MATHEMATICS OF COMPUTATION
Volume 87, Number 311, May 2018, Pages 1307–1326
http://dx.doi.org/10.1090/mcom/3246
Article electronically published on September 19, 2017

ON THE EMPLOYMENT OF INEXACT RESTORATION FOR THE MINIMIZATION OF FUNCTIONS WHOSE EVALUATION IS SUBJECT TO ERRORS

E. G. BIRGIN, N. KREJIĆ, AND J. M. MARTÍNEZ

Abstract. Inexact Restoration is a well established technique for continuous minimization problems with constraints. Recently, it has been used by Krejić and Martínez for optimization of functions whose evaluation is necessarily inexact and comes from an iterative process. This technique will be generalized in the present paper and it will be applied to stochastic optimization and related problems. New convergence results will be given and numerical results will be presented.

1. Introduction

We will consider the problem

$$\text{Minimize } f(x) \text{ subject to } x \in \Omega, \tag{1}$$

where $\Omega \subseteq \mathbb{R}^n$ is nonempty, closed, and convex. We assume that, for different reasons, the exact evaluation of $f(x)$ is not available. Practical situations in which the function that one wants to minimize cannot be evaluated with high precision are frequent. Strictly speaking, one never evaluates exactly a function because we are constrained to use floating point arithmetic. Since evaluation procedures using multiple-precision arithmetic are available, it raises the problem of deciding the accuracy with which one should compute functions and derivatives in order to maximize efficiency. In other cases the computation of the objective function comes from an iterative method whose convergence properties are not always well known. In these cases different measures of accuracy need to be developed; for example, accuracy may be defined as a nonincreasing function on the number of iterations used by the iterative method that evaluates the objective function. More frequently, the objective function is the result of a simulation and the exact functional value depends on the the expectation of a random variable or other statistical parameters. For example, in modern risk-oriented decisions one optimizes functions that involve VaR, CVaR, or other risks measures that may be interpreted in terms of percentiles. Sometimes, simulations include physical experiments. For example, the response of production processes to operation parameters may involve experiments

Received by the editor March 28, 2016, and, in revised form, November 24, 2016.
2010 *Mathematics Subject Classification.* Primary 65K05, 65K10, 90C30, 90C90.
Key words and phrases. Inexact Restoration, stochastic programming, global convergence, numerical experiments.
This work was partially supported by the Brazilian agencies FAPESP (grants 2010/10133-0, 2013/03447-6, 2013/05475-7, 2013/07375-0, and 2014/18711-3) and CNPq (grants 309517/2014-1 and 303750/2014-6) and by the Serbian Ministry of Education, Science, and Technological Development (grant 174030).

with random variations of other parameters whose impact we do not want to evaluate. Finally, macroeconomic decisions usually involve running different models, that should be combined in a proper way in order to generate acceptable predictions. In other cases, the evaluation of the response to decisions could involve increasing research, expert interviews, bibliographic search, and data mining procedures that could greatly increase the cost of evaluations.

The key question in any of these situations is: To which extent do we need to increase the effort to obtain reliable functional evaluations? Common sense indicates that it is not worthwhile to invest a lot of work trying to obtain accurate estimations of the objective function if little work is necessary to decide that the trial set of decisions is far from being optimal. For example, precision on the combination of drugs for a medical treatment is important if the combination promises to be the correct one, but not when one is clearly far away of being in the good direction. On the other hand, the value of a given experiment that leads to an approximate evaluation increases if such experiment has to be performed with rigorous and accepted rules of sampling. The idea of this work, as well as the preliminary one [18] is to trace a parallel between the pair accuracy-value of the function that we want to optimize and the pair feasibility-optimality of constrained optimization. Namely, we aim to optimize a function f subject to the constraint that the evaluation of this function is exact. But, since complete exactness is not possible, we define higher or lower arbitrary levels of inexactness based on the (computational or human) effort that is employed for improving evaluation accuracy.

In Stochastic Optimization the objective function uses to be (inexactly) evaluated employing a sample $\{\xi^1, \ldots, \xi^N\}$ of the random variable under consideration. As a consequence, the evaluation of $f(x)$ is always inexact, and precision is increased when the size N of the sample grows. For example, suppose that $\xi_1, \ldots, \xi_n$ denote the future prices of n assets and that x_j is the fraction of the budget invested on asset j. (In this case we need the constraints $\sum_{j=1}^n x_j = 1$ and $x \geq 0$.) Then the return is a random variable given by $g(x, \xi) = \sum_{j=1}^n \xi_j x_j$ and it is natural to maximize its expectation under additional suitable constraints. The Sample Path Optimization or Sample Average Approximation (SAA) method (see, for example, [23]) applies to optimization problems for which the objective function cannot be computed exactly (or is unaffordable) but can be estimated by simulation. Given N and an independent and identically distributed (i.i.d.) sample $\xi^1, \xi^2, \ldots, \xi^N$ of N replications of the random vector ξ, one considers the function

$$f_N(x) \equiv \frac{1}{N} \sum_{i=1}^{N} g(x, \xi^i)$$

and the problem given by

$$\text{Minimize}_{\ x \in \Omega}\ f_N(x) \tag{2}$$

is approximately solved. Problem (2) is deterministic and, in principle, any standard nonlinear programming solver may be employed for its solution. Assume that we are able to solve (2) globally up to a tolerance ε_{glo}, i.e., we can compute $x_N^*(\varepsilon_{\text{glo}})$ such that $f_N(x_N^*(\varepsilon_{\text{glo}})) \leq f_N(x) + \varepsilon_{\text{glo}}$, for all $x \in \Omega$. Let $\varepsilon_{\text{prob}} \in (0, 1)$ and consider the fulfilment of

$$|f_N(x_N^*(\varepsilon_{\text{glo}})) - f^*| \leq \varepsilon_{\text{glo}} \text{ with probability at least } 1 - \varepsilon_{\text{prob}}, \tag{3}$$

where f^* is the optimal value of problem (1). Then, the decision to be made in the SAA method is to determine the value of N for which solving a single problem of the form (2) guarantees that (3) will hold. A conservative answer to this question, thus yielding to a potentially large value of N and, in consequence, a costly SAA problem (2), is given in [24]. Usually, an infinite sequence of SAA problems with sample size N_k and optimality tolerance $(\varepsilon_{\text{glo}})_k$, such that $N_k \to \infty$ and $(\varepsilon_{\text{glo}})_k \to 0$, needs to be solved (see, for example, [25]). The sequence $\{N_k\}_{k=1}^{\infty}$ is known as *schedule*. Many papers in the literature deal with the theoretical aspects and practical issues (such as convergence, convergence rates, and efficiency) associated with the determination of a schedule and the properties of the optimization method used to solve the SAA optimization problems. See, for example, [3, 4, 14, 16–18, 21, 22] and the references therein.

Inexact Restoration (IR) [1, 2, 5, 6, 9, 11, 12, 15, 19, 20] is a paradigm for nonlinear programming that have been proved to be effective in solving constrained optimization problems in which some structure of the feasible set induces a natural way of recovering feasibility. In this work, an approach based on Inexact Restoration for tackling problem (1) will be introduced. Namely, each iteration starts with a point whose performance (function evaluation) has been computed with certain degree of accuracy. The first procedure is to try a (probably moderate) increase on the accuracy, which corresponds to the "Feasibility Phase" of Inexact Restoration. Later (in the so-called "Optimality Phase"), we try promising new variables and we evaluate performance being tolerant with respect to computational effort. If a (merit) combination of performance and accuracy is not satisfactory, the new trial point is rejected and a new one is tried closer to the one on which we started the iteration. Otherwise, the trial parameters are accepted and define a new iteration. Although the procedure roughly described above could be considered a, perhaps successful, heuristic, we felt the necessity of devising a mathematical justification. Following the framework of Inexact Restoration methods, we proved that the introduced procedure asymptotically finds points with maximal evaluation accuracy and that, for such sequence of points, an optimality condition holds.

