergence criterion, we will have that this criterion will be fulfilled by s_k, which, in turn, tends to zero. In other words, points arbitrarily close to x_k will fulfill an approximate optimality criterion that corresponds to a function which is arbitrarily close to $f(x+s,y)$. Clearly, we should be very close to our main objective, which is to discover an optimization meaning in the property $\|s_k\| \to 0$. This was the idea used in [19] that lead to the choice of GSS [39, 38] as a subalgorithm for an Inexact Restoration method for derivative-free optimization with smooth constraints.

Algorithm 3.1.

Step 1. Choose $y_{k+1} \in Y$.

Step 2. Compute $s_k \in \mathbb{R}^n$ such that $x_k + s_k \in \Omega$ and

$$f(x_k + s_k, y_{k+1}) \leq f(x_k, y_{k+1}) - \alpha\|s_k\|^\nu. \tag{11}$$ {armijo2}

Step 3. If (7) and (8) hold, **return** y_{k+1} and s_k.

Step 4. Re-define $y_{k+1} = y_k^{\rm re}$, compute $s_k \in \mathbb{R}^n$ such that $x_k + s_k \in \Omega$ and such that (11) holds, and **return** y_{k+1} and s_k.

Observe that when $y_{k+1} = y_k^{\rm re}$ the fulfillment of (11) implies trivially the fulfillment of (7) and (8). This is why the test of (7) and (8) is unnecessary at Step 4. It remains to take a decision about the way in which, given y_{k+1} and x_k, one may obtain the descent condition (11) without employing derivatives and in such a way that the value of $\|s_k\|$ is a measure of optimality.

Mimicking the approach of ([19]) we could apply a derivative-free algorithm to the minimization of $f(x_k+s,y_{k+1})+\alpha\|s\|^\nu$. In such a way, if the algorithm has the obvious property of decreasing the objective function at each iteration and sine the objective function vanishes if $s=0$, all the iterates would satisfy $f(x_k+s,y_{k+1})+\alpha\|s\|^\nu < f(x_k,y_{k+1})$. Moreover, after a reasonable number of iterations we would arrive to a point s that approximately minimizes $f(x_k+s,y_{k+1})+\alpha\|s\|^\nu$. At such a point some optimality criterion would be fulfilled approximately. Therefore, in that case, a small size of s would imply that the null increment s is approximately optimal. So, we would not be far from the desired property that $\|s_k\|$ is an optimality measure.

However, this approach is not satisfactory because it demands the whole application up to approximate convergence of an ad hoc derivative-free algorithm. Our goal is to find an approach that obtains (11) with the desired optimality property of $\|s_k\|$ using a single iteration of a suitable derivative-free algorithm.

4 Optimization phase

4.1 Model regularization

{regula}

Let $F : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^n$ is arbitrary. No conditions are imposed on F except, of course, to be defined for all $x \in \Omega$. We assume, however, that there exist a model $M(\cdot, \cdot)$, $L \geq 0$, and $p \geq 0$ such that, for all $\bar{x} \in \Omega$, $M(\bar{x}, \bar{x}) = F(\bar{x})$ and

$$F(x) \leq M(\bar{x}, x) + L\|x - \bar{x}\|^{p+1} \quad (12)$$ {lipscho}

for all $x \in \Omega$. The algorithm based on the minimization of a regularized model that gives rise to the optimization phase proposed in the current section follows.

Algorithm 4.1.1. Let $\sigma_{\min} > 0$, $\sigma_{\text{ini}}^{\text{ub}} > 0$, $\zeta > 1$, $\alpha > 0$, and $x_0 \in \Omega$ be given. Initialize $k \leftarrow 0$.

Step 1. Set $\ell \leftarrow \ell + 1$ and choose $\sigma_{k,\ell} \in [0, \sigma_{\text{ini}}^{\text{ub}}]$.

Step 2. Compute $x_{k,\ell} \in \Omega$ a minimizer of $M(x_k, x) + \sigma_{k,\ell}\|x - x_k\|^{p+1}$ subject to $x \in \Omega$.

Step 3. Test the condition

$$F(x_{k,\ell}) \leq F(x_k) - \alpha\|x_{k,\ell} - x_k\|^{p+1}. \quad (13)$$ {armijoF}

If (13) does not hold, then set $\sigma_{k,\ell+1} = \max\{\sigma_{\min}, \zeta\sigma_{k,\ell}\}$, $\ell \leftarrow \ell + 1$, and go to Step 2.

Step 3. Set $x_{k+1} = x_{k,\ell}$, $\sigma_k = \sigma_{k,\ell}$, $k \leftarrow k + 1$, and go to Step 1.

The theorem below shows Algorithm 4.1.1 is well-defined and, additionally, it gives an evaluation complexity result for each iteration.

{regula2}

Theorem 4.1 *Assume that $M(\cdot, \cdot)$, $L \geq 0$, and $p \geq 0$ are such that (12) holds and that Step 2 is well defined. Then, the kth iteration of Algorithm 4.1.1 is well defined and finishes with the fulfillment of (13) after at most $O(\log(L + \alpha))$ evaluations of F.*

Proof: By (12),

$$\begin{aligned}
F(x_{k,\ell}) &\leq M(x_k, x_{k,\ell}) + L\|x_{k,\ell} - x_k\|^{p+1} \\
&\leq M(x_k, x_{k,\ell}) + \sigma_{k,\ell}\|x_{k,\ell} - x_k\|^{p+1} - \sigma_{k,\ell}\|x_{k,\ell} - x_k\|^{p+1} + L\|x_{k,\ell} - x_k\|^{p+1} \\
&\leq M(x_k, x_k) + \sigma_{k,\ell}\|x_k - x_k\|^{p+1} + (L - \sigma_{k,\ell})\|x_{k,\ell} - x_k\|^{p+1} \\
&= F(x_k) + (L - \sigma_{k,\ell}))\|x_{k,\ell} - x_k\|^{p+1}.
\end{aligned}$$

Therefore, (13) holds if $\sigma_{k,\ell} \geq L + \alpha$ that, by construction, occurs in the worst case when $\ell \geq \log_\zeta((L + \alpha)/\sigma_{\min}) + 2$. Since F is evaluated only at points $x_{k,\ell}$ to test condition (13), this completes the proof. □

We now ellaborate on the properties of the sequence generated by the algorithm. (Este analisis, que deberia estar concluindo antes de la definicion del Algoritm 4.1.2, esta incompleto.)

