A general merit function-based global convergent framework for nonlinear optimization*

Ernesto G. Birgin† Luís Felipe Bueno‡ Tiara Martini§ Dimary Moreno¶
Thadeu Alves Senne† Thiago Siqueira‖

November 21, 2024

Abstract

In this paper, we revisit the convergence theory of the inexact restoration paradigm for nonlinear optimization. The paper first identifies the basic elements of a globally convergent method based on merit functions. Then, the inexact restoration method that employs a two-phase iteration is introduced as a special case. A specific implementation is presented that is supported in the solution of regularized subproblems. The proposed inexact restoration method includes more freedom in the computation of iterates and a novel procedure that integrates the computation of the penalty parameter with the optimization phase. Additionally, a way to speed up the proposed method by solving a quadratic programming subproblem is proposed. An alternative interpretation of the presented method would be to say that the method consists of a globally convergent sequential quadratic programming method that divides the iteration into two phases (feasibility and optimality) when the quadratic subproblem is infeasible. Theoretical results include asymptotic convergence theory as well as a worst-case iteration and evaluation complexity analysis of the introduced methods. The paper concludes by presenting numerical experiments that illustrate the practical application of the proposed methods.

Key words: Inexact Restoration, Sequential Quadratic Programming, algorithms, global convergence, complexity, numerical experiments.

1 Introduction

A wide range of applications can be modeled as optimization problems, where the goal is to minimize or maximize a specific objective while adhering to certain constraints. Properly formulating a situation as an optimization problem enables decision-makers to make more informed choices, which is why this approach is increasingly utilized across various fields. The theoretical framework surrounding

*This work has been partially supported by FAPESP (grants 2013/07375-0, 2020/12455-6, 2021/14011-0, 2022/05803-3, and 2023/08706-1), CNPq (grants 302073/2022-1, 311830/2023-4, and 407147/2023-3), and CAPES.

†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

‡Institute of Science and Technology, Federal University of São Paulo, São José dos Campos, SP, Brazil. emails: lfelipebueno@gmail.com, senne@unifesp.br

§Technological Institute of Aeronautics - ITA, São José dos Campos, SP, Brazil. email: tiara@ita.br

¶Department of Mathematics, PUC-Rio, Rua Marquês de São Vicente, 225, Gávea, 22451-900, Rio de Janeiro, RJ, Brazil. email: dimarydelcmoreno@mat.puc-rio.br.

‖Federal Institute of São Paulo, IFSP - Campus Campos do Jordão, Rua Monsenhor José Vita, 280, Vila Abernessia, 12460000, Campos do Jordão, SP, Brazil. email: thiago.siqueira@ifsp.edu.br

optimization is extensive, with new results emerging regularly. Much of the research focuses on the mathematical properties of optimization problems, while others concentrate on developing computational methods to find solutions. Despite the depth of study in this area, there is no universally accepted best technique for solving optimization problems, regardless of their formulation. Most algorithms are iterative, beginning with one or more initial estimates and generating new approximations that ideally improve upon previous ones.

Many solution-finding methods are tailored to specific problems, while others can be applied more broadly but are generally less efficient than specialized approaches. Conversely, general algorithms are often more accessible to researchers who need to tackle various optimization problems without being experts in the field. We can categorize the literature on solution methods into two groups: those with solid convergence theories and those that rely primarily on empirical strategies for approximating solutions. Interestingly, algorithms lacking theoretical guarantees can sometimes outperform more rigorously grounded options for specific problems. However, methods without theoretical backing typically suffer from reduced robustness and may fail in a broader range of scenarios. For methods with stronger theoretical foundations, the analysis of algorithms may depend on the initial point chosen; if success hinges on a starting point close to the solution, we refer to this as local convergence, while methods that guarantee convergence regardless of starting points exhibit global convergence.

This work aims to identify fundamental elements that ensure global convergence for algorithms addressing general smooth continuous optimization problems. We believe this contribution will enhance the abstract understanding of convergence theory in nonlinear programming methods. Such progress is crucial for analyzing how various specific proposals might fit within a unified theoretical framework that guarantees strong convergence properties. This integration would not only facilitate the development of a cohesive convergence theory for various general optimization methods but also allow for the incorporation of specialized strategies tailored to particular problems that may lack a well-established theoretical basis.

The development of methods that necessitate full restoration of infeasibility traces back to Rosen's gradient projection method, proposed in [50], which extends his earlier work [49] on linear constraints. The restoration process in [50] occurs in a direction orthogonal to the tangent approximation of the constraints used during the minimization phase. In a similar vein, Miele and his collaborators have proposed additional approaches, such as [46], where restoration occurs through a sequence of steps, each orthogonal to the linearization of constraints at intermediate points. The generalized reduced gradient (GRG) method offers another option, projecting the gradient onto a space tangent to the constraints. However, GRG employs the Implicit Function Theorem to differentiate between dependent and independent variables, as described in [54] for linear constraints. Restoration is achieved by determining the values of dependent variables based on the others derived during the optimization phase. A more detailed convergence analysis of Miele's work is presented by Rom and Avriel in [47] and [48], along with a discussion of the distinctions among the methods mentioned.

Several other methods utilizing constraint linearizations adopt a two-phase optimization process: one focused on optimality and the other on feasibility. The steps in these methods are often labeled as tangent (for optimality) and normal (for feasibility), with some literature referring to them as horizontal and vertical. This approach is exemplified in Sequential Linear Programming (SLP) methods [38], Sequential Quadratic Programming (SQP) [34, 33], Interior Point methods [52], and Cylindric Dynamic Control of Infeasibility [14], among others. Unlike GRG-type methods, some algorithms for minimizing nonlinear constraints do not require exact feasibility restoration at each iteration. This can be advantageous when dealing with feasible sets exhibiting large curvature, which would necessitate substantial effort to restore feasibility even when far from the solution.

The philosophy of addressing feasibility and optimality in distinct phases during each iteration characterizes an important class of methods known as Inexact Restoration (IR), which is of particular

interest here. These concepts are rooted in [46, 50], and the initial studies using the IR terminology include [45] and [44]. The most recent IR methods base their convergence theory on the techniques presented in [32] and the complexity theory of [29]. In these methods, a point undergoes a partial restoration of feasibility in the first phase, termed the restoration phase. In the second phase, optimality is enhanced relative to the restored point, typically by remaining within the tangent space of the constraints to manage feasibility deterioration. This approach allows for the generation of a superior point compared to the initial one. Such methods have been successfully applied to numerous significant applications, including problems associated with molecular studies in computational physical chemistry [35, 36, 37], demand adjustment problems [53], hard sphere problems [39, 41], and optimal control problems [10, 11, 42]. The IR paradigm has also been used recently in problems whose functions are subject to inaccurate evaluations [18, 19, 28, 43, 12, 13]. Other classes of problems tackled with IR methods include derivative-free optimization [9, 25, 31] and bilevel optimization [3, 7, 27]. In addition, some articles report on the reliability of IR methods for solving general nonlinear programming problems, as in [15, 22].

When the iterates produced by an algorithm may be infeasible, a mechanism beyond just the objective function is required to determine the relative merit of different points. One alternative for ensuring global convergence in IR methods is the use of filter techniques, see [51]. Another common strategy involves reducing a merit function that combines the objective function with the constraints. In [45], the merit function is a direct combination of the objective function and the measure of infeasibility. Meanwhile, [44] explores the use of Lagrange multipliers, employing the Sharp Lagrangian— a convex combination of the Lagrangian and the norm of the constraints—as the merit function. That paper argues that this method can expedite algorithm convergence by permitting larger steps, since second-order optimality conditions engage the Lagrangian in the tangent space of the constraints rather than the original function. A more recent work that also follows this line is [26]. Our approach aligns with this perspective, also utilizing the Sharp Lagrangian as a merit function.

In summary, our contributions are organized as follows: In Section 2, we identify key elements that ensure the global convergence of optimization methods utilizing the Sharp Lagrangian as a merit function. In Section 3, we demonstrate how general IR strategies can be incorporated into the overarching globalization framework. In Section 4, we propose a specific alternative to meet the IR steps along with an acceleration procedure, using quadratic models with regularization and presenting complexity results for the algorithm. The acceleration is based on a sequential quadratic programming strategy and preserves the convergence properties of the method. In Section 5, we illustrate through numerical experiments how the acceleration procedure can speed up the resolution of some problems. The final considerations of this work are presented in Section 6.

Notation. Given a symmetric matrix $H \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(H)$ denotes its smaller eigenvalue.

2 Globally convergent model algorithm

In this work we consider the optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are twice continuously differentiable and Ω is a non-empty compact convex polytope. In this section, we present the key elements that we have identified for the globalization of a variety of methods that can be applied to the solution of problem (1). The analysis is driven by the convergence theory of IR methods, but it does not use conditions based on the so called "restored point", which means that the theory can be used for other types of method, including those that do not use iterations with two phases.

The analysis begins with the definition of a minimalist model algorithm. The description of the algorithm requires the definition of the Lagrangian function given by

$$L(x, \lambda) = f(x) + h(x)^T \lambda,$$

where $\lambda \in \mathbb{R}^m$ represents the Lagrange multipliers associated with the constraints h, and the definition of the merit function

$$\Phi(x, \lambda, \theta) = \theta L(x, \lambda) + (1 - \theta)\|h(x)\|, \tag{2}$$

where $\theta \in (0, 1)$ plays the role of a penalty parameter. For further reference, we also define

$$c(x) = \frac{1}{2}\|h(x)\|_2^2.$$

Only two conditions are required for the general convergence analysis. The first requires that the merit function decreases by at least an amount proportional to the infeasibility measure at the current iterate. The second requires that the value of the Lagrangian in the new iterate be no greater than the current value plus an increment also proportional to the infeasibility minus an amount of the order of the square of the step. Below we formally state the model algorithm. In the sequel, we show that the first condition guarantees asymptotic feasibility and the second, when combined with the first, ensures that the step size between iterates tends to zero.