The main contribution of the work presented in this paper is to generalize the theory presented in [18] in such a way that we can ensure the convergence of the proposed IR-based method for an arbitrary small tolerance $\varepsilon_{\text{glo}} > 0$, i.e., $\varepsilon_{\text{glo}} \to 0$. In order to achieve convergence to a solution of the original problem with some (high) probability, in the proposed method the design of the schedule (sequence $\{N_k\}_{k=1}^{\infty}$) is obtained as a by-product of the Inexact Restoration framework. Contrary to [18], where one assumes that the scheduling will eventually result in some predefined maximal precision, here we allow the schedule to reach an arbitrary small precision if needed, but minimizing the computational effort of the overall procedure. The theory presented here covers the results in [18] as a special case. The principal advantage of the proposed scheduling is that it is not based on a predefined sequence but relies on the actual progress achieved in the optimization algorithm. Furthermore, we extend the results presented in [18] from the unconstrained case to the constrained case as well.

The efficiency of the proposed approach is demonstrated on two important problems. The first one is a classification problem where one has to solve an optimization problem defined by the objective function in the form of mathematical expectation. The SAA approach is one plausible way of solving this problem. Given that the principal cost of this approach is related to the scheduling, we demonstrate the

efficiency of the scheduling that arises from the IR-based framework. The second example comes from Portfolio Optimization, where we consider a constrained problem with mathematical expectation again.

The rest of this work is organized as follows. In Section 2 the proposed IR algorithm is presented. Convergence theory is established in Section 3. Section 4 approaches crucial implementation details and presents the numerical experiments. Final remarks are given in Section 5.

Notation. $\|\cdot\|$ denotes the Euclidean norm. $\mathbb{R}_+$ denotes the set of nonnegative real numbers. $\mathbb{N}_+$ denotes the positive integer numbers.

2. Inexact Restoration algorithm

The Inexact Restoration (IR) algorithm described in this section aims to approximate the minimum of $f(x)$ by means of successive evaluations of $f_N(x)$ employing samples of N elements. More generally, we may think that we wish to minimize $f(x)$ and that $\lim_{N\to\infty} f_N = f$ in some sense. The number of sample elements N varies (perhaps decreasing) after each evaluation of f_N as a result of the IR tests and decisions. A restoration step in the IR algorithm corresponds to increase the number of elements in the sample whereas the optimization step corresponds to improving the value of $f_N(x)$. Strictly speaking the value of $f_N(x)$ also depends on the chosen sample elements (say, $f_N(x) = f_N(\xi^1, \ldots, \xi^N, x)$ but we will maintain the notation that avoids the mention of the sample elements in order to simplify the notation. The IR approach presented in this paper is inspired by classical Inexact Restoration algorithms for constrained optimization. This idea has been used in [18] in the context of the minimization of functions whose evaluation depends on an iterative process. The main idea consists of considering that the exact evaluation of the objective function corresponds to feasibility of an implicit constrained optimization problem. In such a way, restoration steps correspond to steps where we increase the precision employed to evaluate the objective function. At optimization steps one essentially maintains the sample and one tries to reduce the corresponding sample-based objective function. The IR machinery fits well with this approach and suggests practical ideas for implementation.

For our analysis, we consider a bounded continuous function $h : \mathbb{R}_+ \to \mathbb{R}_+$ such that $h(\delta) = 0$ if and only if $\delta = 0$ and an integer-valued function $\bar{N} : \mathbb{R}_+ \to \mathbb{N}_+$. The practical meaning of these functions will be given after proving the main IR results in order to stress the fact that the IR machinery is independent of this meaning. Moreover, define, for all $x \in \mathbb{R}^n$, $\delta \in \mathbb{R}_+$, $\theta \in (0, 1)$, and $N \geq \bar{N}(\delta)$, the function

$$\Phi(x, \delta, N, \theta) = \theta f_N(x) + (1 - \theta)h(\delta).$$

Inspired by classical IR algorithms, the function Φ aims to take into account optimality defined by the functional values on a sample and feasibility defined by h.

Algorithm 2.1. Let $x_0 \in \Omega$, $\delta_0 \in \mathbb{R}_+$, $N_0 \geq \bar{N}(\delta_0)$, $\theta_0 \in (0, 1)$, $r \in (0, 1)$, $\alpha > 0$, and $\beta > 0$ be given. Set $k \leftarrow 0$.

Step 1 (Restoration Phase). Define $\delta_k^{\text{re}} \in \mathbb{R}_+$ in such a way that

$$h(\delta_k^{\text{re}}) \leq r h(\delta_k) \tag{4}$$

and a sample with $N_{k+\frac{1}{2}}$ elements such that $N_{k+\frac{1}{2}} \geq \bar{N}(\delta_k^{\text{re}})$ and

$$f_{N_{k+\frac{1}{2}}}(x_k) \leq f_{N_k}(x_k) + \beta h(\delta_k). \tag{5}$$

Step 2 (Updating the penalty parameter). If

$$\Phi(x_k, \delta_k^{\mathrm{re}}, N_{k+\frac{1}{2}}, \theta_k) \leq \Phi(x_k, \delta_k, N_k, \theta_k) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right), \tag{6}$$

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

$$\theta_{k+1} = \frac{(1+r)\left(h(\delta_k) - h(\delta_k^{\mathrm{re}})\right)}{2\left(f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k) + h(\delta_k) - h(\delta_k^{\mathrm{re}})\right)}. \tag{7}$$

Step 3 (Optimization Phase).

Step 3.1. Choose $d_k \in \mathbb{R}^n$ such that $x_k + d_k \in \Omega$.

Step 3.2. Choose $\delta_{k+1} \in \mathbb{R}_+$ (perhaps bigger than δ_k^{re}) and $N_{k+1} \geq \bar{N}(\delta_{k+1})$. If

$$f_{N_{k+1}}(x_k + d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \alpha\|d_k\|^2 \tag{8}$$

and

$$\Phi(x_k + d_k, \delta_{k+1}, N_{k+1}, \theta_{k+1}) \leq \Phi(x_k, \delta_k, N_k, \theta_{k+1}) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right), \tag{9}$$

set $t_k = 1$, $x_{k+1} = x_k + d_k$, $k \leftarrow k+1$, and go to Step 1.

Step 3.3. Redefine $\delta_{k+1} \leftarrow \delta_k^{\mathrm{re}}$ and $N_{k+1} \leftarrow N_{k+\frac{1}{2}}$. Find $t_k \in \{1, 10^{-1}, 10^{-2}, \dots\}$ as large as possible such that

$$f_{N_{k+1}}(x_k + t_k d_k) \leq f_{N_{k+1}}(x_k) - \alpha t_k \|d_k\|^2. \tag{10}$$

Set $x_{k+1} = x_k + t_k d_k$, $k \leftarrow k+1$, and go to Step 1.

Remark. At Step 3.2 the algorithm may decide that the number of elements in the sample must decrease. The idea is to implement the algorithm in such a way that a decrease of N is tried when the iterate is far from the solution. The rationale behind this is that very expensive evaluations are not worthwhile when the functional value is still large. The IR framework provides a test to decide if the reduction of N is acceptable. Note that, at first sight, the occurrence of infinitely many steps 3.2 could lead to the generation of a bounded sequence $\{N_k\}$ and, so, to the non-convergence of f_N to f. This possibility will be discarded by the convergence theory. In [18] we proved that, for a similar approach, a finite precision defined by a bound for N is reached. The proof that the satisfactory optimality result holds with N going to infinity is given in the present paper. The motivations for the tests that guarantee arbitrary precision and optimality are provided by the Inexact Restoration techniques developed for smooth constrained optimization. Step 3.2 corresponds to large trial steps in the tangent approximation to the approximate feasible set and Step 3.3 corresponds to backtracking in the tangent space.

3. Convergence theory

In this section we will prove that the sequence generated by Algorithm 2.1 converges to a suitable approximation of a solution to (1). Roughly speaking, limit points of the sequence will be stationary points with a probability as high as desired and the approximation of the sample-computed function f_N will be satisfactory also with probability arbitrarily close to 1. In the theoretical proofs we will employ the following assumptions.

Assumption A1. *The Restoration Phase defined by Step* 1 *of Algorithm* 2.1 *can be computed in finite time for all* k.