{regula1}

Lemma 4.1 *If $\bar{x} \in \Omega$ is a local minimizer of $F(x)$ subject to $x \in \Omega$, $M(\cdot, \cdot)$, $L \geq 0$, and $p \geq 0$ are such that (12) holds, and $\sigma \geq L$, then $\bar{x}$ is a local minimizer of $M(\bar{x}, x) + \sigma \|x - \bar{x}\|^{p+1}$ subject to $x \in \Omega$.*

Proof: Assume, by contradiction, that there exists a sequence x_k contained in Ω that converges to $\bar{x}$ and $M(\bar{x}, x_k) + \sigma \|x_k - \bar{x}\|^{p+1} < M(\bar{x}, \bar{x}) = F(\bar{x})$ for all k. Then, since $\sigma \geq L$, $M(\bar{x}, x_k) + L\|x_k - \bar{x}\|^{p+1} < F(\bar{x})$ for all k. Therefore, by (12), $F(x_k) < F(\bar{x})$ for all k and, thus, $\bar{x}$ is not a local minimizer of $F(x)$. □

We say that $\bar{x}$ is an ε-M-stationary point of F onto Ω if $M(\cdot, \cdot)$, $L \geq 0$, and $p \geq 0$ are such that (12) holds and, for some $\sigma \leq 2L$, there exists a local minimizer $\hat{x} \in \Omega$ of $M(\bar{x}, x) + \sigma \|x - \bar{x}\|^{p+1}$ such that $\|\bar{x} - \hat{x}\| \leq \varepsilon$ and $\sigma \|\bar{x} - \hat{x}\|^{p+1} \leq \varepsilon$.

By (13), we have that, for all k, $F(x_{k+1}) \leq F(x_k) - \alpha \|x_{k+1} - x_k\|^{p+1}$. Therefore, if F is bounded below by f_{low}, we necessarily have that $\|x_{k+1} - x_k\|$ tends to zero. Moreover $\|x_{k+1} - x_k\|$ can be bigger than $\varepsilon > 0$ during at most $\lfloor \alpha^{-1}(F(x_0) - f_{\text{low}})\varepsilon^{-(p+1)} \rfloor$ iterations. This reasoning shows that small values of $\|x_{k+1} - x_k\|$ when x_{k+1} is a minimizer of $M(x_k, x) + \sigma \|x - x_k\|^{p+1}$ indeed mean something in terms of optimality. (Entiendo que aqui, en lugar de ese "means something" entra de alguna forma el Lema 4.1 y la definicion de ε-M-stationary. Es aqui, inclusive, que la necesidad de esas dos desigualdades parecidas que aparecen en la definicion de ε-M-stationary es desvendada.) We are going to use this fact in the implementation of the optimization phase of Algorithm 2.1 suggested in this section.

Assume that there exist a model $M(\cdot, \cdot, \cdot)$, $L \geq 0$, and $\nu > 0$ such that, for all $\bar{x} \in \Omega$, $M(\bar{x}, \bar{x}, y) = f(\bar{x}, y)$ and

$$f(x, y) \leq M(\bar{x}, x, y) + L\|x - \bar{x}\|^{\nu} \quad (14)$$ {lipschi}

for all $x \in \Omega$ and $y \in Y$. Assume that x_k, y_k^{re}, and y_{k+1} correspond to iteration k of Algorithm 2.1. The steps to compute s_k and to choose between accepting the given arbitrary (heuristically chossen) y_{k+1} or to re-define $y_{k+1} = y_k^{\text{re}}$ follow. If $y_{k+1} = y_k^{\text{re}}$, Steps 1–4 should be skipped.

Algorithm 4.1.2. Let $\sigma_{\min} > 0$, $\sigma_{\text{ini}}^{\text{ub}} > 0$, and $\zeta > 1$ be given constants independent of k.

Step 1. Choose $\sigma \in [0, \sigma_{\text{ini}}^{\text{ub}}]$.

Step 2. Compute x_{trial} a solution to

$$\text{Minimize } M(x_k, x, y_{k+1}) + \sigma \|x - x_k\|^{\nu} \text{ subject to } x \in \Omega. \quad (15)$$ {reg1}

Step 3. If $s_k = x_{\text{trial}} - x_k$ satisfies (7) and (8), then **return** y_{k+1} and s_k.

Step 4. Re-define $y_{k+1} = y_k^{\text{re}}$ and choose $\sigma \in [0, \sigma_{\text{ini}}^{\text{ub}}]$.

Step 5. Compute x_{trial} a solution to (15).

Step 6. If $s_k = x_{\text{trial}} - x_k$ does not satisfy (7), then set $\sigma \leftarrow \max\{\sigma_{\min}, \zeta\sigma\}$ and go to Step 5.

Step 7. **Return** y_{k+1} and s_k.