Algorithm 2.1. Let $\Lambda \subset \mathbb{R}^m$ be a given non-empty compact set. Let $\bar{\theta} > 0$, $\alpha_L > 0$, $\alpha_\Phi > 0$, $\tilde{\alpha} > 0$, $x^0 \in \Omega$, $\lambda^0 \in \Lambda$, and $\theta_0 \in (0, 1)$ be given. Set $k \leftarrow 0$.

Step 1. Compute $x^{k+1} \in \Omega$, $\lambda^{k+1} \in \Lambda$, and θ_{k+1} satisfying

$$\bar{\theta} \leq \theta_{k+1} \leq \theta_k, \tag{3}$$

$$\Phi(x^{k+1}, \lambda^{k+1}, \theta_{k+1}) \leq \Phi(x^k, \lambda^k, \theta_{k+1}) - \alpha_\Phi \|h(x^k)\|, \tag{4}$$

and

$$L(x^{k+1}, \lambda^{k+1}) \leq L(x^k, \lambda^k) - \alpha_L \|x^{k+1} - x^k\|^2 + \tilde{\alpha}\|h(x^k)\|. \tag{5}$$

Step 2. Set $k \leftarrow k + 1$ and go to Step 1.

Remark. In the sections that follow, we will show specific ways of computing $x^{k+1} \in \Omega$, $\lambda^{k+1} \in \Lambda$, and θ_{k+1}. For that choices, we will show that there exist parameters $\bar{\theta} > 0$ and $\tilde{\alpha} > 0$ for which (3) and (5) hold. In other words, these two parameters will cease to be parameters of the algorithm to be theoretical constants that are guaranteed to exist.

The result below (Lemma 2.1) shows that the infeasibility of a sequence generated by Algorithm 2.1 is summable. This type of result is usual in many IR algorithms and the proof we present is essentially the one given in [26]. We have chosen to keep it here to emphasize that only conditions (3) and (4) are necessary, disconnecting them from the algorithm presented in [26]. In addition, we have more freedom of choice in the parameter α_Φ than in [26] and we present a bound on the computational effort to achieve a certain degree of feasibility, as done in [29], but without relying on Lagrange multipliers.

Assumption A1 *The set $\Omega \subset \mathbb{R}^n$ is convex and compact. The set $\Lambda \subset \mathbb{R}^m$ is compact. Functions f and h are L_f- and L_h-continuously differentiable, respectively. That is, there exist constants $L_f > 0$ and $L_h > 0$ such that, for all $x, y \in \Omega$,*

$$\|\nabla f(y) - \nabla f(x)\| \leq L_f \|y - x\| \tag{6}$$

and

$$\|J_h(y) - J_h(x)\| \leq L_h\|y - x\|, \tag{7}$$

where $J_h : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ *represents the Jacobian of* h.

For future reference, we mention here the existence of some constants guaranteed by Assumption A1. By the continuity of f, h, and J_h plus the compacity of Ω, there exist constants C_f, C_h, and D_Ω such that for all $x, y \in \Omega$,

$$\max\{|f(x)|, \|\nabla f(x)\|\} \leq C_f, \tag{8}$$

$$\max\{\|h(x)\|, \|J_h(x)\|\} \leq C_h, \tag{9}$$

and

$$\|y - x\| \leq D_\Omega. \tag{10}$$

Moreover, since f is continuously differentiable, by the convexity and compacity of Ω, we can apply the Mean Value Theorem to obtain that, for all $x, y \in \Omega$,

$$|f(y) - f(x)| \leq C_f\|y - x\|. \tag{11}$$

In addition, using the Fundamental Theorem of Calculus, constant L_h from Assumption A1 is such that, for all $x, y \in \Omega$,

$$\|h(y) - h(x)\| \leq \|J_h(x)(y - x)\| + L_h\|y - x\|^2 \tag{12}$$

and

$$\|h(y)\| \leq \|h(x) + J_h(x)(y - x)\| + L_h\|y - x\|^2. \tag{13}$$

Using that Λ is a compact set and combing (6), (7), (8), (9), (10), (11) and (12), there exist constants $\bar{\lambda}$, C_L, and L_L such that, for all $x, y \in \Omega$ and all $\lambda, \mu \in \Lambda$,

$$\|\lambda\| \leq \bar{\lambda}, \tag{14}$$

$$|L(x, \lambda)| \leq C_L, \tag{15}$$

$$|L(y, \lambda) - L(x, \lambda)| \leq L_L\|y - x\|, \tag{16}$$

$$\|\nabla L(y, \lambda) - \nabla L(x, \lambda)\| \leq L_L\|y - x\|, \tag{17}$$

and, by (17),

$$L(y, \lambda) \leq L(x, \lambda) + \nabla L(x, \lambda)^T(y - x) + L_L\|y - x\|^2, \tag{18}$$

where $\nabla L(x, \lambda)$ is the gradient of $L(x, \lambda)$ with respect to x. Moreover, there exists L_c such that for all $x, y \in \Omega$,

$$\|\nabla c(x) - \nabla c(y)\| \leq L_c\|x - y\| \tag{19}$$

and

$$c(y) \leq c(x) + \nabla c(x)^T(y - x) + L_c\|y - x\|^2. \tag{20}$$

Lemma 2.1 *Suppose that Assumption A1 holds. Let* $\{x^k\}$ *be a sequence generated by Algorithm 2.1. Then, there exists* $\bar{h} > 0$ *such that for all* $k \in \mathbb{N}$

$$\sum_{j=0}^{k} \|h(x^j)\| \leq \bar{h}. \tag{21}$$

In addition, given $\varepsilon_{\text{feas}} > 0$, *the number of iterations* k *such that* $\|h(x^k)\| > \varepsilon_{\text{feas}}$ *is limited by* $\bar{h}/\varepsilon_{\text{feas}}$ *and* $\lim_{k\to\infty} h(x^k) = 0$.

Proof: By (3), we have that $1 > \theta_j \geq \theta_{j+1} \geq \bar{\theta} > 0$ for all j. Therefore, by defining $\eta_j = (1-\theta_j)/\theta_j = 1/\theta_j - 1$, we have that

$$0 < \eta_j \leq \eta_{j+1} \leq 1/\bar{\theta} - 1 < 1/\bar{\theta}$$

for all j. Thus,

$$\sum_{j=0}^{k-1}(\eta_{j+1} - \eta_j) = \eta_k - \eta_0 < \eta_k < 1/\bar{\theta}. \tag{22}$$

By the definition (2) of Φ, conditions (3) and (4), the fact that $\theta_j \in (0,1)$ for all j and the definition of η_j, we have that, for all j,

$$L(x^{j+1}, \lambda^{j+1}) + \eta_{j+1}\|h(x^{j+1})\| \leq L(x^j, \lambda^j) + \eta_{j+1}\|h(x^j)\| - \alpha_\Phi \|h(x^j)\|.$$

Adding and subtracting $\eta_j\|h(x^j)\|$ to the second term and doing a telescoping sum we get

$$L(x^k, \lambda^k) + \eta_k\|h(x^k)\| \leq L(x^0, \lambda^0) + \eta_0\|h(x^0)\| + \sum_{j=0}^{k-1}(\eta_{j+1} - \eta_j)\|h(x^j)\| - \alpha_\Phi \sum_{j=0}^{k-1}\|h(x^j)\|.$$

Thus, by (9), (15), (22), and the positivity of η_j for all j, we get

$$\sum_{j=0}^{k}\|h(x^j)\| \leq \frac{1}{\alpha_\Phi}\left(2C_L + \eta_0 C_h + \frac{C_h}{\bar{\theta}} + \alpha_\Phi C_h\right) \equiv \bar{h}. \tag{23}$$

The limit on the number of iterations k such that $\|h(x^k)\| > \varepsilon_{\text{feas}}$ follows from (23), and $\lim_{k\to\infty} h(x^k) = 0$ follows from the fact that $\varepsilon_{\text{feas}} > 0$ is arbitrary. □

The following lemma shows that the step size tends to zero. This result is different from the typical results in IR methods because it does not use the restored point, as is usually done.

Lemma 2.2 *Suppose that Assumption A1 holds. Let $\{x^k\}$ be a sequence generated by Algorithm 2.1. Then, for all k,*

$$\sum_{j=0}^{k}\|x^{j+1} - x^j\|^2 \leq \frac{2C_L + \tilde{\alpha}\bar{h}}{\alpha_L}, \tag{24}$$

where $\bar{h}$ is such that (21) *holds and C_L satisfies* (15)*. Additionally, given $\varepsilon_{\text{opt}} > 0$, the number of iterations k such that $\|x^{k+1} - x^k\| > \varepsilon_{\text{opt}}$ is bounded by $((2C_L + \tilde{\alpha}\bar{h})/\alpha_L)\varepsilon_{\text{opt}}^{-2}$ and, consequently,* $\lim_{k\to\infty}\|x^{k+1} - x^k\| = 0$.

Proof: Since (5) holds for all k, we have that

$$L(x^{k+1}, \lambda^{k+1}) - L(x^0, \lambda^0) \leq -\alpha_L \sum_{j=0}^{k}\|x^{j+1} - x^j\|^2 + \tilde{\alpha}\sum_{j=0}^{k}\|h(x^j)\|.$$

Therefore, (24) follows from Lemma 2.1 and (15). The bound on the number of iterations such that $\|x^{k+1} - x^k\| > \varepsilon_{\text{opt}}$ and $\lim_{k\to\infty}\|x^{k+1} - x^k\| = 0$ follows trivially from this result. □

3 Inexact Restoration as a particular case

In this section we present a particular case of Algorithm 2.1 that computes x^{k+1}, λ^{k+1} and θ_{k+1} with a two-phase strategy. In the first phase, an intermediate point y^k is computed, partially restoring the feasibility with respect to x^k. This mechanism captures the essential elements of an iteration of Inexact Restoration methods.