Assumption A2. *There exists* $c_L > 0$ *such that, for all* k,

$$\|\nabla f_{N_{k+\frac{1}{2}}}(x) - \nabla f_{N_{k+\frac{1}{2}}}(y)\| \leq c_L \|x - y\| \text{ for all } x, y \in \Omega,$$

i.e., the gradient of $f_{N_{k+\frac{1}{2}}}(\cdot)$ *is Lipschitz-continuous with the same Lipschitz constant for all* k. *In addition, there exists* $c > 0$ *such that, for all* k, *we have that* $f_{N_k}(x) \leq c$ *for all* $x \in \Omega$.

Assumption A3. *The iterates* x_k *generated by Algorithm* 2.1 *lie in a bounded set. (This assumption is obviously satisfied if* Ω *is bounded.)*

Assumption A4. *There exists* $\tau > 0$ *such that, for all* k, *the chosen direction* d_k *satisfies*

$$f_{N_{k+\frac{1}{2}}}(x_k + td_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \alpha t \|d_k\|^2 \text{ for all } t \in [0, \tau]. \tag{11}$$

The plausibility of Assumption A4 will be shown in the following two lemmas. In Lemma 3.1 we prove that Assumption A4 is satisfied in the unconstrained case under a very general gradient-related choice for d_k. In Lemma 3.2 we prove that Assumption A4 also holds for a general closed and convex Ω if we take d_k as a gradient projection. Note that the value of $\tau > 0$ in Assumption A4 does not depend on k. This is the reason for which, in Lemma 3.1 below, the upper bound on the norm of d_k is required.

Lemma 3.1. *Suppose that Assumption* A2 *holds. If there exist* $c_{\text{angle}} \in (0, 1)$ *and* $c_{\text{big}} > c_{\text{small}} > 0$ *such that for all* $k \in \mathbb{N}$ *we choose* d_k *satisfying*

$$d_k^T \nabla f_{N_{k+\frac{1}{2}}}(x_k) \leq -c_{\text{angle}} \|d_k\| \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\| \tag{12}$$

and

$$c_{\text{big}} \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\| \geq \|d_k\| \geq c_{\text{small}} \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\|, \tag{13}$$

then Assumption A4 *holds for any* $\alpha \in (0, \frac{1}{2} c_{\text{angle}}/c_{\text{big}}]$ *taking* $\tau = 2\alpha/c_{\text{L}}$.

Proof. By Assumption A2 and elementary calculus, we have that, for all $t > 0$,

$$f_{N_{k+\frac{1}{2}}}(x_k + td_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) + td_k^T \nabla f_{N_{k+\frac{1}{2}}}(x_k) + c_{\text{L}} t^2 \|d_k\|^2/2. \tag{14}$$

Consequently, by (12),

$$f_{N_{k+\frac{1}{2}}}(x_k + td_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - tc_{\text{angle}} \|d_k\| \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\| + c_{\text{L}} t^2 \|d_k\|^2/2. \tag{15}$$

Therefore, by (13),

$$f_{N_{k+\frac{1}{2}}}(x_k + td_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) + (c_{\text{L}} t^2/2 - tc_{\text{angle}}/c_{\text{big}}) \|d_k\|^2 \tag{16}$$

and, since $\alpha \leq \frac{1}{2} c_{\text{angle}}/c_{\text{big}}$,

$$f_{N_{k+\frac{1}{2}}}(x_k + td_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) + (c_{\text{L}} t^2/2 - 2\alpha t) \|d_k\|^2. \tag{17}$$

Therefore, (11) holds taking $\tau = 2\alpha/c_{\text{L}}$. □

Lemma 3.2. *Suppose that Assumption* A2 *holds. Assume that* $\sigma_{\max} \geq \sigma_{\min} > 0$ *are given parameters and that, for all* $k \in \mathbb{N}$, *we choose* $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$ *and compute* $d_k = P_\Omega(x_k - \sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)) - x_k$, *where* P_Ω *denotes the projection operator onto* Ω. *Then, Assumption* A4 *holds taking* $\alpha = 1/(4\sigma_{\max})$ *and* $\tau = \min\{1, 1/(2\sigma_{\max} c_{\mathrm{L}})\}$.

Proof. By the definition of d_k we have that $x_k + d_k$ is the projection of $x_k - \sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)$ onto the convex set Ω. Therefore, the points x_k, $x_k - \sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)$, and $x_k + d_k$ are the vertices of a triangle, the angle corresponding to $x_k + d_k$ is obtuse, and the opposite side to this vertex is the biggest one. Therefore,

$$\|d_k + \sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)\|^2 \leq \|\sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)\|^2.$$

So,

$$\|d_k\|^2 + \sigma_k^2 \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\|^2 + 2\sigma_k d_k^T \nabla f_{N_{k+\frac{1}{2}}}(x_k) \leq \|\sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)\|^2.$$

Thus,

$$\nabla f_{N_{k+\frac{1}{2}}}(x_k)^T d_k \leq -\frac{1}{2\sigma_k} \|d_k\|^2. \tag{18}$$

Since x_k and $x_k + d_k \in \Omega$ and Ω is convex, we obtain, by Assumption A2 and elementary calculus, that

$$f_{N_{k+\frac{1}{2}}}(x_k + t d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) + t \nabla f_{N_{k+\frac{1}{2}}}(x_k)^T d_k + \frac{1}{2} t^2 c_{\mathrm{L}} \|d_k\|^2$$

for all $t \in [0, 1]$. Therefore, by (18),

$$f_{N_{k+\frac{1}{2}}}(x_k + t d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \frac{1}{2\sigma_k} t \|d_k\|^2 + \frac{1}{2} t^2 c_{\mathrm{L}} \|d_k\|^2$$

for all $t \in [0, 1]$. Then,

$$f_{N_{k+\frac{1}{2}}}(x_k + t d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \left[\frac{1}{2\sigma_{\max}} - \frac{1}{2} t c_{\mathrm{L}}\right] t \|d_k\|^2$$

for all $t \in [0, 1]$. Thus, since $\tau < 1$, for all $t \in [0, \tau]$, we have:

$$f_{N_{k+\frac{1}{2}}}(x_k + t d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \left[\frac{1}{2\sigma_{\max}} - \frac{1}{2} \tau c_{\mathrm{L}}\right] t \|d_k\|^2$$

and, by the hypothesis on τ,

$$f_{N_{k+\frac{1}{2}}}(x_k + t d_k) \leq f_{N_{k+\frac{1}{2}}}(x_k) - \left[\frac{1}{2\sigma_{\max}} - \frac{1}{4\sigma_{\max}}\right] t \|d_k\|^2$$

for all $t \in [0, \tau]$. Then, by the choice of α in the hypothesis, Assumption A4 holds. □

In the following results we will assume, without mentioning explicitly, that Assumptions A1–A4 hold. In the next lemma we prove that the penalty parameter θ_k is nonincreasing and bounded away from zero. This result is crucial in the IR framework to guarantee that, ultimately, the necessity of minimizing the objective function is preserved and that this goal is not overwhelmed by the search of feasibility.

Lemma 3.3. *The sequence* $\{\theta_k\}$ *generated by Algorithm* 2.1 *remains in* $(0,1)$*, is nonincreasing, and is bounded below by a positive quantity that only depends on* β*,* r*, and* θ_0*.*

Proof. At each iteration, we have that $\theta_{k+1} = \theta_k$ when (6) takes place and that θ_{k+1} is computed by (7) otherwise. If (6) does not hold, it turns out that

$$\Phi(x_k, \delta_k^{\mathrm{re}}, N_{k+\frac{1}{2}}, \theta) > \Phi(x_k, \delta_k, N_k, \theta) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right) \tag{19}$$

for $\theta = \theta_k$. On the other hand, since $h(\delta_k^{\mathrm{re}}) < h(\delta_k)$ and, thus,

$$h(\delta_k^{\mathrm{re}}) - h(\delta_k) \leq \frac{1-r}{2}(h(\delta_k^{\mathrm{re}}) - h(\delta_k)),$$

we have that the negation of (19) always holds strictly if $\theta = 0$. Since Φ is a linear function on θ, this means that there exists a unique $\theta \in (0, \theta_k)$ that verifies

$$\Phi(x_k, \delta_k^{\mathrm{re}}, N_{k+\frac{1}{2}}, \theta) = \Phi(x_k, \delta_k, N_k, \theta) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right). \tag{20}$$

This θ is the quantity θ_{k+1} computed at (7). Therefore, in order to prove the lemma, it only remains to be shown that θ_{k+1} defined by (7) is bounded away from zero. In fact, by (7), (4), and (5) we have:

$$\begin{aligned}\frac{1}{\theta_{k+1}} &= 2\left[\frac{f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k)}{h(\delta_k) - h(\delta_k^{\mathrm{re}})} + 1\right]\frac{1}{1+r} \\ &\leq 2\left[\frac{\beta h(\delta_k)}{(1-r)h(\delta_k)} + 1\right]\frac{1}{1+r} = 2\left[\frac{\beta}{(1-r)} + 1\right]\frac{1}{1+r}.\end{aligned}$$

This completes the proof. □

Lemma 3.4 shows that each iteration of the algorithm satisfies both the requirement of decreasing the approximate objective function and the function Φ, that combines optimality and feasibility. Note that, together with Assumption A4, Lemma 3.4 implies that Algorithm 2.1 is well defined.