4.2 Simplex derivatives (in this section we have $\Omega = \mathbb{R}^n$)

{simplex}

As in the case of Section 4.1, let us analyze first the case of a general function $F : \mathbb{R}^n \to \mathbb{R}$. In this section, we assume that F admits Lipschitz-continuous first derivatives. Therefore, there exists $L \geq 0$ such that

$$F(x) - F(\bar{x}) \leq \nabla F(\bar{x})^T (x - \bar{x}) + L\|x - \bar{x}\|^2 \tag{16}$$ {lipscha}

for all $\bar{x}, x \in \mathbb{R}^n$. Moreover, we assume that, for all $\bar{x} \in \mathbb{R}^n$, there exists a sequence (of gradient approximations) $\{g_\ell(\bar{x})\}_{\ell=0}^\infty$ such that either

$$\lim_{\ell \to \infty} \|g_\ell(\bar{x})\| = \|\nabla F(\bar{x})\| = 0 \tag{17}$$ {limige}

or

$$\|g_\ell(\bar{x}) - \nabla F(\bar{x})\| \leq \frac{\|g_\ell(\bar{x})\|}{4} \tag{18}$$ {cuatro}

for all $\ell \geq 0$ ~~large enough~~. A sequence of gradient approximations satisfying (17,18) can be obtained using the simplex techniques described in [26].

The algorithm based on simplex derivatives that gives rise to the optimization phase proposed in the current section follows.

Algorithm 4.2.1. Let $\sigma_{\min} > 0$, $\sigma_{\rm ini}^{\rm ub} > 0$, $\zeta > 1$, $\alpha > 0$, and $x_0 \in \Omega$ be given. Initialize $k \leftarrow 0$.

Step 1. Set $\ell \leftarrow 0$ and choose $\sigma_{k,\ell} \in [0, \sigma_{\rm ini}^{\rm ub}]$.

Step 2. Compute $g_\ell(x_k)$ and set $x_{k,\ell} = x_k - \frac{1}{2\sigma_{k,\ell}} g_\ell(x_k)$.

Step 3. Test the descent condition

$$F(x_{k,\ell}) \leq F(x_k) - \alpha \|x_{k,\ell} - x_k\|^2. \tag{19}$$ {armijazo}

Step 4. If (19) does not hold, then set $\sigma_{k,\ell+1} = \max\{\sigma_{\min}, \zeta\sigma_{k,\ell}\}$, $\ell \leftarrow \ell + 1$, and go to Step 2.

Step 4. Set $x_{k+1} = x_{k,\ell}$, $\sigma_k = \sigma_{k,\ell}$, $k \leftarrow k + 1$, and go to Step 1.

Note that, by definition, $x_{k,\ell}$ is the minimizer of $g_\ell(x_k)(x - x_k) + \sigma_{k,\ell}\|x - x_k\|^2$. Therefore,

$$g_\ell(x_k)(x_{k,\ell} - x_k) + \sigma_{k,\ell}\|x_{k,\ell} - x_k\|^2 \leq 0. \tag{20}$$ {mimi}

The theorem below shows Algorithm 4.2.1 is well-defined and, in addition, it provides an evaluation complexity result for its iterations.

{simplex1}

Theorem 4.2 *The kth iteration of Algorithm 4.2.1 computes an infinite sequence $\{g_\ell(x_k)\}_{\ell=0}^\infty$ that satisfies (17) or stops satisfying (19) after at most $O(\log(L + \alpha))$ evaluations of F.*

Proof: Assume that (17) does not hold. Therefore, (18) takes place for all $\ell \geq 0$. Assume

$$\sigma_{k,\ell} \geq 2(L+\alpha). \tag{21}$$ {yallego}

Therefore,

$$\frac{L+\alpha}{4\sigma_{k,\ell}} \leq \frac{1}{8}.$$

So,

$$\frac{L+\alpha}{4\sigma_{k,\ell}}\|g_\ell(x_k)\| \leq \frac{1}{8}\|g_\ell(x_k)\|.$$

Hence,

$$\frac{L+\alpha}{4\sigma_{k,\ell}}\|g_\ell(x_k)\| \leq \frac{1}{4}\|g_\ell(x_k)\| - \frac{1}{8}\|g_\ell(x_k)\|.$$

Then, by (18),

$$\frac{L+\alpha}{4\sigma_{k,\ell}}\|g_\ell(x_k)\| \leq \frac{1}{4}\|g_\ell(x_k)\| - \frac{1}{2}\|g_\ell(x_k) - \nabla F(x_k)\|.$$

Thus,

$$\frac{L+\alpha}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 \leq \frac{1}{4}\|g_\ell(x_k)\|^2 - \frac{1}{2}\|g_\ell(x_k)\|\|g_\ell(x_k) - \nabla F(x_k)\|.$$

Therefore,

$$\frac{L+\alpha}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2 \leq \frac{1}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 - \frac{1}{2\sigma_{k,\ell}}\|g_\ell(x_k)\|\|g_\ell(x_k) - \nabla F(x_k)\|.$$

So,

$$0 \leq -\frac{L+\alpha}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2 + \frac{1}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 - \frac{1}{2\sigma_{k,\ell}}\|g_\ell(x_k)\|\|g_\ell(x_k) - \nabla F(x_k)\|.$$

Hence,

$$\frac{L+\alpha}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2 - \frac{1}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 + \frac{1}{2\sigma_{k,\ell}}\|g_\ell(x_k)\|\|g_\ell(x_k) - \nabla F(x_k)\| \leq 0.$$

Then,

$$\frac{L}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2 - \frac{1}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 + \frac{1}{2\sigma_{k,\ell}}\|g_\ell(x_k)\|\|g_\ell(x_k) - \nabla F(x_k)\| \leq -\frac{\alpha}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2.$$

Thus, by Cauchy inequality,

$$\frac{L}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2 - \frac{1}{4\sigma_{k,\ell}}\|g_\ell(x_k)\|^2 + \frac{1}{2\sigma_{k,\ell}}g_\ell(x_k)^T(g_\ell(x_k) - \nabla F(x_k)) \leq -\frac{\alpha}{4\sigma_{k,\ell}^2}\|g_\ell(x_k)\|^2.$$