Algorithm 3.1. Let $\Lambda \subset \mathbb{R}^m$ be a given non-empty compact set. Let $\bar{\theta} > 0$, $\alpha_L > 0$, $\alpha_\Phi > 0$, $\beta_f > 0$, $r \in (0,1)$, $x^0 \in \Omega$, $\lambda^0 \in \Lambda$, and $\theta_0 \in (0,1)$ be given. Set $k \leftarrow 0$.

Step 1. *Two-phase iteration.*

Step 1.1. *Phase I: Restoration phase.*
Compute $y^k \in \Omega$ satisfying

$$\|h(y^k)\| \leq r\|h(x^k)\| \tag{25}$$

and

$$f(y^k) \leq f(x^k) + \beta_f \|h(x^k)\|. \tag{26}$$

Step 1.2. *Phase II: Optimization phase integrated with penalty parameter update.*
Compute $x^{k+1} \in \Omega$, $\lambda^{k+1} \in \Lambda$, and θ_{k+1} satisfying (3), (4), and

$$L(x^{k+1}, \lambda^{k+1}) \leq L(y^k, \lambda^{k+1}) - \alpha_L \|x^{k+1} - y^k\|^2. \tag{27}$$

Step 2. Set $k \leftarrow k+1$ and go to Step 1.

The following three lemmas show that Algorithm 3.1 is a particular case of Algorithm 2.1. Specifically, they show that, for every iteration k, x^{k+1} and λ^{k+1} generated by Algorithm 3.1 satisfy (5) with a particular choice of $\bar{\alpha}$, since (3) and (4) hold by the definition of the algorithm.

Lemma 3.1 *Suppose that Assumption A1 holds. Let $x^k, y^k \in \Omega$ be such that* (25) *and* (26) *hold. Then, for every $\lambda^k, \lambda \in \Lambda$,*

$$L(y^k, \lambda) \leq L(x^k, \lambda^k) + \bar{\beta}\|h(x^k)\| \tag{28}$$

with $\bar{\beta} = \beta_f + (r+1)\bar{\lambda}$, where $\bar{\lambda}$ satisfies (14).

Proof: By the definition of the Lagrangian, (25), (26), and (14), we have that

$$\begin{aligned} L(y^k, \lambda) - L(x^k, \lambda^k) &= f(y^k) - f(x^k) + h(y^k)^T\lambda - h(x^k)^T\lambda^k \\ &\leq \beta_f\|h(x^k)\| + \|\lambda\|\|h(y^k)\| + \|\lambda^k\|\|h(x^k)\| \\ &\leq \beta_f\|h(x^k)\| + \bar{\lambda}r\|h(x^k)\| + \bar{\lambda}\|h(x^k)\|. \end{aligned}$$

Therefore, (28) holds with the aforementioned $\bar{\beta}$. □

Lemma 3.2 *Suppose that Assumption A1 holds. Let $\{x^k\}$, $\{y^k\}$, and $\{\lambda^k\}$ be generated by Algorithm 3.1. Then, for all $k \in \mathbb{N}$,*

$$L(x^{k+1}, \lambda^{k+1}) \leq L(x^k, \lambda^k) - \alpha_L\|x^{k+1} - y^k\|^2 + \bar{\beta}\|h(x^k)\|, \tag{29}$$

where $\bar{\beta} = \beta_f + (r+1)\bar{\lambda}$ and $\bar{\lambda}$ satisfies (14).

Proof: Condition (29) follows from applying (28) with $\lambda = \lambda^{k+1}$ in the right-hand side of (27). □

Corollary 3.1 *Suppose that Assumption A1 holds. Let $\{x^k\}$ and $\{y^k\}$ be sequences generated by Algorithm 3.1. Then, there exists $\bar{h} > 0$ such that, for all $k \in \mathbb{N}$,* (21) *holds and*

$$\sum_{j=1}^{k} \|x^{j+1} - y^j\|^2 \leq \frac{2C_L + \bar{\beta}\bar{h}}{\alpha_L}. \tag{30}$$

In addition, given $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$, the number of iterations k such that $\|h(x^k)\| > \varepsilon_{\text{feas}}$ or $\|x^{k+1} - y^k\| > \varepsilon_{\text{opt}}$ is limited by

$$\frac{\bar{h}}{\varepsilon_{\text{feas}}} + \frac{2C_L + \bar{\beta}\bar{h}}{\alpha_L \varepsilon_{\text{opt}}^2} + 1 = O(\varepsilon_{\text{feas}}^{-1} + \varepsilon_{\text{opt}}^{-2})$$

and, consequently, $\lim_{k\to\infty} h(x^k) = 0$ and $\lim_{k\to\infty} \|x^{k+1} - y^k\| = 0$.

Proof: Note that Lemma 2.1 also holds for the sequence $\{x^k\}$ generated by Algorithm 3.1, since it only requires the conditions (3) and (4) to hold. On the other hand, (30) can be obtained exactly like (24) in Lemma 2.2, substituting $\|x^{j+1} - x^j\|$ with $\|x^{j+1} - y^j\|$, $\tilde{\alpha}$ with $\bar{\beta}$, and applying (29) from Lemma 3.2. The limit on the number of iterations is an immediate consequence of the maximum number of iterations in which each condition applies separately. □

Up to this point, the fact of Algorithm 3.1 being a particular case of Algorithm 2,1 was not yet established. This will be done in Lemma 3.3 below, with the additional hypothesis (31). However, Lemma 3.2 and Corollary 3.1 summarize the properties of Algorithm 3.1 without the need of this additional hypothesis.

Lemma 3.3 *Suppose that Assumption A1 holds. Let $\{x^k\}$ and $\{y^k\}$ be generated by Algorithm 3.1. Assume that there exists $\beta > 0$ such that, for all k, y^k computed at Step 1.1 is such that*

$$\|x^k - y^k\| \leq \beta \|h(x^k)\|. \tag{31}$$

Then, for all k, (5) *holds with $\tilde{\alpha} = \bar{\beta} + 3\alpha_L D_\Omega \beta$, where D_Ω satisfies* (10).

Proof: By Cauchy-Schwarz, we have that

$$\begin{aligned} \|x^{k+1} - x^k\|^2 &= \|x^{k+1} - y^k\|^2 + \|y^k - x^k\|^2 + 2\langle x^{k+1} - y^k, y^k - x^k\rangle \\ &\leq \|x^{k+1} - y^k\|^2 + \|y^k - x^k\|^2 + 2\|x^{k+1} - y^k\|\|y^k - x^k\|. \end{aligned}$$

Therefore, by (31) and (10),

$$\begin{aligned} -\|x^{k+1} - y^k\|^2 &\leq -\|x^{k+1} - x^k\|^2 + \|y^k - x^k\|^2 + 2\|x^{k+1} - y^k\|\|y^k - x^k\| \\ &\leq -\|x^{k+1} - x^k\|^2 + 3D_\Omega \beta \|h(x^k)\|. \end{aligned}$$

Thus, (5) follows from (29) with $\tilde{\alpha} = \bar{\beta} + 3\alpha_L D_\Omega \beta$. □

We end this section by showing that (31) implies (26). With that, we could have defined Algorithm 3.1 with (31) in place of (26). This algorithm would be a less flexible algorithm for which the same results that were given in this section for Algorithm 3.1 would apply.

Lemma 3.4 *Suppose that Assumption A1 holds. Let $\{x^k\}$ and $\{y^k\}$ be sequences such that* (31) *holds for some $\beta > 0$. Then* (26) *holds with $\beta_f = C_f \beta$, where C_f satisfies* (8).

Proof: By (11) and (31), we have that

$$f(y^k) \leq f(x^k) + |f(y^k) - f(x^k)| \leq f(x^k) + C_f \|y^k - x^k\| \leq f(x^k) + C_f \beta \|h(x^k)\|.$$

□

4 Practical Inexact Restoration alternative

In this section we present a particular case of Algorithm 3.1. Specifically, we present a practical two-phase alternative to compute x^{k+1}, λ^{k+1} and θ_{k+1} that satisfy (3), (4), and (27). As the suggested alternative satisfies the hypothesis (31) of Lemma 3.3, the algorithm presented is in turn a particular case of Algorithm 2.1. The procedures we use are essentially those used in [29], but with a little more freedom in the algorithmic choices, some of which have already been considered in [28] for problems with inaccurate evaluations of the functions. The novelty lies in an innovative way of estimating the Lagrange multipliers, resulting from the integration of the optimization phase and the calculation of the penalty parameter.

The algorithm will be built using three modules. An additional module suggests a possible acceleration procedure. Each module is presented first and the algorithm as a whole is presented at the end of the section.

4.1 Restoration phase module

Algorithm 4.1 presented below is a practical approach to computing, in Step 1.1 of Algorithm 3.1, $y^k \in \Omega$ that satisfies (25) and (26). To do this, the algorithm minimizes $c(x)$ in Ω using a regularization technique. The algorithm ends when it either obtains a restored point y^k that satisfies (25) or finds an infeasible point y^k that is approximately stationary of infeasibility, with some established tolerance. In the latter case, we declare that the restoration phase failed and we interrupt the search for a solution of problem (1) by declaring that the problem may be infeasible. In the former case, Lemma 4.4 shows that y^k satisfies (31) for a certain value of β, which in turn implies, by Lemma 3.4, that (26) holds for a certain value of β_f. This proves that Algorithm 4.1 is indeed a way to implement Step 1.1 of Algorithm 3.1.