Lemma 3.4. *For all* $k = 0, 1, 2, \ldots$, (8) *and* (9) *hold. Moreover, either* $t_k = 1$ *is set at Step* 3.2*, or, after a finite number of evaluations which only depend on* τ*,* (10) *holds and* $t_k \in \{1, 10^{-1}, 10^{-2}, \ldots\}$ *is set at Step* 3.3*. In addition, in this case, we have that* $t_k \geq \tau/10$ *and*

$$\Phi(x_{k+1}, \delta_{k+1}, N_{k+1}, \theta_{k+1}) \leq \Phi(x_k, \delta_k, N_k, \theta_{k+1}) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right). \tag{21}$$

Proof. The fact that (10) takes place after a finite number of evaluations that only depends on τ is a trivial consequence of Assumption A4. Also by Assumption A4, the first value of t such that (10) holds is not smaller than $\tau/10$. Let us show now that, at Step 3.3, when (10) holds, we also obtain (21).

If (6) holds then we have that $\theta_{k+1} = \theta_k$ and, therefore, (6) also holds substituting θ_k with θ_{k+1}. If (6) does not hold, the choice (7) of θ_{k+1} is such that (6) holds (by equality) substituting θ_k with θ_{k+1}. Therefore, in any case we have

$$\Phi(x_k, \delta_{k+1}, N_{k+1}, \theta_{k+1}) \leq \Phi(x_k, \delta_k, N_k, \theta_{k+1}) + \frac{1-r}{2}\left(h(\delta_k^{\mathrm{re}}) - h(\delta_k)\right). \tag{22}$$

On the other hand,

$$\begin{aligned}\Phi(x_{k+1},\delta_{k+1},N_{k+1},\theta_{k+1}) &= \theta_{k+1}f_{N_{k+1}}(x_{k+1})+(1-\theta_{k+1})h(\delta_{k+1},)\\ &\le \theta_{k+1}\left[f_{N_{k+1}}(x_k)-\alpha t_k\|d_k\|^2\right]+(1-\theta_{k+1})h(\delta_{k+1})\\ &= \theta_{k+1}f_{N_{k+1}}(x_k)+(1-\theta_{k+1})h(\delta_{k+1})-\theta_{k+1}\alpha t_k\|d_k\|^2\\ &= \Phi(x_k,\delta_{k+1},N_{k+1},\theta_{k+1})-\theta_{k+1}\alpha t_k\|d_k\|^2\\ &\le \Phi(x_k,\delta_{k+1},N_{k+1},\theta_{k+1})\end{aligned}$$

and (21) holds by (22). □

In the following theorem we prove that the algorithm is satisfactory with respect to feasibility, which, in this case, means accurate evaluation of the objective function. This property is represented by the statement $h(\delta_k^{\mathrm{re}})\to 0$ when $k\to\infty$. Note that validity of the theorem is independent of the meaning of h which, up to now, has been associated with accuracy (or feasibility) only in terms of motivation.

Theorem 3.1. *The series $\sum_{k=0}^{\infty}h(\delta_k)$ and $\sum_{k=0}^{\infty}h(\delta_k^{\mathrm{re}})$ are convergent.*

Proof. By (4) we only need to prove that $\sum_{k=0}^{\infty}h(\delta_k)$ is convergent. By Lemma 3.3, there exists $\theta_*>0$, the limit of the nonincreasing sequence $\{\theta_k\}$. Define $\rho_k=(1-\theta_k)/\theta_k$ for all $k\in\mathbb{N}$. Clearly $\{\rho_k\}$ is nondecreasing and bounded above by $1/\theta_*-1$. Therefore,

$$\sum_{k=0}^{\infty}(\rho_{k+1}-\rho_k)=\lim_{k\to\infty}\rho_{k+1}-\rho_0<\infty. \tag{23}$$

Now, since $\{h(\delta_k)\}$ is bounded, (23) implies that

$$\sum_{k=0}^{\infty}(\rho_{k+1}-\rho_k)h(\delta_k)<\infty. \tag{24}$$

On the other hand, by (21) in Lemma 3.4, (11), and the definition of x_{k+1}, we have that

$$\Phi(x_{k+1},\delta_{k+1},N_{k+1},\theta_{k+1})\le\Phi(x_k,\delta_k,N_k,\theta_{k+1})-\frac{(1-r)^2}{2}h(\delta_k). \tag{25}$$

Inequality (25) is equivalent to

$$f_{N_{k+1}}(x_{k+1})+\rho_{k+1}h(\delta_{k+1})\le f_{N_k}(x_k)+\rho_{k+1}h(\delta_k)-\frac{(1-r)^2}{2\theta_{k+1}}h(\delta_k).$$

Adding and subtracting $\rho_k h(\delta_k)$ and rearranging, as $\theta_{k+1}\in(0,1)$, we get:

$$\begin{aligned}\frac{(1-r)^2}{2}h(\delta_k)\le&(\rho_{k+1}-\rho_k)h(\delta_k)\\ &+\left[(f_{N_k}(x_k)+\rho_k h(\delta_k))-\left(f_{N_{k+1}}(x_{k+1})+\rho_{k+1}h(\delta_{k+1})\right)\right].\end{aligned}$$

Summing over all k, we obtain

$$\begin{aligned}\sum_{k=0}^{\infty}\frac{(1-r)^2}{2}h(\delta_k)\le&\sum_{k=0}^{\infty}(\rho_{k+1}-\rho_k)h(\delta_k)\\ &+\left[(f_{N_0}(x_0)+\rho_0h(\delta_0))-\lim_{k\to\infty}\left(f_{N_k}(x_k)+\rho_kh(\delta_k)\right)\right]\end{aligned}$$

that, by (24) and the boundedness of $\{x_k\}$ and $\{\rho_k\}$, implies that

$$\sum_{k=0}^{\infty} \frac{(1-r)^2}{2} h(\delta_k) < \infty.$$

Thus,

$$\sum_{k=0}^{\infty} h(\delta_k) < \infty,$$

as we wanted to prove. □

Finally, in Theorem 3.2, we prove that we are able to find points at which the gradient of the approximate sample-based function is as small as desired.

Theorem 3.2. *The series $\sum_{k=0}^{\infty} \|d_k\|^2$ is convergent.*

Proof. The inequalities (8) and (10), and the lower bound for t_k stated at Lemma 3.4 (say $t_k \geq t_*$ for all k) imply that

$$\begin{aligned} f_{N_{k+1}}(x_{k+1}) - f_{N_k}(x_k) &= f_{N_{k+1}}(x_{k+1}) - f_{N_{k+\frac{1}{2}}}(x_k) + f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k) \\ &\leq -\alpha t_k \|d_k\|^2 + f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k) \\ &\leq -\alpha t_* \|d_k\|^2 + |f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k)| \\ &\leq -\alpha t_* \|d_k\|^2 + \beta h(\delta_k). \end{aligned}$$

Thus, for an arbitrary integer k, we have

$$f_{N_k}(x_k) - f_{N_0}(x_0) \leq -\alpha t_* \sum_{j=0}^{k-1} \|d_j\|^2 + \beta \sum_{j=k}^{k-1} h(\delta_j).$$

Therefore, by Assumptions A2 and A3, and Theorem 3.1,

$$\alpha t_* \sum_{j=0}^{\infty} \|d_j\|^2 < \infty,$$

as we wanted to prove. □

If h is associated with feasibility, then Theorem 3.1 means that limit points of the sequence generated by Algorithm 2.1 are feasible points; while if h is associated with precision of evaluation of the objective function of (1), Theorem 3.1 means that in the limit the objective function is evaluated exactly. On the other hand, Theorem 3.2 implies that limit points satisfy an optimality condition as well.