Therefore, by the definition of $x_{k,\ell}$,

$$L\|x_{k,\ell} - x_k\|^2 - \sigma_{k,\ell}\|x_{k,\ell} - x_k\|^2 - (x_{k,\ell} - x_k)^T(g_\ell(x_k) - \nabla F(x_k)) \leq -\alpha\|x_{k,\ell} - x_k\|^2.$$

So, by (20),

$$g_\ell(x_k)^T(x_{k,\ell}-x_k)+\sigma_{k,\ell}\|x_{k,\ell}-x_k\|^2+L\|x_{k,\ell}-x_k\|^2-\sigma_{k,\ell}\|x_{k,\ell}-x_k\|^2 \\ -(x_{k,\ell}-x_k)^T(g_\ell(x_k)-\nabla F(x_k)) \le -\alpha\|x_{k,\ell}-x_k\|^2.$$

Hence,

$$L\|x_{k,\ell}-x_k\|^2+(x_{k,\ell}-x_k)^T\nabla F(x_k) \le -\alpha\|x_{k,\ell}-x_k\|^2.$$

Then, by (16),

$$F(x_{k,\ell})-F(x_k) \le -\alpha\|x_{k,\ell}-x_k\|^2.$$

By construction, (21) occurs in the worst case when $\ell \ge \log_\zeta((L+\alpha)/\sigma_{\min})+2$. Since F is evaluated only at points $x_{k,\ell}$ to test (19), this completes the proof. □

Here we should say something about the (optimality) propoerties of the sequence generated by Algorithm 4.2.1.

We now describe the proposal based on simplex derivatives to implement to optimization phase of Algorithm 2.1. This proposal applies to the case $\nu = 2$ only. Assume that x_k, $y_k^{\rm re}$, and y_{k+1} correspond to iteration k of Algorithm 2.1. Assume that, for all $y \in Y$, defining $F(x) = f(x, y)$, we have that (a) condition (16) is satisfied and, (b) it can be defined sequences of approximate gradients $\{g_\ell(x_k)\}_{\ell=0}^\infty$ satisfying (17) or (18). (Is it clear that (b) must hold for all $y \in Y$?) Steps to compute s_k and to choose between accepting the given arbitrary (heuristically chossen) y_{k+1} or to re-define $y_{k+1} = y_k^{\rm re}$ follow. If $y_{k+1} = y_k^{\rm re}$, Steps 1–5 should be skipped.

Algorithm 4.2.2. Let $\sigma_{\min} > 0$, $\sigma_{\rm ini}^{\rm ub} > 0$, and $\zeta > 1$ be given constants independent of k. Let $\varepsilon > 0$ be given. (Could ε depend on k?).

Step 1. Set $\ell \leftarrow 0$ and choose $\sigma \in [0, \sigma_{\rm ini}^{\rm ub}]$.

Step 2. Compute $g_\ell(x_k)$. If $\|g_\ell(x_k)\| \le \varepsilon$, then go to Step 5.

Step 3. If $s_k = -\frac{1}{2\sigma} g_\ell(x_k)$ satisfies (7) and (8), then **return** y_{k+1} and s_k.

Step 4. Set $\sigma \leftarrow \max\{\sigma_{\min}, \zeta\sigma\}$, $\ell \leftarrow \ell + 1$, and go to Step 2.

Step 5. Re-define $y_{k+1} = y_k^{\rm re}$.

Step 6. Set $\ell \leftarrow 0$ and choose $\sigma \in [0, \sigma_{\rm ini}^{\rm ub}]$.

Step 7. Compute $g_\ell(x_k)$. If $\|g_\ell(x_k)\| \le \varepsilon$ then **return** y_{k+1} and s_k. (Who is s_k in this case? Can we return $s_k = 0$ or does it make sense to return $s_k = -\frac{1}{2\sigma} g_\ell(x_k)$?)

Step 8. If $s_k = -\frac{1}{2\sigma} g_\ell(x_k)$ satisfies (7), then **return** y_{k+1} and s_k.

Step 9. Set $\sigma \leftarrow \max\{\sigma_{\min}, \zeta\sigma\}$, $\ell \leftarrow \ell + 1$, and go to Step 7.

Remark. At Step 2, $g_\ell(x_k)$ corresponds to a simplex gradient of function $F(\cdot) = f(\cdot, y_{k+1})$ evaluated at x_k, with $y_{k+1} \ne y_k^{\rm re}$; while, at Step 7, it corresponds to a simplex gradient of function $F(\cdot) = f(\cdot, y_k^{\rm re})$.

4.3 Dense directions (in this section we have $\Omega = \mathbb{R}^n$) {dense}

For fixing ideas let us begin with the problem of minimizing $F : \mathbb{R}^n \to \mathbb{R}$.

Algorithm 4.3.1. Let $\Delta > 0$, $\{d_0, d_1, d_2, \dots\}$ a dense set in the unitary sphere, and $x_0 \in \mathbb{R}^n$ be given. Initialize $k \leftarrow 0$.

Step 1. Set $\ell \leftarrow 0$.

Step 2. Test condition

$$F(x_k + 2^{-\ell}\Delta d_{k+\ell}) \le F(x_k) - \alpha 2^{-\ell}\Delta. \quad (22)$$ {armijo6}

Step 3. If (22) does not hold, then set $\ell \leftarrow \ell + 1$ and go to Step 2.

Step 4. Set $s_k = 2^{-\ell}\Delta d_{k+\ell}$, $x_{k+1} = x_k + s_k$, $k \leftarrow k + 1$ and go to Step 1.

Clearly, there are two possibilities. Either every iteration of this algorithm finishes with the fulfillment of (22) for some ℓ or there exists k such that for all $\ell = 0, 1, 2, \dots$ (22) fails to be satisfied. In this case we say that the algorithm stalls at x_k. For proving the characterization of this situation we need a Lipschitz assumption that is stated below.