Algorithm 4.1. Let $x^k \in \Omega$, $r \in (0,1)$, $0 < r_{\text{feas}} \ll r$, $0 < \sigma_{\min} \leq \sigma_{\max}$, $M > 0$, $\beta_c \geq 0$, $\kappa_R > 0$, $\alpha_R > 0$, $\kappa_\varphi > 1$, and $1 < \tau_1 \leq \tau_2$ be given.

Step 1. Compute

$$c_{\text{target}} = \frac{1}{2} r^2 \|h(x^k)\|^2 \ \text{ and } \ \epsilon_c = r_{\text{feas}} \|h(x^k)\|.$$

Step 2. Initialize $\ell \leftarrow 0$ and choose $z^0 \in \Omega$ such that $\|h(z^0)\| \leq \|h(x^k)\|$ and $\|z^0 - x^k\| \leq \beta_c \|h(x^k)\|$.

Step 3. If $c(z^\ell) \leq c_{\text{target}}$ or $\|P_\Omega(z^\ell - \nabla c(z^\ell)) - z^\ell\| \leq \epsilon_c$, then stop returning $y^k \equiv z^\ell$.

Step 4. Initialize $j \leftarrow 0$, choose $\sigma_{\ell j} \in [0, \sigma_{\max}]$ and $B_\ell \in \mathbb{R}^{n \times n}$ symmetric such that $\|B_\ell\| \leq M$ and $\lambda_{\min}(B_\ell + \sigma_{\ell j} I) \geq 1/M$.

Step 5. By approximately solving the problem

$$\text{Minimize } \nabla c(z^\ell)^T (z - z^\ell) + \frac{1}{2}(z - z^\ell)^T B_\ell (z - z^\ell) + \frac{\sigma_{\ell j}}{2} \|z - z^\ell\|^2 \ \text{ subject to } \ z \in \Omega, \tag{32}$$

find $z^{\ell j} \in \Omega$ satisfying

$$\nabla c(z^\ell)^T (z^{\ell j} - z^\ell) + \left(1 + \frac{1}{\kappa_\varphi}\right) \frac{1}{2} (z^{\ell j} - z^\ell)^T (B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell) \leq 0 \tag{33}$$

and

$$\left\| P_\Omega \left(z^{\ell j} - \left[\nabla c(z^\ell) + (B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell) \right] \right) - z^{\ell j} \right\| \leq \kappa_R \|z^{\ell j} - z^\ell\|. \tag{34}$$

Step 6. Consider the condition

$$c(z^{\ell j}) \leq c(z^\ell) - \alpha_R \|z^{\ell j} - z^\ell\|^2. \tag{35}$$

If (35) does not hold, then compute $\sigma_{\ell,j+1} = \max\{\sigma_{\min}, \tau\sigma_{\ell j}\}$ for some $\tau \in [\tau_1, \tau_2]$, set $j \leftarrow j+1$, and go to Step 5.

Step 7. Define $z^{\ell+1} = z^{\ell j}$, set $\ell \leftarrow \ell + 1$, and go to Step 3.

Remark. In Step 2, $z^0 = x^k$ is a natural choice. However, other problem-dependent choices are possible provided they satisfy the required conditions.

The next result shows that the loop in Steps 5 and 6 ends in finite time.

Lemma 4.1 *Suppose that Assumption A1 holds. Then, for every $\ell \geq 0$ and $j \geq 0$, if $z^{\ell j}$ is computed at Step 5 of Algorithm 4.1 with $\sigma_{\ell j} \geq 2(L_c + \|B_\ell\|/2 + \alpha_R)$, where L_c satisfies* (20)*, then* (35) *holds. Moreover, for every $\ell \geq 0$ and $j \geq 0$,*

$$\sigma_{\ell j} \leq \max\left\{2\tau_2\left(L_c + \frac{\|B_\ell\|}{2} + \alpha_R\right), \sigma_{\max}\right\}. \tag{36}$$

Proof: By (20) and the fact that condition (33) implies that $\nabla c(z^\ell)^T(z^{\ell j} - z^\ell) + \frac{1}{2}(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell) \leq 0$, we have that

$$\begin{aligned} c(z^{\ell j}) - c(z^\ell) &\leq \nabla c(z^\ell)^T(z^{\ell j} - z^\ell) + L_c\|z^{\ell j} - z^\ell\|^2 \\ &= \nabla c(z^\ell)^T(z^{\ell j} - z^\ell) + \tfrac{1}{2}(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell) \\ &\quad -\tfrac{1}{2}(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell) + L_c\|z^{\ell j} - z^\ell\|^2 \\ &\leq (L_c + \|B_\ell\|/2 - \sigma_{\ell j}/2)\,\|z^{\ell j} - z^\ell\|^2. \end{aligned}$$

Therefore, (35) holds whenever $\sigma_{\ell j} \geq 2(L_c + \|B_\ell\|/2 + \alpha_R)$, and (36) follows from the way of choosing $\sigma_{\ell 0}$ and $\sigma_{\ell j}$ for $j \geq 1$. □

Lemma 4.1, the initialization of $\sigma_{\ell 0}$ and the way of computing $\sigma_{\ell,j+1}$ for $j \geq 0$ imply that, at each iteration ℓ of Algorithm 4.1, the loop of Steps 5 and 6 is executed a maximum of $\log_{\tau_1}(2(L_c + \|B_\ell\|/2 + \alpha_R)/\sigma_{\min}) + 2 \leq \log_{\tau_1}(2(L_c + M/2 + \alpha_R)/\sigma_{\min}) + 2 = O(1)$ times. Each execution of the loop does exactly one evaluation of h (needed to evaluate c). Lemma 4.3, which uses the result of Lemma 4.2, gives an upper bound on the number of iterations needed for Algorithm 4.1 to find an iterate z^ℓ that satisfies the stopping condition of Step 3. The bound on the number of iterations implies also a bound on the number of evaluations of h and of the Jacobian of h.

Lemma 4.2 *Suppose that Assumption A1 holds. For every $\ell \geq 0$,*

$$\left\|P_\Omega\left(z^{\ell+1} - \nabla c(z^{\ell+1})\right) - z^{\ell+1}\right\| \leq [3\tau_2\,(L_c + M + \kappa_R + \alpha_R) + \sigma_{\max}]\,\|z^{\ell+1} - z^\ell\|, \tag{37}$$

where L_c satisfies (19)*.*

Proof: Let $v = z^{\ell j} - \nabla c(z^{\ell j})$ and $u = z^{\ell j} - [\nabla c(z^\ell) + B_\ell(z^{\ell j} - z^\ell) + \sigma_{\ell j}(z^{\ell j} - z^\ell)]$. Then,

$$\begin{aligned} \|v - u\| &= \left\|\nabla c(z^\ell) - \nabla c(z^{\ell j}) + B_\ell(z^{\ell j} - z^\ell) + \sigma_{\ell j}(z^{\ell j} - z^\ell)\right\| \\ &\leq \|\nabla c(z^\ell) - \nabla c(z^{\ell j})\| + \|B_\ell\|\|z^{\ell j} - z^\ell\| + \sigma_{\ell j}\|z^{\ell j} - z^\ell\| \\ &\leq (L_c + M + \sigma_{\ell j})\|z^{\ell j} - z^\ell\|. \end{aligned}$$

Therefore, using the non-expansivity of projections and (34), we obtain

$$\begin{aligned}\left\|P_\Omega(z^{\ell j} - \nabla c(z^{\ell j})) - z^{\ell j}\right\| &= \|P_\Omega(v) - P_\Omega(u) + P_\Omega(u) - z^{\ell j}\| \\ &\leq \|v - u\| + \kappa_R \|z^{\ell j} - z^\ell\| \\ &\leq (L_c + M + \sigma_{\ell j} + \kappa_R)\|z^{\ell j} - z^\ell\|.\end{aligned}$$

Thus, by (36), since $z^{\ell+1} = z^{\ell j}$ for some j, we obtain (37). □

Lemma 4.3 *Suppose that Assumption A1 holds. Then, the number of iterations required by Algorithm 4.1 to compute z^ℓ that satisfies the stopping criterion in Step 3 is no greater than*

$$N_R = \frac{\left(3\tau_2(L_c + M + \alpha_R + \kappa_R) + \sigma_{\max}\right)^2}{2\,\alpha_R\, r_{\text{feas}}^2} = O(1),$$

where L_c satisfies (19)*. Since the Jacobian of h, needed to evaluate ∇c, is evaluated once per iteration, the same limit applies to the number of evaluations of the Jacobian of h. In addition, the number of evaluations of h (needed to evaluate c) is limited by*

$$N_R^h = N_R \left[\log_{\tau_1}\left(\frac{2(L_c + M/2 + \alpha_R)}{\sigma_{\min}}\right) + 2\right] = O(1). \tag{38}$$

Proof: Assume that

$$\|P_\Omega(z^s - \nabla c(z^s) - z^s\| > \epsilon_c \tag{39}$$

for all $s \in \{1, \ldots, \ell\}$. By (37) and (39), we have that

$$\ell\,\epsilon_c^2 \leq \sum_{s=1}^{\ell} \|P_\Omega(z^s - \nabla c(z^s)) - z^s\|^2 \leq (3\tau_2\,(L_c + M + \kappa_R + \alpha_R) + \sigma_{\max})^2 \sum_{s=1}^{\ell} \|z^s - z^{s-1}\|^2.$$

Therefore, by (35),

$$c(z^\ell) - c(z^0) = \sum_{s=1}^{\ell} \left[c(z^s) - c(z^{s-1})\right] \leq -\alpha_R \sum_{s=1}^{\ell} \|z^s - z^{s-1}\|^2 \leq -\frac{\alpha_R\,\ell\,\epsilon_c^2}{(3\tau_2\,(L_c + M + \kappa_R + \alpha_R) + \sigma_{\max})^2}.$$

Thus, since z^0 is such that $c(z^0) \leq c(x^k)$ and $\epsilon_c^2 = 2r_{\text{feas}}^2 c(x^k)$, we have that

$$c(z^\ell) \leq c(x^k) - \ell \left\{\frac{2\,\alpha_R\, r_{\text{feas}}^2 c(x^k)}{(3\tau_2\,(L_c + M + \kappa_R + \alpha_R) + \sigma_{\max})^2}\right\}. \tag{40}$$

If $\ell \geq N_R$, then the right-hand side of (40) is non-positive and, consequently, $c(z^\ell) \leq c_{\text{target}}$. This means that in at most N_R iterations at least one of the two conditions in Step 3 is met. □

We end this section by showing that, if Algorithm 4.1 ends with success, then the computed restored point $y^k \equiv z^\ell$ satisfies (31) for a certain value of β.