Corollary 3.1. *If $\Omega = \mathbb{R}^n$ and we choose α and d_k as in the hypotheses of Lemma* 3.1*, we have that*

$$\lim_{k \to \infty} \|\nabla f_{N_{k+\frac{1}{2}}}(x_k)\| = 0.$$

Proof. The desired result follows from (13) and Theorem 3.2. □

Corollary 3.2. *If Ω is an arbitrary closed and convex set and we choose α and d_k as in the hypotheses of Lemma* 3.2*, we have that*

$$\lim_{k \to \infty} \|P_\Omega(x_k - \sigma_k \nabla f_{N_{k+\frac{1}{2}}}(x_k)) - x_k\| = 0$$

for all $k \in \mathbb{N}$.

Proof. The desired result follows from Theorem 3.2. □

4. Implementation features and numerical experiments

Algorithm 2.1, as well as the Projected Gradient method (that will be used for comparison purposes), were implemented in Fortran 90. All tests were conducted on a computer with 3.5 GHz Intel Core i7 processor and 16GB 1600 MHz DDR3 RAM memory, running OS X Yosemite (version 10.10.4). Codes were compiled by the GFortran Fortran compiler of GCC (version 4.9.2) with the -O3 optimization directive enabled. Regarding the parameters of Algorithm 2.1, arbitrarily but based on previous experimentation with Inexact Restoration methods, we set $\delta_0 = 0.01$, $N_0 = \bar{N}(\delta_0)$, $\theta_0 = 0.9$, and $\alpha = 10^{-4}$. The remaining parameters will be mentioned below.

4.1. **Implementation details.** Defining $h(\delta) = \delta$, and $\bar{N}(\delta) = \lceil 1/\delta \rceil$, the interpretation of the convergence results presented in the previous section is that, given $\varepsilon_{\text{opt}} > 0$ and $\underline{N} > 0$, Algorithm 2.1 obtains, in finite time, a sample with $N_{k+1} \geq \underline{N}$ elements and a point x_k such that x_k satisfies an optimality condition for minimizing $f_{N_{k+1}}(x)$ subject to $x \in \Omega$ with precision ε_{opt}. At Step 3.1 of Algorithm 2.1, we considered

$$d_k = P_\Omega(x_k - \nabla f_{N_{k+1}}(x_k)) - x_k.$$

So, by Lemma 3.2, if Assumption A2 holds, Assumption A4 holds as well. With this choice, a natural stopping criterion for Algorithm 2.1 is given by

$$N_{k+1} \geq \underline{N} \text{ and } \|P_\Omega(x_k - \nabla f_{N_{k+1}}(x_k)) - x_k\|_\infty \leq \varepsilon_{\text{opt}}. \tag{26}$$

Moreover, due to the choice of the search direction, in order to evaluate the performance of Algorithm 2.1, it would be natural to compare it against the classical Projected Gradient (PG) method applied to the minimization of $f_{\underline{N}}(\cdot)$ and considering the stopping criterion

$$\|P_\Omega(x_k - \nabla f_{\underline{N}}(x_k)) - x_k\|_\infty \leq \varepsilon_{\text{opt}}. \tag{27}$$

At Step 3.2, we always try $\delta_{k+1} = 0.01$ and $N_{k+1} = \bar{N}(\delta_{k+1})$. The idea behind this radical choice is to drastically reduce sample sizes that may be unnecessarily large at the beginning or far from a solution.

The freedom in the condition (4) at Step 1 allows one to use an adaptive choice for the samples employed at each iteration of the method. The idea is that, if one is far from the solution, it is not worthwhile to use a very big sample, but large samples are justified if we think that we are close to a solution. Specifically, we define

$$\delta_k^{\text{re}} = \begin{cases} r_1\delta_k, & \text{if } N_k \geq \underline{N} \text{ or } \|P_\Omega(x_{k-1} - \nabla f_{N_k}(x_{k-1})) - x_{k-1}\|_\infty > \varepsilon_{\text{opt}}, \\ r_2\delta_k, & \text{otherwise,} \end{cases} \tag{28}$$

where $r_1 = 1 - 10^{-6}$ and $r_2 = 0.1$ (the parameter r being given by $\max\{r_1, r_2\}$). The effect of this should be that a suitable approximation to a solution should be obtained employing reasonable small samples being big samples reserved to the case in which we are close to a solution. There is no practical way to guarantee that, for a given value of $\beta > 0$, the choice (28) of δ_k^{re} plus $N_{k+\frac{1}{2}} = \bar{N}(\delta_k^{\text{re}})$ will satisfy (5). Therefore, in practice, we compute

$$\max_{k=0,1,2,\ldots} \{[\beta_k]_+\},$$

where

$$\beta_k \equiv \frac{f_{N_{k+\frac{1}{2}}}(x_k) - f_{N_k}(x_k)}{h(\delta_k)}$$

and $[\cdot]_+ = \max\{0, \cdot\}$, in order to verify whether this sequence appears to be bounded above (meaning that Assumption A1 holds) or not.

4.2. **A classification problem.** We now describe a classification scheme that can be written as an optimization problem. Assume that there exists an (oracle) unknown function $\omega : B \to \{1, -1\}$ that is used to classify any $\xi \in B \subseteq \mathbb{R}^m$ attributing to it 1 or -1. To fix ideas we may think in $B = \{\xi \in \mathbb{R}^m \mid \xi_{\min} \leq \xi_i \leq \xi_{\max}, i = 1, \ldots, m\}$, where $\xi_{\max} \geq \xi_{\min}$ are given values. Usually, $N > 0$ and an independent and identically distributed sample or training set of labelled examples

$$D = \{(\xi^i, \omega_i),\ i = 1, \ldots, N, \text{where } \xi^i \in B \text{ and } \omega_i \equiv \omega(\xi^i) \in \{1, -1\}\}$$

are given, and the goal is to find a classifier $C_x(\xi)$ that depends on unknown parameters $x \in \mathbb{R}^n$ such that $C_x(\xi^i) \leq 0$ if $\omega(\xi^i) = -1$ and $C_x(\xi^i) \geq 0$, if $\omega(\xi^i) = 1$. Since such a classifier may not exist, one seeks to minimize the function given by

$$f(x) = f_N(\xi^1, \ldots, \xi^N, x) = \frac{1}{N}\left[\sum_{i \in I} \max\{0, C_x(\xi^i)\}^2 + \sum_{i \in O} \max\{0, -C_x(\xi^i)\}^2\right],$$

where $I = \{i \in \{1, \ldots, N\} \mid \omega_i = -1\}$ and $O = \{i \in \{1, \ldots, N\} \mid \omega_i = 1\}$, subject to appropriate constraints on the parameters x (that may define the shape of the classifier). This was the problem considered in [8] (where it was arbitrarily considered the case $N = 10\,000$). On the other hand, in this work, we consider the stochastic problem given by

$$\text{Minimize } E\left(\sum_{i \in \mathcal{I}} \max\{0, C_x(\xi)\}^2 + \sum_{i \in \mathcal{O}} \max\{0, -C_x(\xi)\}^2\right) \text{ subject to } x \in \Omega,$$

where $\mathcal{I} = \{\xi \in B \mid \omega(\xi) = -1\}$, $\mathcal{O} = \{\xi \in B \mid \omega(\xi) = 1\}$, and Ω represents the constraints on x. It is assumed that consulting the oracle $\omega(\cdot)$ is expensive and that the goal is to find $C_x(\cdot)$ in order to (a) mimic the oracle answer and (b) understanding the relevance (or not) of each component of ξ in the classification.