{assumpdense1}

Assumption A3 *There exists $L > 0$ such that, whenever k is an iteration at which Algorithm 4.3.1 stalls, there exists a neighborhood of x_k such that, for all z_1, z_2 in this neighborhood,*

$$|F(z_1) - F(z_2)| \le L\|z_1 - z_2\|. \quad (23)$$ {lipschu}

{assumpdense2}

Assumption A4 *F is bounded below.*

Here, it would be nice to give an interpretation of the results in the theorem below, that are not clear to me. Are (24) and (25) saying, respectively, that x_* and x_k satisfy a derivative-free optimality condition? If the answer is 'yes', it is in some sense umpleasent that the tolerance is α instead of ε.

{teodense1}

Theorem 4.3 *Suppose that Assumptions A3 and A4 hold. Then:*

1. *If Algorithm 4.3.1 generates an infinite sequence of iterates $\{x_k\}$, the series $\sum_{k=0}^{\infty} \|s_k\|$ is convergent and the sequence $\{x_k\}$ converges to a point x_* such that for any given unitary direction d and any given $\ell \in \{0, 1, 2, \ldots\}$, we have that*

$$\frac{F(x_* + 2^{-\ell}\Delta d) - F(x_*)}{2^{-\ell}\Delta} \ge -\alpha. \quad (24)$$ {tesilin}

2. *If the algorithm stalls at x_k, for all d in the unitary sphere, there exist infinitely many indices $\ell \in \{0, 1, 2, \dots\}$ verifying*

$$\frac{F(x_k + 2^{-\ell}\Delta d) - F(x_k)}{2^{-\ell}\Delta} > -2\alpha. \quad (25)$$ {saba}

Proof: Consider first the case in which Algorithm 4.3.1 generates an infinite sequence $\{x_k\}$. By (22) and the definition of s_k,

$$F(x_k + s_k) \le F(x_k) - \alpha \|s_k\| \qquad (26) \quad \{armijo7\}$$

and $x_{k+1} = x_k + s_k$ for all $k = 0, 1, 2, \dots$. Since F is assumed to be bounded below, (26) implies that the series $\sum_{k=0}^{\infty} \|s_k\|$ is convergent. Thus $\{x_k\}$ is a Cauchy sequence and converges to some $x_* \in \mathbb{R}^n$.

Let us fix $\ell \in \{0, 1, 2, \dots\}$ and let d be an arbitrary direction belonging to the unitary sphere. Since $\|s_k\| \to 0$, there exists k_0 such that for all $k \ge k_0$, $\|s_k\| < 2^{-\ell}\Delta$. This implies that, for all $k \ge k_0$ the increment $2^{-\ell}\Delta d_{k+\ell}$ has been tested and rejected, not satisfying (22). Therefore,

$$F(x_k + 2^{-\ell}\Delta d_{k+\ell}) > F(x_k) - \alpha 2^{-\ell}\Delta \qquad (27) \quad \{endelta\}$$

for all $k \ge k_0$. Since $\{d_{k+\ell}, k \ge k_0\}$ is dense in the unitary sphere it turns out that there exists an infinite set of indices K such that $\lim_{k\in K} d_{k+\ell} = d$. Taking limits in (27) for $k \in K$ we obtain that

$$F(x_* + 2^{-\ell}\Delta d) \ge F(x_*) - \alpha 2^{-\ell}\Delta. \qquad (28) \quad \{enellimi\}$$

Consider now the case in which the algorithm finishes at iteration k with an infinite loop between Step 2 and Step 3. Observe that the set $\{d_{k+\ell}, \ell = 0, 1, 2, \dots\}$ is dense in the unitary sphere, as well as $\{d_0, d_1, d_2, \dots\}$. Let d be an arbitrary direction in the unitary sphere. By the density of $\{d_{k+\ell}, \ell = 0, 1, 2, \dots\}$ there exists an infinite sequence of indices $\ell \in K_1$ such that

$$\lim_{\ell \in K_1} d_{k+\ell} = d.$$

Since (22) does not hold for all $\ell = 0, 1, 2, \dots$ we have that

$$F(x_k + 2^{-\ell}\Delta d_{k+\ell}) > F(x_k) - \alpha 2^{-\ell}\Delta \qquad (29) \quad \{notarmijo6\}$$

for all $\ell \in K_1$. But, by the Lipschitz condition (23),

$$|F(x_k + 2^{-\ell}\Delta d_{k+\ell}) - F(x_k + 2^{-\ell}\Delta d)| \le L 2^{-\ell}\Delta \|d_{k+\ell} - d\|$$

for all $\ell \in K_1$ large enough. Therefore,

$$F(x_k + 2^{-\ell}\Delta d - F(x_k + 2^{-\ell}\Delta d_{k+\ell}) \ge -L 2^{-\ell}\Delta \|d_{k+\ell} - d\|$$

for all $\ell \in K_1$ large enough. So,

$$F(x_k + 2^{-\ell}\Delta d) \ge F(x_k + 2^{-\ell}\Delta d_{k+\ell}) - L 2^{-\ell}\Delta \|d_{k+\ell} - d\|$$

for all $\ell \in K_1$ large enough. Thus, by (29),

$$F(x_k + 2^{-\ell}\Delta d) > F(x_k) - \alpha 2^{-\ell}\Delta - L 2^{-\ell}\Delta \|d_{k+\ell} - d\| \qquad (30) \quad \{sabanda\}$$

for all $\ell \in K_1$ large enough. Let ℓ_0 be such that

$$L\|d_{k+\ell} - d\| \le \alpha$$

for all $\ell \geq \ell_0$, $\ell \in K_1$. Then

$$-L\|d_{k,\ell} - d\| \geq -\alpha$$

for all $\ell \geq \ell_0$, $\ell \in K_1$. Thus, by (30),

$$F(x_k + 2^{-\ell}\Delta d) > F(x_k) - \alpha 2^{-\ell}\Delta - \alpha 2^{-\ell}\Delta = F(x_k) - 2\alpha 2^{-\ell}\Delta \quad (31)$$ {saban}

for all $\ell \in K_1$ large enough. Therefore, (25) is proved. □

We now describe the proposal based on dense directions to implement to optimization phase of Algorithm 2.1. This proposal applies to the case $\nu = 1$ only. Assume that x_k, $y_k^{\rm re}$, and y_{k+1} correspond to iteration k of Algorithm 2.1. Assume that, for all $y \in Y$, condition (23) is satisfied defining $F(x) = f(x, y)$. Steps to compute s_k and to choose between accepting the given arbitrary (heuristically chossen) y_{k+1} or to re-define $y_{k+1} = y_k^{\rm re}$ follow. If $y_{k+1} = y_k^{\rm re}$, Steps 1–4 should be skipped.