Lemma 4.4 *Suppose that Assumption A1 holds. For all $\ell \geq 0$ and $j \geq 0$,* (31) *holds with $y^k \equiv z^{\ell j}$ and*

$$\beta = \beta_c + N_R M \kappa_\varphi C_h, \tag{41}$$

where C_h satisfies (9)*.*

Proof: Let $\varphi(t)$ be the univariate function defined as the objective function of (32) evaluated at $z^\ell + t(z^{\ell j} - z^\ell)$, i.e.,

$$\varphi(t) = t\nabla c(z^\ell)^T(z^{\ell j} - z^\ell) + \frac{t^2}{2}(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell).$$

By the definition of the algorithm, $z^{\ell j}$ satisfies (33), that can be rewritten as $\varphi(1) \leq \varphi(1/\kappa_\varphi)$. If $z^{\ell j} \neq z^\ell$, since $\varphi(t)$ is convex and $1/\kappa_\varphi < 1$, then the minimizer $t^\star$ of φ, given by

$$t^\star = -\frac{\nabla c(z^\ell)^T(z^{\ell j} - z^\ell)}{(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell)},$$

satisfies $t^\star \geq 1/\kappa_\varphi$. Therefore, by Cauchy-Schwarz and $\lambda_{\min}(B_\ell + \sigma_{\ell j} I) \geq 1/M$, we have that

$$\frac{1}{\kappa_\varphi} \leq t^* = -\frac{\nabla c(z^\ell)^T(z^{\ell j} - z^\ell)}{(z^{\ell j} - z^\ell)^T(B_\ell + \sigma_{\ell j} I)(z^{\ell j} - z^\ell)} \leq \frac{M}{\|z^{\ell j} - z^\ell\|}\|\nabla c(z^\ell)\|.$$

In consequence, by (9) and (35),

$$\|z^{\ell j} - z^\ell\| \leq \kappa_\varphi M\|\nabla c(z^\ell)\| \leq \kappa_\varphi M\|J_h(z^\ell)^T h(z^\ell)\| \leq \kappa_\varphi M C_h\|h(z^\ell)\| \leq \kappa_\varphi M C_h\|h(x^k)\|.$$

Therefore, since, by definition, $z^{\ell+1} = z^{\ell j}$ for some j, we have that, for all $\ell \geq 0$,

$$\|z^{\ell+1} - z^\ell\| \leq \kappa_\varphi M C_h\|h(x^k)\|. \tag{42}$$

(Note that (42) holds trivially whenever $z^{\ell+1} = z^\ell$.) By Lemma 4.3, the number of iterations performed by Algorithm 4.1 is bounded by N_R. Then, using (42),

$$\|z^\ell - x^k\| \leq \|z^0 - x^k\| + \sum_{s=1}^{\ell}\|z^s - z^{s-1}\| \leq \beta_c\|h(x^k)\| + N_R\kappa_\varphi M C_h\|h(x^k)\|,$$

which implies the desired result. □

Right after Algorithm 4.1, Lemma 4.1 showed that the loop containing Steps 5 and 6 stops in a finite number of iterations. This result is related to the complexity of the algorithm, but also to its well-definiteness. However, a small detail regarding its well-definiteness was left out. We did not show that an approximate solution to problem (32) exists that satisfies (33). This detail can now be explained using the notation introduced in the proof of Lemma 4.4. If z is an exact solution of (32) then $t^\star(z) = 1$ is an exact solution of minimizing $\varphi(t)$ subject to $z^\ell + t(z - z^\ell) \in \Omega$. Therefore, $\varphi(1) \leq \varphi(1/\kappa_\varphi)$ holds. When z tends to the solution of (32), the solution $t^\star(z)$ of minimizing $\varphi(t)$ subject to $z^\ell + t(z - z^\ell) \in \Omega$ is such that $t^\star(z) \to 1$. Therefore, for z sufficiently close to the solution of (32), it must hold that $\varphi(t^\star(z)) \leq \varphi(1/\kappa_\varphi)$.

4.2 Module combining the penalty parameter update and the optimization phase

In this section we describe an algorithm for the optimization phase that integrates a way of determining the penalty parameter of the merit function. Regarding the computation of the penalty parameter, we emphasize here that the process presented can be repeated within the same iteration, whenever convenient, with updated estimates of the Lagrange multipliers. This feature is a special feature because it allows the use of more precise updates of the Lagrange multipliers compared to earlier Inexact Restoration methods that update the penalty parameter with an estimate of the Lagrange

multipliers computed prior to the optimization phase. Regarding the optimization phase, the novelty compared to the procedures in previous works is that we allow the Lagrange multipliers used to solve the subproblem (43) not to be the same as those used in the descent condition (46). This allows the update of these multipliers (see Step 3) to take into account the information from the current optimization process, which is an improvement on previous results from the literature.

Algorithm 4.2 below is intended to be used in Step 1.2 of Algorithm 3.1 to compute x^{k+1}, λ^{k+1}, and θ_{k+1} satisfying (3), (4), and (27). By the definition of the algorithm, $\theta_{k+1} = \theta$ for some θ computed at Step 5. Lemma 4.5 shows that every θ computed at Step 5 is such that (3) holds, for a certain value of $\bar{\theta}$. By the definition of the algorithm, $x^{k+1} = z^j$ and $\lambda^{k+1} = \lambda$ for some z^j computed at Step 2 and λ chosen at Step 3 that satisfy (46) and (49). Then $x^{k+1} = z^j$ and $\lambda^{k+1} = \lambda$ satisfy (4) and (27).

Algorithm 4.2. Let $x^k, y^k \in \Omega$, $\lambda^k \in \Lambda$, $\theta_k \in (0,1)$ $r \in (0,1)$, $\alpha_\Phi \in (0, 1-r)$, $0 < \sigma_{\min} \leq \sigma_{\max}$, $M > 0$, $\kappa_T > 0$, $\kappa_P > 0$, $\alpha_L > 0$, and $1 < \tau_1 \leq \tau_2$ be given.

Step 1. Initialize $j \leftarrow 0$, choose $\sigma_j \in [0, \sigma_{\max}]$, $\lambda' \in \Lambda$, and $H \in \mathbb{R}^{n\times n}$ symmetric such that $\|H\| \leq M$.

Step 2. By approximately solving

$$\underset{x\in\Omega}{\text{Minimize}}\, \nabla L(y^k, \lambda')^T(x-y^k) + \frac{1}{2}(x-y^k)^T H(x-y^k) + \frac{\sigma_j}{2}\|x-y^k\|^2 \text{ subject to } J_h(y^k)(x-y^k) = 0, \tag{43}$$

find $z^j \in \Omega$ satisfying

$$\nabla L(y^k, \lambda')^T(z^j - y^k) + \frac{1}{2}(z^j - y^k)^T(H + \sigma_j I)(z^j - y^k) \leq 0 \tag{44}$$

and

$$\|J_h(y^k)(z^j - y^k)\| \leq \kappa_T \|z^j - y^k\|^2. \tag{45}$$

Step 3. Choose $\lambda \in \Lambda$.

Step 4. Consider the condition

$$L(z^j, \lambda) \leq L(y^k, \lambda) - \alpha_L \|z^j - y^k\|_2^2. \tag{46}$$

If (46) does not hold, then compute $\sigma_{j+1} = \max\{\sigma_{\min}, \tau\sigma_j\}$ for some $\tau \in [\tau_1, \tau_2]$, set $j \leftarrow j+1$, and go to Step 2.

Step 5. Consider condition

$$\Phi(y^k, \lambda, \theta_k) \leq \Phi(x^k, \lambda^k, \theta_k) - \alpha_\Phi \|h(x^k)\|. \tag{47}$$

If (47) holds, then set $\theta = \theta_k$. Otherwise, compute

$$\theta = \frac{(1-\alpha_\Phi)\|h(x^k)\| - \|h(y^k)\|}{L(y^k, \lambda) - L(x^k, \lambda^k) + \|h(x^k)\| - \|h(y^k)\|}. \tag{48}$$

Step 6. Consider condition

$$\Phi(z^j, \lambda, \theta) \leq \Phi(x^k, \lambda^k, \theta) - \alpha_\Phi \|h(x^k)\|. \tag{49}$$

If (49) does not hold, then compute $\sigma_{j+1} = \max\{\sigma_{\min}, \tau\sigma_j\}$ for some $\tau \in [\tau_1, \tau_2]$, set $j \leftarrow j+1$, and go to Step 2.

Step 7. Stop returning $x^{k+1} \equiv z^j$, $\lambda^{k+1} \equiv \lambda$, and $\theta_{k+1} \equiv \theta$.