Example 1. In this example, we consider $m = 2$, $n = 3$, $C_x(\xi) = C_{(c,r)}(\xi) = \|\xi - c\|^2 - r^2$, and $\Omega \equiv \mathbb{R}^n$ meaning that the classifier is a circle of radius $|r|$ centered at c in $\mathbb{R}^2$. Following [8], we consider four instances of this example corresponding to four different oracles as follows:

- **Instance 1.1.** $\omega(\xi) = -1$ if ξ belongs to a circle with radius 7 centered at the origin and $\omega(\xi) = 1$, otherwise.
- **Instance 1.2.** $\omega(\xi) = -1$ if ξ belongs to a square of side 7 centered at the origin and $\omega(\xi) = 1$, otherwise.
- **Instance 1.3.** $\omega(\xi) = -1$ if ξ belongs to a rectangle with height equal to 7 and width equal to 14 centered at the origin and $\omega(\xi) = 1$, otherwise.
- **Instance 1.4.** $\omega(\xi) = -1$ if ξ belongs to a triangle with vertices $(-7, 0)$, $(0, -7)$, and $(7, 7)$ and $\omega(\xi) = 1$, otherwise.

In all cases we considered $B = \{\xi \in \mathbb{R}^2 \mid -10 \leq \xi_i \leq 10, i = 1, 2\}$.

Example 2. In this example, we consider $m = 2$, $n = q(q+1)$, $C_x(\xi) = C_{(A,b)}(\xi) = \xi^T A\xi + b^T\xi - 1$, and $\Omega \subset \mathbb{R}^{q\times q}$ as the set of positive definite matrices $A \in \mathbb{R}^{q\times q}$ with all their eigenvalues $\lambda_i(A)$, $i = 1, \ldots, q$, satisfying

$$0 < \lambda_{\min} \leq \lambda_i(A) \leq \lambda_{\max} < +\infty,$$

where $0 < \lambda_{\min} \leq \lambda_{\max} < +\infty$ are given parameters. This means that the classifier is an ellipse in $\mathbb{R}^2$. It is worth noting that, given a square matrix A, its projection onto the closed and convex set Ω can be computed in two steps [10, 13]. In the first step one symmetrizes A and in the second step one computes the QDQ^T decomposition of the symmetrized matrix and replaces its eigenvalues by their projection onto the interval $[\lambda_{\min}, \lambda_{\max}]$. In the numerical experiments, we consider $\lambda_{\min} = 10^{-4}$ and $\lambda_{\max} = 10^4$. Moreover, the same four instances corresponding to the oracles described for Example 1, that here we named Instances 2.1–2.4, are also considered.

4.3. **Numerical results of the classification problem.** As described in two previous subsections, in the numerical experiments we evaluated the performance of the Inexact Restoration method (Algorithm 2.1), called "IR method" from now on, and we compare it against the classical Projected Gradient method, called "PG method" from now on. In the stopping criteria (26) and (27) for the IR and the PG method, respectively, we considered $\varepsilon_{\text{opt}} = 10^{-4}$ and $\underline{N} \in \{10^4, 10^5, \ldots, 10^8\}$.

The cost that dominates the computations in both methods is the evaluation of the objective function. In the case of the PG method, that is applied to the minimization of $f_{\underline{N}}(\cdot)$, the cost of evaluating the objective function is given by $c\underline{N}$, where $c > 0$ is a constant that represents the cost of evaluating $C(p)$, i.e., verifying whether a point $p \in \mathbb{R}^2$ is "inside" or "outside" the classifier C. Therefore, the total cost of the PG method is given by the total number of functional evaluations, "#fcnt" times $c\underline{N}$. In order to avoid big figures, we will scale this total effort dividing it by $c\underline{N}$ and, therefore, the reported "total effort" of the PG method will be given by the total number of functional evaluations #fcnt. In the case of the IR method, which evaluates the objective function with samples of varying sizes, the scaled total effort is given by

$$\frac{cN_0}{\underline{N}} + \sum_{k=0}^{\#\text{it}} \left[\frac{c(s_{k+\frac{1}{2}} N_{k+\frac{1}{2}} + s_{k+1} N_{k+1})}{c\underline{N}} \right] = \frac{1}{\underline{N}} \left(N_0 + \sum_{k=0}^{\#\text{it}} \left[s_{k+\frac{1}{2}} N_{k+\frac{1}{2}} + s_{k+1} N_{k+1} \right] \right),$$

where #it is the number of iterations performed until the stopping criterion is reached and $s_{k+\frac{1}{2}}$ and s_{k+1} are the number of functional evaluations with samples of sizes $N_{k+\frac{1}{2}}$ and N_{k+1}, respectively, performed at iteration k.

Table 1 presents a comparison of the performances of both methods when applied to Instances 1.1–1.4 of Example 1 and Instances 2.1–2.4 of Example 2 varying $\underline{N} \in \{10^4, 10^5, \ldots, 10^8\}$. In all cases, except the case of the PG method applied to Instance 2.2, the methods achieved their respective stopping criteria and found qualitative equivalent solutions. In the case in which the PG method was unable to achieve its stopping criterion (marked with "–" in the table), it stopped due to a very small step in the line search (smaller than 10^{-16}). Moreover, even in this case, the projected gradient was relatively small (of the order of 10^{-3}) and the final

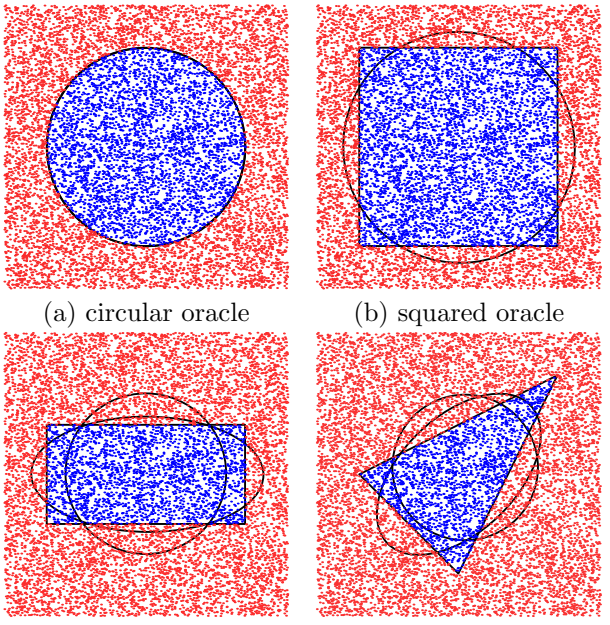
TABLE 1. Comparison between the Projected Gradient (PG) and the Inexact Restoration (IR) methods applied to Instances 1.1 to 1.4 of Example 1 and Instances 2.1 to 2.4 of Example 2 varying $\underline{N} \in \{10^4, 10^5, \ldots, 10^8\}$.

			Example 1: Circular classifier		Example 2: Ellipsoidal classifier	
			Total Effort		Total Effort	
			PG method	IR method	PG method	IR method
$\underline{N} = 10^4$	Oracle	circle	100	114	350	153
		square	64	63	–	1,667
		rectangle	31	54	831	2,525
		triangle	31	46	1,005	1,559
$\underline{N} = 10^5$	Oracle	circle	66	13	340	17
		square	67	55	–	1,457
		rectangle	37	35	1,205	1,784
		triangle	29	29	1,081	747
$\underline{N} = 10^6$	Oracle	circle	66	3	183	6
		square	67	46	–	682
		rectangle	37	24	884	1,614
		triangle	28	24	686	459
$\underline{N} = 10^7$	Oracle	circle	60	2	197	3
		square	67	40	–	671
		rectangle	37	13	1,453	969
		triangle	28	20	541	256
$\underline{N} = 10^8$	Oracle	circle	59	1	197	7
		square	67	5	–	466
		rectangle	37	2	1,488	155
		triangle	28	3	655	296

iterate was a "reasonable approximation" to a solution. Figure 1 shows a pictorial representation of the solutions found.

A comparison of the figures in the table shows that the IR method is more efficient in 32 out of the 40 considered combinations of instances and $\underline{N}$ (i.e., 80% of the cases). The PG method is more efficient in seven other instances and there is a tie between the two methods in a single instance. The instances in which the PG method is competitive are related to small values of $\underline{N}$; while the instances in which the IR method is more efficient are the instances with very large values of $\underline{N}$. If we consider the instances with $\underline{N} = 10^8$ only, the IR method is at least twice faster than the PG method and, in some cases, it is ten times faster and up to more than fifty times faster than the PG method. The cases in which the IR method appears to be extremely fast are cases in which samples with size $\underline{N}$ are required by the method (i.e., used to evaluate the objective function) only once or twice. Figure 2 displays, for the four instances of Example 2 with $\underline{N} = 10^8$, the evolution of the size of the samples as the method evolves. It is very clear that, as desired, big samples are used only when the method approaches a solution.