Algorithm 4.3.2. Let $\Delta > 0$ and $\{d_0, d_1, d_2, \dots\}$ a dense set in the unitary sphere be given, independent of k.

Step 1. Set $\ell \leftarrow 0$.

Step 2. If $s_k = 2^{-\ell}\Delta d_{k+\ell}$ satisfies (7) and (8), then **return** y_{k+1} and s_k.

Step 3. If $\ell < 2k$, then set $\ell \leftarrow \ell + 1$ and go to Step 2.

Step 4. Re-define $y_{k+1} = y_k^{\rm re}$.

Step 5. Set $\ell \leftarrow 0$.

Step 6. If $s_k = 2^{-\ell}\Delta d_{k+\ell}$ satisfies (7), then **return** y_{k+1} and s_k.

Step 7. If $\ell < 2k$, then set $\ell \leftarrow \ell + 1$ and go to Step 6.

Step 8. **Return** y_{k+1} and $s_k = 0$.

4.4 Subgradient approach(in this section we have $\Omega = \mathbb{R}^n$)

{bagirov}

The full script of every previous subsection for the optimization phase is the following. We first introduce a derivative-free algorithm that applyes to a generic F (locally Lipschitz in this case). The we prove the algorithm is well-defined and prove some some convergence result that says something about optimality. The we define $F(x) = f(x, y)$, extend the assumtions and make a proposal for an iteration of the optimization phase. OK. In this ection, we only have the last part, i.e., the proposal for the optimization phase. All the other parts are missing.

Based in the approach of [3] ...

We now describe the proposal based on subgradients to implement to optimization phase of Algorithm 2.1. This proposal applies to the case $\nu = 1$ only. Assume that x_k, $y_k^{\rm re}$, and y_{k+1} correspond to iteration k of Algorithm 2.1. Assume that, for all $y \in Y$, $F(x) = f(x, y)$ is locally

Lipschitz. Steps to compute s_k and to choose between accepting the given arbitrary (heuristically chossen) y_{k+1} or to re-define $y_{k+1} = y_k^{\rm re}$ follow. If $y_{k+1} = y_k^{\rm re}$, Steps 1–4 should be skipped.

Algorithm 4.4.1. Let $\gamma > 0$ and $\delta > 0$ be given (small) constants, independent of k. Moreover, let $\kappa \in (0, 1)$ be a given constant, independent of k. Constants κ and δ must satisfy $\kappa\delta \geq \alpha$, where $\alpha > 0$ is the parameter of Algorithm 2.1. This means, that, in fact, we must require $\delta > \alpha$ and choose $\kappa \in [\alpha/\delta, 1)$. (We can re-write this simplified requirement if we confirm all this is right.) Initialize $\ell \leftarrow 1$.

Step 1. Choose $\bar{d} \in \mathbb{R}^n$ such that $\|\bar{d}\| = 1$.

Step 2. Compute a quasisecant vector $\bar{v} \in \mathbb{R}^n$ using x_k, γ, d, and the function (of x) $f(x, y_{k+1})$. This is not exactly Bagirov. Given the direction $\bar{d}$, we should use his algorithm below to compute the descent direction s_k and the corresponding quasisecant v_k.

Step 3. If $\|\bar{v}\| > \delta$ and $s_k = -\frac{\gamma}{\|\bar{v}\|}\bar{v}$ satisfies (7) and (8), then **return** y_{k+1} and s_k. (There is a difference here with respect to the previous version. In the previous version we required this first s_k to satisfy (32). Do we need this? There is a relation between satisfying (32) and satisfying (**??**). We should check this latter.

Step 4. Re-define $y_{k+1} = y_k^{\rm re}$, choose $\bar{d}_0 \in \mathbb{R}^n$ such that $\|\bar{d}_0\| = 1$, and set $\ell \leftarrow 1$.

Step 5. Compute a quasisecant vector $v_\ell \in \mathbb{R}^n$ using x_k, γ, $\bar{d}_{\ell-1}$, and the function $f(x, y_{k+1})$. This is not exactly Bagirov. Given the direction $\bar{d}$, we should use his algorithm below to compute the descent direction s_k and the corresponding quasisecant v_k.

Step 6. If $\ell = 1$ then define $\bar{v}_\ell = v_\ell$. Otherwise, define $\bar{v}_\ell$ as the point in the segment $[v_\ell, \bar{v}_{\ell-1}]$ that is closest to the origin in the Euclidian norm.

Step 7. If $\|\bar{v}_\ell\| \leq \delta$ then **return** y_{k+1} and $s_k = 0$.

Step 8. Compute $\bar{d}_\ell = -\bar{v}_\ell/\|\bar{v}_\ell\|$.

Step 9. If

$$f(x_k + \gamma\bar{d}_\ell, y_{k+1}) \leq f(x_k, y_{k+1}) - \kappa\gamma\|\bar{v}_\ell\| \quad (32)$$

{armijobagirov}

then **return** y_{k+1} and $s_k = \gamma\bar{d}_\ell$.

Step 10. Set $\ell \leftarrow \ell + 1$, and go to Step 5.