The following lemma shows the properties of the penalty parameter θ computed in Step 5.

Lemma 4.5 *Suppose that Assumption A1 holds. Suppose also that the input parameters x^k, y^k, and r and λ' chosen at Step 1 are such that (25) and (28) with $\lambda = \lambda'$ hold for some $\bar{\beta}$. Then, the value of θ computed at Step 5 is such that*

$$\Phi(y^k, \lambda', \theta) \leq \Phi(x^k, \lambda^k, \theta) - \alpha_\Phi \|h(x^k)\| \tag{50}$$

holds and

$$0 < \bar{\theta} \equiv (1 - r - \alpha_\Phi)/(\bar{\beta} + 1) \leq \theta \leq \theta_k. \tag{51}$$

Proof: If $h(x^k) = 0$, then, by (25), we have that $h(y^k) = 0$. Therefore, (28) reduces to $L(y^k, \lambda) \leq L(x^k, \lambda^k)$, (47) holds, $\theta = \theta_k$, and (50) holds by the definition of Φ. Assume now that $\|h(x^k)\| > 0$ and consider the function $\widehat{\Phi} : [0, 1] \to \mathbb{R}$ given by

$$\widehat{\Phi}(t) = \Phi(y^k, \lambda, t) - \Phi(x^k, \lambda^k, t) + \alpha_\Phi \|h(x^k)\|.$$

If $\widehat{\Phi}(\theta_k) \leq 0$, then (47) holds, $\theta = \theta_k$, and, therefore, (50) holds as well. If $\widehat{\Phi}(\theta_k) > 0$, since $\widehat{\Phi}$ is an affine function such that $\widehat{\Phi}(\theta_k) > 0$ and

$$\widehat{\Phi}(0) = \|h(y^k)\| - (1 - \alpha_\Phi)\|h(x^k)\| \leq (r - 1 + \alpha_\Phi)\|h(x^k)\| < 0,$$

then there exists exactly one value $t \in (0, \theta_k)$ such that $\widehat{\Phi}(t) = 0$. This value is the one given in (48); and $\widehat{\Phi}(\theta) = 0$ implies that (50) holds.

In the case $\theta = \theta_k$, $\theta \in [\bar{\theta}, \theta_k]$ obviously holds. When θ is computed by (48), $\theta < \theta_k$. Moreover, by (25) and (28), we have that

$$\frac{1}{\theta} \leq \frac{L(y^k, \lambda) - L(x^k, \lambda^k) + \|h(x^k)\|}{(1 - \alpha_\Phi)\|h(x^k)\| - r\|h(x^k)\|} \leq \frac{(\bar{\beta} + 1)\|h(x^k)\|}{(1 - \alpha_\Phi - r)\|h(x^k)\|} \leq \frac{\bar{\beta} + 1}{1 - \alpha_\Phi - r}$$

which implies $\theta \geq \bar{\theta}$. □

In the following lemma, we show that if the regularization parameter σ_j is large enough, then the sufficient descent conditions (46) and (49) both hold. In the sequel, bounds are set on the number of iterations and on the number of evaluations of the functions that define problem (1) and their derivatives.

Lemma 4.6 *Suppose that Assumption A1 holds. If $\sigma_j \geq \left(L_L + \frac{\|H\|}{2} + \tilde{\alpha}\right)$, where*

$$\tilde{\alpha} = \max\left\{\alpha_L, \frac{1 - \bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\right\} + 2\bar{\lambda}(\kappa_T + L_h) \tag{52}$$

and L_h, $\bar{\lambda}$, L_L satisfy (7), (14), (16), then the returned z^j, λ, and θ satisfy (46) and (49).

Proof: By (18), (44), and the facts that $\|H\| \leq M$ and $\sigma_j \geq 2(M + \tilde{\alpha} + L_L)$, we have that

$$\begin{aligned}
L(z^j, \lambda') &\leq L(y^k, \lambda') + \nabla L(y^k, \lambda')^T(z^j - y^k) + L_L\|z^j - y^k\|^2 \\
&\quad + \tfrac{1}{2}(z^j - y^k)^T H(z^j - y^k) - \tfrac{1}{2}(z^j - y^k)^T H(z^j - y^k) \\
&\leq \nabla L(y^k, \lambda')^T(z^j - y^k) + \tfrac{1}{2}(z^j - y^k)^T H(z^j - y^k) \\
&\quad + L(y^k, \lambda') + (M + \tilde{\alpha} + L_L)\|z^j - y^k\|^2 - \tilde{\alpha}\|z^j - y^k\|^2 \qquad (53) \\
&\leq \nabla L(y^k, \lambda')^T(z^j - y^k) + \tfrac{1}{2}(z^j - y^k)^T H(z^j - y^k) \\
&\quad + \tfrac{\sigma_j}{2}\|z^j - y^k\|^2 + L(y^k, \lambda') - \tilde{\alpha}\|z^j - y^k\|^2 \\
&\leq L(y^k, \lambda') - \tilde{\alpha}\|z^j - y^k\|^2.
\end{aligned}$$

Therefore, by (53), the definition of the Lagrangian, and (14), we have that

$$\begin{aligned} L(z^j,\lambda) &= L(y^k,\lambda) + [L(z^j,\lambda) - L(z^j,\lambda')] + [L(z^j,\lambda') - L(y^k,\lambda)] \\ &\leq L(y^k,\lambda) + (\lambda-\lambda')^T h(z^j) + [L(y^k,\lambda') - \tilde{\alpha}\|z^j - y^k\|^2 - L(y^k,\lambda)] \\ &\leq L(y^k,\lambda) + (\lambda-\lambda')^T (h(z^j) - h(y^k)) - \tilde{\alpha}\|z^j - y^k\|^2 \\ &\leq L(y^k,\lambda) + 2\bar{\lambda}\|h(z^j) - h(y^k)\| - \tilde{\alpha}\|z^j - y^k\|^2. \end{aligned}$$

Thus, by (12) and (45),

$$L(z^j,\lambda) \leq L(y^k,\lambda) - (\tilde{\alpha} - 2\bar{\lambda}(\kappa_T + L_h))\|z^j - y^k\|^2 \leq L(y^k,\lambda) - \max\left\{\alpha_L, \frac{1-\bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\right\}\|z^j - y^k\|^2,$$

i.e., (46) holds. In addition, we have that

$$L(z^j,\lambda) - L(y^k,\lambda) \leq -\frac{1-\bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\|z^j - y^k\|^2. \tag{54}$$

Let us now prove that (49) also holds. Note that, by (50),

$$\begin{aligned} \Phi(z^j,\lambda,\theta) - \Phi(x^k,\lambda^k,\theta) &= [\Phi(z^j,\lambda,\theta) - \Phi(y^k,\lambda,\theta)] + [\Phi(y^k,\lambda,\theta) - \Phi(x^k,\lambda^k,\theta)] \\ &\leq [\Phi(z^j,\lambda,\theta) - \Phi(y^k,\lambda,\theta)] - \alpha_\Phi\|h(x^k)\|^2. \end{aligned} \tag{55}$$

Define $v = \Phi(z^j,\lambda,\theta) - \Phi(y^k,\lambda,\theta)$. By the definition of Φ and (54), we have that

$$\begin{aligned} v &= \theta[L(z^j,\lambda) - L(y^k,\lambda)] + (1-\theta)\left[\|h(z^j)\| - \|h(y^k)\|\right] \\ &\leq \theta\left[-\tfrac{1-\bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\|z^j - y^k\|^2\right] + (1-\theta)\left[\|h(z^j)\| - \|h(y^k)\|\right]. \end{aligned} \tag{56}$$

By (13) and (45), we have that

$$\|h(z^j)\| - \|h(y^k)\| \leq \|J_h(y^k)(z^j - y^k)\| + L_h\|z^j - y^k\|^2 \leq (\kappa_T + L_h)\|z^j - y^k\|^2. \tag{57}$$

Combining (57), (56), and (51), we have that

$$\begin{aligned} v &\leq \theta\left[-\tfrac{1-\bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\|z^j - y^k\|^2\right] + (1-\theta)\left[(\kappa_T + L_h)\|z^j - y^k\|^2\right] \\ &\leq \bar{\theta}\left[-\tfrac{1-\bar{\theta}}{\bar{\theta}}(\kappa_T + L_h)\|z^j - y^k\|^2\right] + (1-\bar{\theta})(\kappa_T + L_h)\|z^j - y^k\|^2 = 0. \end{aligned}$$

Then, by the definition of v and (55), we have that (49) holds. □

Lemma 4.7 *Suppose that Assumption A1 holds. Then, for all $j \geq 0$,*

$$\sigma_j \leq \max\left\{2\tau_2\left(L_L + \frac{M}{2} + \tilde{\alpha}\right), \sigma_{\max}\right\}, \tag{58}$$

where $\tilde{\alpha}$ is as defined in (52)*. In addition, Algorithm 4.2 performs a single evaluation of ∇f and J_h and the number of evaluations of f and h is bounded by*

$$N_O^{f,h} = \log_{\tau_1}\left(\frac{2\left(L_c + M/2 + \tilde{\alpha}\right)}{\sigma_{\min}}\right) + 2 = O(1). \tag{59}$$

Proof: By Lemma 4.6, the fact that $\|H\| \leq M$ and the way of choosing σ_0 and σ_j for $j \geq 1$, we obtain (58). Moreover, the updating rule of σ_j also implies that the loop of Steps 2 to 6 iterates a maximum of $N_O^{f,h}$ times. In each iteration j, f and h are evaluated at z^j. By the definition of the algorithm, ∇f and J_h are evaluated only once, at y^k. □

4.3 The overall strategy

In this section, we put together the two modules defined above to present a practical proposal for Algorithm 3.1.