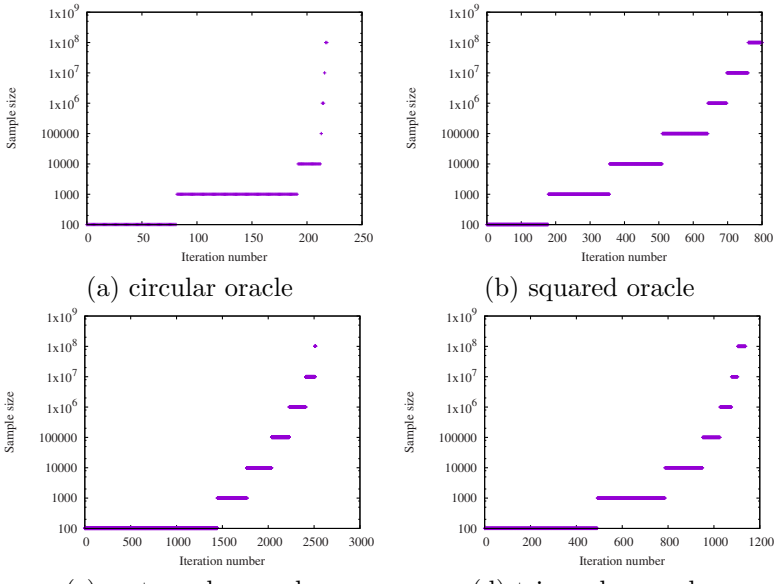
4.4. **A portfolio optimization problem.** Assume that, using simulation, we are able to produce as many scenarios of future prices of a set of assets as we desire. One of the strategies used to compose a portfolio consists of maximizing the expected return minus a multiple of the estimated standard deviation. This objective function satisfies adequately the requirement of combining future profits with a cautious attention to risk. Moreover, variations on the weight given to the standard



(a) circular oracle (b) squared oracle

(c) rectangular oracle (d) triangular oracle

FIGURE 1. Solutions to the four considered instances of Examples 1 (circular classifier) and 2 (ellipsoidal classifier). In the case of instances (a) and (b) both solutions coincide.



(a) circular oracle (b) squared oracle

(c) rectangular oracle (d) triangular oracle

FIGURE 2. Samples size as the method evolves in the four instances of Example 2 (ellipsoidal classifier).

deviation produce different solutions and the corresponding pairs return/deviation represent the classical Pareto curve of nondominated decisions, among which the investor should choose the one that best fits his/her desires, perhaps based on non-quantifiable criteria. Fixing the weight of the standard deviation we are in presence of a problem of the type analyzed in this paper. As in previous examples, we are interested in the optimization of the original problem, employing a moderate number of scenarios at the portfolio trials that are far from a solution.

Let us consider a portfolio consisting of n assets and assume that $\xi \in \mathbb{R}^n$ is a continuous random variable that represents a certain scenario of future prices of the assets. Thus, if x_j is the present value of z units of the asset j, then $\xi_j x_j$ is the inflation-discounted prize of z units of the asset j one period later under the scenario represented by ξ. The expected return of a portfolio $x \in \mathbb{R}^n$ is $E(\sum_{i=1}^n \xi_j x_j)$; while its standard deviation is given by $\sigma(\sum_{i=1}^n \xi_j x_j)$. Including the budget constraint $\sum_{j=1}^n x_j = 1$ and nonnegativity constraints $x \geq 0$, the objective of an investor could be

$$\begin{aligned} &\text{Maximize } \gamma \left[E(\sum_{j=1}^n \xi_j x_j) \right] + (1-\gamma) \left[-\sigma(\sum_{j=1}^n \xi_j x_j) \right] \\ &\text{subject to } \sum_{j=1}^n x_j = 1 \text{ and } x \geq 0, \end{aligned}$$

where $\gamma \in [0,1]$ is a given parameter. As in the previous subsection, we assume that a large integer $\underline{N} > 0$ and small tolerance $\varepsilon_{\text{opt}} > 0$ are given and that the objective is to find a point $x^* \in \Omega$ such that

$$\|P_\Omega\left(x^* - \nabla f_N(x^*)\right) - x^*\|_\infty \leq \varepsilon_{\text{opt}}$$

for some $N \geq \underline{N}$, where

$$\Omega = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n x_j = 1 \text{ and } x \geq 0 \right\},$$

$$f_N(x) = \gamma \left[\sum_{j=1}^n \bar{\xi}_j x_j \right] + (1-\gamma) \left[-\sqrt{\frac{1}{N} \sum_{i=1}^N \left(\sum_{j=1}^n [\xi_i]_j x_j - \sum_{j=1}^n \bar{\xi}_j x_j \right)^2} \right],$$

and

$$\bar{\xi} = \frac{1}{N} \sum_{i=1}^N \xi_i \in \mathbb{R}^n.$$

For the scenarios' generation, following [7], we considered the date range from January 14, 2013, to January 12, 2015, that, for the NYSE, implies in 503 observable days. Companies were taken from the listing of the NYSE available at `http://www.nasdaq.com/screening/company-list.aspx` that contains 3 300 companies (accessed on January 13, 2015). We considered the first 100 listed companies excluding those with less than 503 observations and those that contain the character ^ in its symbol. We took historical data (adjusted close value) from Yahoo! Finance using the web tool available at `http://finance.jasonstrimpel.com`, that is able to download multiple stock data series on one spreadsheet. We named this real data $\hat{D} \in \mathbb{R}^{503 \times 100}$. To obtain a matrix $D \in \mathbb{R}^{503 \times n}$ with $n = 101$, we added a

TABLE 2. Comparison between the Projected Gradient (PG) and the Inexact Restoration (IR) methods applied to the portfolio optimization problem with $\underline{N} \in \{10^6, 10^7\}$ and $\gamma \in \{1, 0.9, \ldots, 0.4\}$.

			Portfolio optimization problem	
			Total Effort	
			PG method	IR method
$\underline{N} = 10^6$	parameter γ	1.0	29	6
		0.9	115	66
		0.8	50	31
		0.7	40	22
		0.6	36	17
		0.5	26	12
		0.4	25	19
$\underline{N} = 10^7$	parameter γ	1.0	28	3
		0.9	116	17
		0.8	50	5
		0.7	40	4
		0.6	36	4
		0.5	26	9
		0.4	26	12

last column given by $(1, 1, \ldots, 1)^T \in \mathbb{R}^{503}$ corresponding to a risk-free asset. Using matrix D as a source, we computed the matrix of growing factors $G \in \mathbb{R}^{502 \times n}$ whose elements g_{ij} are given by $g_{ij} = d_{i+1,j}/d_{ij}$ for $i = 1, \ldots, 502$ and $j = 1, \ldots, n$. Each scenario $\xi \in \mathbb{R}^n$ will represent a scenario that corresponds to applying 250 random growing factors to the unitary vector $e = (1, \ldots, 1)^T \in \mathbb{R}^n$ (note that 250 observable days corresponds to a calendar year).

For comparison purposes, it is worth noting that the asset with largest expected profit is asset $j = 21$ with $\bar{\xi}_{21} \approx 1.8601$ (when we consider 10^6 as well as 10^7 scenarios). Therefore, when $\gamma = 1$, the solution to the problem at hand (that in this case reduces to a linear programming problem) is given by investing all the budget on asset $j = 21$. On the other hand, when $\gamma = 0$, the considered problem reduces to minimizing the standard deviation of the portfolio. In this case, the solution is given by investing all the budget in the risk-free asset $j = 101$, that behaves exactly in the same way for all possible scenarios. It is worth noting that the objective function is not differentiable at points where the standard deviation vanishes. For that reason, in the numerical experiments we considered $\gamma \in \{1, 0.9, \ldots, 0.4\}$. We also considered that $\varepsilon_{\text{opt}} = 10^{-4}$ and $\underline{N} \in \{10^6, 10^7\}$. Table 2 shows the results. In all cases both methods (PG and IR) satisfied the stopping criterion and found equivalent solutions. In the case $\underline{N} = 10^6$, the Inexact Restoration method used approximately half of the computational effort required by the Projected Gradient method; while in the larger case ($\underline{N} = 10^7$), the IR method was almost one order of magnitude faster than the PG method. Figure 3 shows the value of the expected return versus the standard deviation in the solutions found for the different values of the parameter γ.