4.5 Gradient sampling

{gs}

The optimization algorithm presented below is based on [37]. We present the algorithm with nonormalized search direction, other options are possible, see [37]. **Maybe we can work with limited Armijo line search, procedure 4.3 in the original paper?** Assume that $f(x, y)$, for all $y \in Y$ is locally Lipschitz and continuously differentiable on an open and dense set D.

Algorithm 4.5.1 Let $\alpha > 0$ and $\delta, \varepsilon > 0$ be given (small) constants independent of k. Let $\beta, \gamma \in (0, 1)$ be line search parameters, reduction factors $\eta, \theta \in (0, 1]$. Let $m \geq n + 1$ be the sampling size. Assume that $\delta_k \geq \delta, \varepsilon_k \geq \varepsilon$ and $x_k \in D$ are given.

Step 1. Choose y_{k+1} and compute $g_k = \nabla f(x_k, y_{k+1})$.

Step 2. If $\|g_k\| \le \delta$ and $\varepsilon_k \le \varepsilon$ stop.

Step 3. If $s_k = -\gamma g_k$ satisfies (7) and (8) and $x_k + s_k \in D$ then **return** y_{k+1} and s_k.

Step 4. Redefine $y_{k+1} = y_k^{\mathrm{re}}$. Denote $f(x) = f(x, y_{k+1})$.

Step 5. Sample $\{x_{ki}\}_{i=1}^m$ uniformly and independently from $B(x_k, \varepsilon_k)$. If $\{x_{ki}\}_{i=1}^m \not\subset D$, then stop. Otherwise set $G_k = co\{\nabla f(x_k), \nabla f(x_{k1}, \ldots, \nabla f(x_{km})\}$. Find

$$g_k = \arg\min_{g \in G_k} \|g\| \tag{33}$$ {mingk}

Step 6. If $\|g_k\| \le \delta,\ \varepsilon_k \le \varepsilon$ stop.

Step 7. If $\|g_k\| \le \delta_k$ set $\delta_k = \theta\delta_k,\ \varepsilon_k = \eta\varepsilon_k$, and goto Step 5. Otherwise, set $\delta_{k+1} = \delta_k,\ \varepsilon_{k+1} = \varepsilon_k$ and $d_k = -g_k$.

Step 8. Line search: Find the smallest $j = 0, 1, \ldots$ such that for $\gamma_k = \beta^j$ there holds

$$f(x_k + \gamma_k d_k) < f(x_k) - \alpha\gamma_k \|g_k\|^2. \tag{34}$$ {gsarmijo}

Step 9. If $x_k + \gamma_k d_k \in D$ set $x_{k+1} = x_k + \gamma_k d_k$. Otherwise let $x_{k+1} \in D$ be any point such that

$$f(x_{k+1}) < f(x_k) - \alpha\gamma_k \|g_k\|^2 \tag{35}$$ {notD1}

and

$$\|x_k + \gamma_k d_k - x_{k+1}\| \le \min\{\gamma_k, \varepsilon_k\} \|d_k\|. \tag{36}$$ {notD2}

Return $s_k = x_{k+1} - x_k$ and y_{k+1}.

5 Numerical experiments

6 Conclusions

References

[1] R. Andreani, S. L. Castro, J. L. Chela, A. Friedlander, and S. A. Santos, An inexact-restoration method for nonlinear bilevel programming problems, *Computational Optimization and Applications* 43, pp. 307–328, 2009.

[2] A. Asimakopulos, *Keynes's General Theory and Accumulation*, Cambridge University Press, 1991.

[3] A. M. Bagirov, L. Jin, N. Karmitsa, A. AI. Nuaimati, and N. Sultanova Subgradient method for nonconvex nonsmooth optimization, *Journal of Optimization Theory and Applications* 157, pp. 416–435, 2013.

[4] N. Banihashemi and C. Y. Kaya, Inexact Restoration for Euler discretization of box-constrained optimal control problems, *Journal of Optimization Theory and Applications* 156, pp. 726–760, 2013.

[5] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, Princeton, 2009.

[6] E. G. Birgin, L. F. Bueno, and J. M. Martínez, Assessing the reliability of general-purpose Inexact Restoration methods, *Journal of Computational and Applied Mathematics* 282, pp. 1–16, 2015.

[7] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, and S. A. Santos, On the use of third-order models with fourth-order regularization for unconstrained optimization, *Optimization Letters* 14, pp. 815–838, 2020.

[8] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models, *Mathematical Programming* 163, pp. 359–368, 2017.

[9] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models, *SIAM Journal on Optimization* 26, pp. 951–967, 2016.

[10] E. G. Birgin, N. Krejić, and J. M. Martínez, On the employment of Inexact Restoration for the minimization of functions whose evaluation is subject to errors, *Mathematics of Computation* 87, pp. 1307–1326, 2018.

[11] E. G. Birgin, N. Krejić, J. M. Martínez, Iteration and evaluation complexity on the minimization of functions whose computation is intrinsically inexact, *Mathematics of Computation* 89, pp. 253–278, 2020.

[12] E. G. Birgin and J. M. Martínez, The use of quadratic regularization with cubic descent for unconstrained optimization, *SIAM Journal on Optimization* 27, pp. 1049–1074, 2017.

[13] E. G. Birgin and J. M. Martínez, On regularization and active-set methods with complexity for constrained optimization, *SIAM Journal on Optimization* 28, pp.1367–1395, 2018.

[14] E. G. Birgin and J. M. Martínez, A Newton-like method with mixed factorizations and cubic regularization for unconstrained minimization, Technical Report MCDO25062018, University of São Paulo, Brazil, 2018.

[15] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.

[16] E. G. Birgin, J. M. Martínez, and M. Raydan, Spectral Projected Gradient Methods: Review and Perspectives, *Journal of Statistical Software* 60, issue 3, 2014.