Algorithm 4.3. Let $x^0 \in \Omega$, $\lambda^0 \in \Lambda$, $\theta_0 \in (0,1)$, $r \in (0,1)$, $0 < r_{\text{feas}} \ll r$, $0 < \sigma_{\min} \leq \sigma_{\max}$, $M > 0$, $\beta_c \geq 0$, $\alpha_R > 0$, $\alpha_L > 0$, $\alpha_\Phi \in (0, 1-r)$, $\kappa_R > 0$, $\kappa_T > 0$, $\kappa_P > 0$, $\kappa_\varphi > 1$, and $1 < \tau_1 \leq \tau_2$ be given. Set $k \leftarrow 0$.

Step 1. Use Algorithm 4.1 with parameters x^k, r, r_{feas}, $\sigma_{\min}$, $\sigma_{\max}$, M, β_c, κ_R, α_R, κ_φ, τ_1, and τ_2 to compute y^k. If $\|h(y^k)\| \not\leq r\|h(x^k)\|$, then stop by declaring failure of the restoration phase and return y^k as an approximate stationary point of minimizing $\|c(x)\|$ subject to $x \in \Omega$.

Step 2. Use Algorithm 4.2 with parameters x^k, y^k, λ^k, r, α_Φ, $\sigma_{\min}$, $\sigma_{\max}$, M, κ_T, κ_P, α_L, τ_1, and τ_2 to compute x^{k+1}, λ^{k+1}, θ_{k+1}.

Step 3. Set $k \leftarrow k+1$ and go to Step 1.

The next result shows that Algorithm 4.3 is a special case of Algorithms 2.1 and 3.1.

Lemma 4.8 *Suppose that Assumption A1 holds. If Algorithm 4.3 does not stop at Step 1 by declaring failure of the restoration phase, then Algorithm 4.3 is a special case of Algorithms 2.1 and 3.1.*

Proof: If Algorithm 4.1 does not stop by declaring a failure of the restoration phase, then (25) holds by the definition of the algorithm. Furthermore, for all k, by Lemma 4.4, (31) holds with $\beta = \beta_c + N_R M \kappa_\varphi C_h$. This ensures that the hypotheses of Lemma 3.4 are satisfied and therefore (26) holds with $\beta_f = C_f \beta$. Since (25) and (31) hold for all k, the hypotheses of Lemma 3.1 are satisfied. Therefore, (28) holds for $\lambda = \lambda'$, with $\bar{\beta} = \beta_f + (r+1)\bar{\lambda}$. This gives the hypotheses of Lemma 4.5. Therefore, the sequence of θ_k is non-increasing and, for all k, θ_k is bounded from below by $\bar{\theta} = (1 - r - \alpha_\Phi)/(\bar{\beta} + 1)$, i.e. (3) holds. The conditions (4) and (27) hold because the values z^j, λ, and θ returned by Algorithm 4.2 satisfy (46) and (49) and $x^{x+1} = z^j$, $\lambda^{k+1} = \lambda$, and $\theta_{k+1} = \theta$. Therefore, Algorithm 4.3 is a special case of Algorithm 3.1. By Lemmas 4.4 and 3.3, it is also a special case of Algorithm 2.1. □

4.4 Global convergence to stationary points

Up to this point, we have established that the infeasibility of the iterates of Algorithm 4.3 is summable and that the squared norm of the difference of consecutive iterates is also summable. This means that limit points are feasible and that the step taken in each iteration converges to zero. In this section we show that the limit points of the sequence generated by Algorithm 4.3 satisfy an optimality condition. To do this, we need to recall the concept of L-AGP introduced in [4]. From now on, we assume that Ω is described as in the following assumption.

Assumption A2 *The non-empty compact convex polytope Ω is given by*

$$\Omega = \{x \in \mathbb{R}^n \mid Ax = b,\ Cx \leq d\},$$

where $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, $C \in \mathbb{R}^{q \times n}$, and $d \in \mathbb{R}^q$.

Under Assumption A2, we say that a feasible point $x^\star$ satisfies the L-AGP condition for problem (1) if there exists a sequence $\{x^k\} \subset \Omega$ such that $\lim_{k\to\infty} x^k = x^\star$ and $\lim_{k\to\infty} P_{S^k}(x^k - \nabla f(x^k)) - x^k = 0$, where

$$S^k = \{x \in \mathbb{R}^n \mid x \in \Omega,\ J_h(x^k)(x - x^k) = 0\}. \tag{60}$$

The L-AGP condition is a true optimality condition, i.e., it is satisfied by any minimizer, regardless of whether a constraint qualification is satisfied. A point $x^\star$ that satisfies the L-AGP condition is KKT if a weak constraint qualification holds at $x^\star$, see [5].

The following hypothesis establishes the optimality condition that must be satisfied by the approximate solution z^j of the optimization phase subproblem (43) to be returned by the Algorithm 4.2. Similar conditions have been used in [29] and [28].

Assumption A3 *For all k, $x^{k+1} \equiv z^j$, computed at Step 2 of Algorithm 4.3 by calling Algorithm 4.2, satisfies*

$$\left\| P_{S^k}\left(z^j - (\nabla L(y^k, \lambda') + (H + \sigma_j I)(z^j - y^k)\right) - z^j \right\| \leq \kappa_P \|z^j - y^k\|, \tag{61}$$

where S^k is as defined in (60).

The following lemma connects the step in the optimization phase with a measure of optimality.

Lemma 4.9 *Suppose that Assumptions A1, A2, and A3 hold. Then, for all k,*

$$\left\| P_{S^k}\left(y^k - \nabla L(y^k, \lambda')\right) - y^k \right\| \leq \kappa \|x^{k+1} - y^k\|, \tag{62}$$

where

$$\kappa = \kappa_P + M + \max\left\{2\tau_2\left(L_L + \frac{M}{2} + \tilde{\alpha}\right), \sigma_{\max}\right\} + 2 \tag{63}$$

and $\tilde{\alpha}$ is as defined in (52).

Proof: Let j be such that $x^{k+1} = z^j$. Define $u = \nabla L(y^k, \lambda') + (H + \sigma_j I)(x^{k+1} - y^k))$. By the non-expansiveness of projections, (61), and the fact that $\|H\| \leq M$, we have that

$$\begin{aligned}
&\|P_{S^k}(y^k - \nabla L(y^k, \lambda')) - y^k\| \\
&\quad \leq \|P_{S^k}(y^k - \nabla L(y^k, \lambda')) - P_{S^k}(x^{k+1} - u)\| + \|P_{S^k}(x^{k+1} - u) - x^{k+1}\| + \|(x^{k+1} - y^k))\| \\
&\quad \leq \|y^k - x^{k+1} + (H + \sigma_j I)(x^{k+1} - y^k)\| + (\kappa_P + 1)\|x^{k+1} - y^k\| \\
&\quad \leq (M + \sigma_j + \kappa_P + 2)\|x^{k+1} - y^k\|.
\end{aligned}$$

Therefore, (62) follows from (58). □

We are now ready to state an asymptotic optimality result for Algorithm 4.3 and, in the sequel, a complexity result.

Theorem 4.1 *Suppose that Assumptions A1, A2, and A3 hold. Assume that Algorithm 4.3 did not stop by declaring a failure of the restoration phase and, therefore, it generated sequences $\{x^k\}$ and $\{y^k\}$. If $x^\star$ is a limit point of $\{x^k\}$ or $\{y^k\}$, then x^* satisfies the L-AGP condition.*

Proof: By Lemmas 4.8 and 2.1, if $x^\star$ is an accumulation point of $\{x^k\}$, then $x^\star$ is feasible. Therefore, by (31), $\{x^k\}$ and $\{y^k\}$ have the same accumulation points. On the other hand, by [26, Corollary 3.1, p. 202],

$$\lim_{k\to\infty} \|P_{S^k}(y^k - \nabla f(y^k)) - y^k\| = \lim_{k\to\infty} \|P_{S^k}(y^k - \nabla L(y^k, \lambda')) - y^k\|,$$

while, by(30) and (62),

$$\lim_{k\to\infty} \|P_{S^k}(y^k - \nabla L(y^k, \lambda')) - y^k\| = 0.$$

Thus, every accumulation point of $\{x^k\}$ and $\{y^k\}$ satisfies the L-AGP condition, as we wanted to prove. □

Corollary 4.1 *Under the assumptions of Theorem 4.1, if $x^\star$ satisfies the L-AGP-regularity condition [5], then x^* is a KKT point.*

Theorem 4.2 *Suppose that Assumptions A1, A2, and A3 hold. Given $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$, if Algorithm 4.3 does not stop by declaring failure of the restoration phase, then it computes an iterate $y^k \in \Omega$ such that*

$$\|h(y^k)\| \leq \varepsilon_{\text{feas}}$$

and

$$\|P_{S^k}(y^k - \nabla f(y^k)) - y^k\| \leq \varepsilon_{\text{opt}}.$$

To perform this task, the iteration k and the number of evaluations of f, ∇f, h, and J_h are bounded above by N_{IR}, $N_{IR}\, N_O^{f,h}$, N_{IR}, $N_{IR}(N_R^h + N_O^{f,h}) + 1$, and $2N_{IR} + 1$, respectively, where

$$N_{IR} = \frac{\bar{h}}{\varepsilon_{\text{feas}}} + \frac{\kappa^2(2C_L + \bar{\beta}\bar{h})}{\alpha_L\, \varepsilon_{\text{opt}}^2} + 1 = O(\varepsilon_{\text{feas}}^{-1} + \varepsilon_{\text{opt}}^{-2}),$$