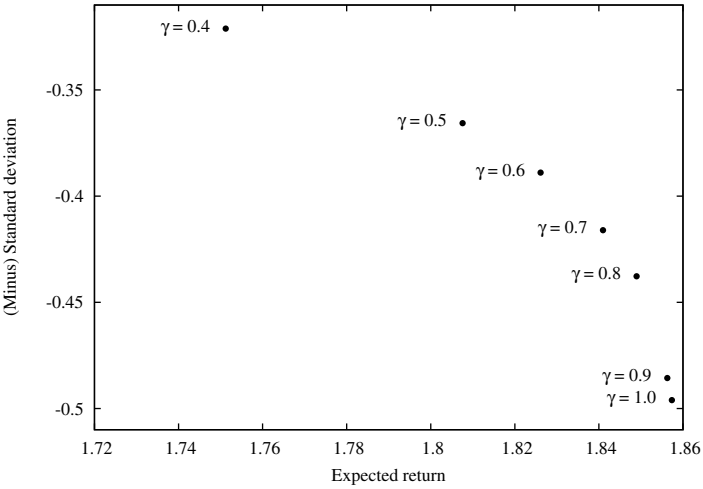


FIGURE 3. Expected return versus standard deviation for γ varying from 1 to 0.4.

5. Concluding remarks

Perhaps the main theoretical meta-optimization consequence of this work is that the key word in Inexact Restoration is "inexactness" instead of "feasibility" or even "restoration". In [18] this consequence was timidly suggested as we still relied on a reformulation of the minimization of $f(x)$ using an auxiliary constraint $z = f(x)$. Here we discarded such interpretation at all and, as a matter of fact, our present preference is the reverse one: Instead of thinking inexactness through the model of infeasibility, we believe that it is more useful to think infeasibility through the model of inexactness. For example, in nonlinear programming, the evaluation of $f(x)$ at an infeasible point may be considered an inexact evaluation of $f(x)$ and we conjecture that many useful ideas may come from the encounter between the IR framework and the treatment of inexactness in mathematical problems.

Acknowledgments

The authors would like to thank the two referees for their comments and remarks that helped to improve the first version of this work.

References

[1] R. Andreani, S. L. C. Castro, J. L. Chela, A. Friedlander, and S. A. Santos, *An inexact-restoration method for nonlinear bilevel programming problems*, Comput. Optim. Appl. **43** (2009), no. 3, 307–328, DOI 10.1007/s10589-007-9147-4. MR2511917

[2] N. Banihashemi and C. Yalçın Kaya, *Inexact restoration for Euler discretization of box-constrained optimal control problems*, J. Optim. Theory Appl. **156** (2013), no. 3, 726–760, DOI 10.1007/s10957-012-0140-4. MR3022307

[3] F. Bastin, *Trust-region algorithms for nonlinear stochastic programming and mixed logit models*, Ph.D. Thesis, University of Namur, Namur, Belgium, 2004.

[4] F. Bastin, C. Cirillo, and P. L. Toint, *An adaptive Monte Carlo algorithm for computing mixed logit estimators*, Comput. Manag. Sci. **3** (2006), no. 1, 55–79, DOI 10.1007/s10287-005-0044-y. MR2196398

[5] E. G. Birgin, L. F. Bueno, and J. M. Martínez, *Assessing the reliability of general-purpose inexact restoration methods*, J. Comput. Appl. Math. **282** (2015), 1–16, DOI 10.1016/j.cam.2014.12.031. MR3313085

[6] E. G. Birgin and J. M. Martínez, *Local convergence of an inexact-restoration method and numerical experiments*, J. Optim. Theory Appl. **127** (2005), no. 2, 229–247, DOI 10.1007/s10957-005-6537-6. MR2186122

[7] E. G. Birgin and J. M. Martínez, *On the application of an augmented Lagrangian algorithm to some portfolio problems*, EURO J. Comput. Optim. **4** (2016), no. 1, 79–92, DOI 10.1007/s13675-015-0052-9. MR3500983

[8] E. G. Birgin, J. M. Martínez, and M. Raydan, *Spectral projected gradient methods: Review and perspectives*, Journal of Statistical Software **60** (2014), no. 3 (DOI: 10.18637/jss.v060.i03). `http://www.jstatsoft.org/v60/i03`.

[9] L. F. Bueno, A. Friedlander, J. M. Martínez, and F. N. C. Sobral, *Inexact restoration method for derivative-free optimization with smooth constraints*, SIAM J. Optim. **23** (2013), no. 2, 1189–1213, DOI 10.1137/110856253. MR3069098

[10] R. Escalante and M. Raydan, *Dykstra's algorithm for a constrained least-squares matrix problem*, Numer. Linear Algebra Appl. **3** (1996), no. 6, 459–471, DOI 10.1002/(SICI)1099-1506(199611/12)3:6⟨459::AID-NLA82⟩3.3.CO;2-J. MR1419256

[11] A. Fischer and A. Friedlander, *A new line search inexact restoration approach for nonlinear programming*, Comput. Optim. Appl. **46** (2010), no. 2, 333–346, DOI 10.1007/s10589-009-9267-0. MR2644209

[12] M. A. Gomes-Ruggiero, J. M. Martínez, and S. A. Santos, *Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints*, SIAM J. Sci. Comput. **31** (2009), no. 3, 1628–1652, DOI 10.1137/070707828. MR2491539

[13] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl. **103** (1988), 103–118, DOI 10.1016/0024-3795(88)90223-6. MR943997

[14] T. Homem-de-Mello, *Variable-sample methods for stochastic optimization*, ACM Transactions on Modeling and Computer Simulation **13** (2003), 108–133.

[15] C. Y. Kaya and J. M. Martínez, *Euler discretization and inexact restoration for optimal control*, J. Optim. Theory Appl. **134** (2007), no. 2, 191–206, DOI 10.1007/s10957-007-9217-x. MR2332460

[16] N. Krejić and N. Krklec, *Line search methods with variable sample size for unconstrained optimization*, J. Comput. Appl. Math. **245** (2013), 213–231, DOI 10.1016/j.cam.2012.12.020. MR3016246

[17] N. Krejić and N. Krklec Jerinkić, *Nonmonotone line search methods with variable sample size*, Numer. Algorithms **68** (2015), no. 4, 711–739, DOI 10.1007/s11075-014-9869-1. MR3325823

[18] N. Krejić and J. M. Martínez, *Inexact restoration approach for minimization with inexact evaluation of the objective function*, Math. Comp. **85** (2016), no. 300, 1775–1791, DOI 10.1090/mcom/3025. MR3471107

[19] J. M. Martinez, *Inexact-restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming*, J. Optim. Theory Appl. **111** (2001), no. 1, 39–58, DOI 10.1023/A:1017567113614. MR1850678

[20] J. M. Martínez and E. A. Pilotta, *Inexact-restoration algorithm for constrained optimization*, J. Optim. Theory Appl. **104** (2000), no. 1, 135–163, DOI 10.1023/A:1004632923654. MR1741394

[21] R. Pasupathy, *On choosing parameters in Retrospective Approximation algorithms for stochastic root finding and simulation optimization*, Operations Research **58** (2010), 889–901.

[22] J. O. Royset, *On sample size control in sample average approximations for solving smooth stochastic programs*, Comput. Optim. Appl. **55** (2013), no. 2, 265–309, DOI 10.1007/s10589-012-9528-1. MR3053787

[23] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems*: *Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, John Wiley & Sons, Ltd., Chichester, 1993. MR1241645

[24] A. Ruszczyński and A. Shapiro (eds.), *Stochastic Programming*, Handbooks in Operations Research and Management Science, vol. 10, Elsevier Science B.V., Amsterdam, 2003. MR2051791

[25] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming*: *Modeling and Theory*, MPS/SIAM Series on Optimization, vol. 9, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2009. MR2562798

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 address: 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 address: 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 address: martinez@ime.unicamp.br