[17] E. G. Birgin, N. Krejić, and J. M. Martínez, On the employment of Inexact Restoration for the minimization of functions whose evaluation is subject to errors, *Mathematics of Computation* 87, pp. 1307–1326, 2018.

[18] E. G. Birgin and J. M. Martínez, Local convergence of an Inexact Restoration method and numerical experiments, *Journal of Optimization Theory and Applications* 127, pp. 229–247, 2005

[19] L. F. Bueno, A. Friedlander, J. M. Martínez, and F. N. C. Sobral, Inexact Restoration method for derivative-free optimization with smooth constraints, *SIAM Journal on Optimization* 23, pp. 1189–1231, 2013.

[20] C. Cartis, N. I. M. Gould, and Ph. L. Toint, On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization, *SIAM Journal on Optimization* 20, pp. 2833–2852, 2010.

[21] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Adaptive cubic regularization methods for unconstrained optimization. Part I: motivation motivation, convergence and numerical results, *Mathematical Programming* 127, pp. 245–295, 2011.

[22] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function and derivative complexity, *Mathematical Programming* 130, pp. 295–319, 2011.

[23] C. Cartis, N. I. M. Gould, and Ph. L. Toint, Universal regularization methods - varying the power, the smoothness and the accuracy, Preprint RAL-P-2016-010, Rutherford Appleton Laboratory, Chilton, England, 2016.

[24] B. H. Cervelin, D. Conti, M. A. Diniz-Ehrhardt, and J. M. Martínez, A computer model for particle-like simulation in broiler houses, *Computers and Electronics in Agriculture* 141, pp. 1–14, 2017.

[25] F. E. Curtis, D. P. Robinson, and M. Samadi, A trust-region algorithm with a worst-case iteration complexity of $O(\varepsilon^{-3/2})$, *Mathematical Programming* 162, pp. 1–32, 2017.

[26] A. R. Conn, Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[27] S. Dempe, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht, 2002.

[28] J. P. Dussault, ARCq: a new adaptive regularization by cubics, *Optimization Methods and Software* 33, pp. 322–335, 2018.

[29] Y. Ermoliev, *Stochastic Programming Methods*, Academy of Sciences of the USSR, Kiev, 1967.

[30] A. Fischer and A. Friedlander, A new line search Inexact Restoration approach for nonlinear programming, *Computational Optimization and Applications* 46, pp. 333–346, 2010.

[31] C. C. Gonzaga, E. W. Karas, and M. Vanti, A globally convergent filter method for nonlinear programming, *SIAM Journal on Optimization* 14, pp. 646–669, 2004.

[32] G. N. Grapiglia and Yu. Nesterov, Globally convergent second-order schemes for minimizing twice differentiable functions, CORE Discussion Paper 2016/28, Université Catholique de Louvain, Louvain, Belgium, 2016.

[33] G. N. Grapiglia, J-Y Yuan, and Y-X Yuan, On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization, *Mathematical Programming* 152, pp. 491–520, 2015.

[34] A. Griewank, *The modification of Newton's method for unconstrained optimization by bounding cubic terms*, Technical Report NA/12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 1981.

[35] A. O. Herrera, H. D. Scolnik, G. Chichilnisky, G. C. Gallopin, and J. E. Hardoy, *Catastrophe or New Society? A Latin America world model*, IDRC, Ottawa, ON, CA, 1976.

[36] C. Y. Kaya and J. M. Martínez, Euler discretization and Inexact Restoration for optimal control, *Journal of Optimization Theory and Applications* 134, pp. 191–206, 2007.

[37] Kiwiel, K.C., Convergence of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization, SIAM J. Optim. 18,2 (2007), 379-388.

[38] T. G. Kolda, R. M. Lewis, and V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, *SIAM Review* 45, pp. 385–482, 2003.

[39] T. G. Kolda, R. M. Lewis, and V. Torczon, Stationarity results for generating set search for linearly constrained optimization, *SIAM Journal on Optimization* 17, pp. 943–968, 2006.

[40] N. Krejić and J. M. Martínez, Inexact Restoration approach for minimization with inexact evaluation of the objective function, *Mathematics of Computation* 85, pp. 1775–1791, 2016.

[41] Z. Luo, J. Pang, and D. Ralph, *Mathematical Programs with equilibrium constraints*, Cambridge University Press, Cambridge, 1996.

[42] J. M. Martínez, Local minimizers of quadratic functions under spherical and ball constraints, *SIAM Journal on Optimization* 4, pp. 159–176, 1994.

[43] J. M. Martínez, Inexact Restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming, *Journal of Optimization Theory and Applications* 111, pp. 39–58, 2001.

[44] J. M. Martínez, On high-order model regularization for constrained optimization, *SIAM Journal on Optimization* 27, pp. 2447–2458, 2017.

[45] J. M. Martínez and E. A. Pilotta, Inexact restoration algorithms for constrained optimization, *Journal of Optimization Theory and Applications* 104, pp. 135–163, 2000.

[46] J. M. Martínez and M. Raydan, Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization, *Journal of Global Optimization* 68, pp. 367–385, 2017.

[47] Y. Nesterov and B. T. Polyak, Cubic regularization of Newton's method and its global performance, *Mathematical Programming* 108, pp. 177–205, 2006.

[48] R. Pasupathy, On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization, *Operations Research* 58, pp. 889–901, 2010.

[49] E. Polak, J. O. Royset, Eficient sample sizes in stochastic nonlinear programing, *Journal of Computational and Applied Mathematics* 217, pp. 301–310, 2008.

[50] J. O. Royset, Optimality functions in stochastic programming, *Mathematical Programming* 135, pp. 293–321, 2012.

[51] J. O. Royset and R. Szechtman, Optimal budget allocation for Sample Average Approximation, *Operations Research* 61, pp. 762–776, 2013.

[52] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, Wiley, New York, 1993.