$N_O^{f,h} = O(1)$ is as defined in (59), $N_R^h = O(1)$ is as defined in (38), $\bar{h}$ satisfies (21), κ is defined in (63), C_L satisfies (15), and $\bar{\beta}$ is defined as in Lemma 3.2. In the case that Algorithm 4.3 stops declaring failure of the restoration phase, with the same bounds on the number of iterations and evaluations, it computes y^k satisfying

$$\|P_\Omega(y^k - \nabla c(y^k)) - y^k\| \leq \frac{r_{\text{feas}}}{r}\|h(y^k)\|.$$

Proof: By Corollary 3.1 we have that the maximum number of iterations until we get that $\|h(y^k)\| \leq \varepsilon_{\text{feas}}$ and $\|x^{k+1} - y^k\| \leq \varepsilon_{\text{opt}}/\kappa$ is N_{IR}, while by (62) we have that if $\|x^{k+1} - y^k\| \leq \varepsilon_{\text{opt}}/\kappa$ then $\|P_\Omega(y^k - (\nabla f(y^k) + J_h(y^k)^T\mu^k)) - y^k\| \leq \varepsilon_{\text{opt}}$. The remaining results follow from the definition of Algorithm 4.3 and Lemmas 4.3 and 4.7. □

Even in the case where Ω is an n-dimensional box, the presence of the tangent space with respect to h makes the projection onto S^k a non-trivial task. For this reason, it doesn't seem simple to test (61) in practice. A sequential optimality condition that is weaker than the L-AGP condition is AKKT [4]. We say that the AKKT condition holds in a feasible point $x^\star$ if there exist sequences $\{x^k\} \subset \mathbb{R}^n$, $\{\lambda^k\} \subset \mathbb{R}^m$, $\{\mu^k\} \subset \mathbb{R}^p$ and $\{\omega^k\} \subset \mathbb{R}^q_+$ such that $\lim_{k\to\infty} x^k = x^\star$, $\lim_{k\to\infty} \nabla f(x^k) + J_h(x^k)^T\lambda^k + A^T\mu^k + C^T\omega^k = 0$, and $\lim_{k\to\infty} \min\{d - Cx^k, \omega^k\} = 0$. Associated with this condition we have a stopping condition for the subproblem that is easier to test in practice, which is described in the following hypothesis.

Assumption A4 *For all k, there exist $\nu^k \in \mathbb{R}^m$, $\mu^k \in \mathbb{R}^p$, and $\omega^k \in \mathbb{R}^q_+$ such that $x^{k+1} \equiv z^j$, computed at Step 2 of Algorithm 4.3 by calling Algorithm 4.2, satisfies*

$$\|\nabla f(y^k) + J_h(y^k)^T\nu^k + (H + \sigma_j I)(z^j - y^k) + A^T\mu^k + C^T\omega^k\| \leq \kappa_P\|z^j - y^k\| \tag{64}$$

and

$$\|\min\{d - Cz^j, \omega^k\}\| \leq \kappa_P\|z^j - y^k\|. \tag{65}$$

The following lemma connects the step in the optimization phase with a measure of optimality.

Lemma 4.10 *Suppose that Assumptions A1, A2, and A4 hold. Then, for all k,*

$$\|\nabla f(y^k) + J_h(y^k)^T\nu^k + A^T\mu^k + C^T\omega^k\| \leq \hat{\kappa}\|x^{k+1} - y^k\| \tag{66}$$

and

$$\|\min\{d - Cy^k, \omega^k\}\| \leq \hat{\kappa}\|x^{k+1} - y^k\|, \tag{67}$$

where

$$\hat{\kappa} = \max\left\{\kappa_P + M + \max\left\{2\tau_2\left(L_L + \frac{M}{2} + \tilde{\alpha}\right), \sigma_{\max}\right\}, \sqrt{n}(\kappa_P + \|C\|)\right\} \tag{68}$$

and $\tilde{\alpha}$ is defined in (52)*.*

Proof: Let j be such that $x^{k+1} = z^j$. Then, by (64), the fact that $\|H\| \leq M$, and (58), we have that

$$\begin{aligned}
&\|\nabla f(y^k) + J_h(y^k)^T\nu^k + A^T\mu^k + C^T\omega^k\| \\
&\leq \|\nabla f(y^k) + J_h(y^k)^T\nu^k + A^T\mu^k + C^T\omega^k + (H + \sigma_j I)(z^j - y^k)\| + \|(H + \sigma_j I)(z^j - y^k)\| \\
&\leq \kappa_P\|x^{k+1} - y^k\| + \left(M + \max\left\{2\tau_2\left(L_L + \frac{M}{2} + \tilde{\alpha}\right), \sigma_{\max}\right\}\right)\|x^{k+1} - y^k\|.
\end{aligned}$$

Thus, we have (66). Note now that, since $y^k \in \Omega$ and $\omega^k \in \mathbb{R}^q_+$, then $0 \leq \min\{d - Cy^k, \omega^k\}$. Therefore, by (65), for all $i = 1, \ldots, q$, we have that

$$[\min\{d - Cy^k, \omega^k\}]_i \leq [\min\{d - Cx^{k+1}, \omega^k\}]_i + |[C(x^{k+1} - y^k)]_i| \leq \kappa_P\|x^{k+1} - y^k\| + \|C\|\|x^{k+1} - y^k\|.$$

Thus, $\|\min\{d - Cy^k, \omega^k\}\| \leq \sqrt{n}(\kappa_P + \|C\|)\|x^{k+1} - y^k\|$, from which (67) follows. □

Theorem 4.3 *Suppose that Assumptions A1, A2, and A4 hold. Assume that Algorithm 4.3 did not stop by declaring a failure of the restoration phase and, therefore, it generated sequences $\{x^k\}$ and $\{y^k\}$. If $x^\star$ is a limit point of $\{x^k\}$ or $\{y^k\}$, then $x^\star$ satisfies the AKKT condition. Moreover, if $x^\star$ satisfies the AKKT-regularity condition [6], then x^* is a KKT point. In addition, given $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$, the algorithm computes an iterate $y^k \in \Omega$ such that $\|h(y^k)\| \leq \varepsilon_{\text{feas}}$ and $\|\nabla f(y^k) + J_h(y^k)^T\nu^k + A^T\mu^k + C^T\omega^k\| \leq \varepsilon_{\text{opt}}$, where ν^k, μ^k and ω^k are defined as in Assumption A4. To perform this task, all complexity bounds are as in Theorem 4.2, substituting κ from Lemma 4.9 with $\hat{\kappa}$ from Lemma 4.10.*

Proof: The proof is analogous to the proofs of Theorem 4.1 and 4.2. □

4.5 Acceleration procedure

We have seen so far that Algorithm 3.1 is a special case of Algorithm 2.1 and that, as done in Algorithm 4.3, we can use Algorithms 4.1 and 4.2 to compute y^k, x^{k+1}, λ^{k+1}, and θ_{k+1} at each two-phase iteration k of Algorithm 3.1. However, Algorithm 2.1 is more general and allows alternative procedures to these. One alternative would be to try an acceleration strategy which, if successful, would return x^{k+1}, λ^{k+1}, and θ_{k+1}. Otherwise, we could rely on the algorithm for which we have theoretical guarantees that the iteration can be completed successfully. The following algorithm describes an acceleration strategy based on the approximate solution of a quadratic programming subproblem.

Algorithm 4.4. Let $x^k \in \Omega$, $\lambda^k \in \Lambda$, $\theta_k \in (0, 1)$ $0 \leq \sigma_{\max}$, $M > 0$, and $\kappa_P > 0$ be given.

Step 1. Choose $\lambda' \in \Lambda$, $\sigma \in [0, \sigma_{\max}]$, and a symmetric matrix $H \in \mathbb{R}^{n\times n}$ such that $\|H\| \leq M$.

Step 2. By approximately solving the problem

$$\underset{z\in\Omega}{\text{Minimize}}\, \frac{1}{2}(z-x^k)^T H(z-x^k) + \nabla L(x^k, \lambda')^T(z-x^k) + \frac{\sigma}{2}\|z-x^k\|^2 \text{ subject to } J_h(x^k)(z-x^k) = -h(x^k),$$

try to compute $z \in \Omega$, $\nu \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$, and $\omega \in \mathbb{R}^q_+$ satisfying

$$\|\nabla f(x^k) + J_h(x^k)^T\nu + (H + \sigma I)(z - x^k) + A^T\mu + C^T\omega\| \leq \kappa_P\|z - x^k\| \tag{69}$$

and

$$\|\min\{d - Cz, \omega\}\| \leq \kappa_P\|z - x^k\|. \tag{70}$$

The description of the accelerated algorithm follows.

Algorithm 4.5. The description of this algorithm is almost identical to the description of Algorithm 4.3. The difference is that there is a Step 0 in which it tries, by using Algorithm 4.4 a finite number of times limited by K, to compute $z \in \Omega$, $\nu \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$, and $\omega \in \mathbb{R}^q_+$ that satisfy (69) and (70) for some $\lambda' \in \Lambda$, $\sigma \in [0, \sigma_{\max}]$, and $H \in \mathbb{R}^{n\times n}$ such that $\|H\| \leq M$. In case of success, if (4) and (5) hold with $x^{k+1} = z$, $\lambda^{k+1} = P_\Lambda(\nu)$, and $\theta_{k+1} = \theta_k$, then it defines $x^{k+1} = z$, $\lambda^{k+1} = P_\Lambda(\nu)$, $\theta_{k+1} = \theta_k$, $\nu^k = \nu$, $\mu^k = \mu$, and $\omega^k = \omega$, and goes to Step 3[1]. (In Step 3, the command "go to Step 1" must be replaced with "go to Step 0").

Theorem 4.4 *Suppose that Assumptions A1 and A2 hold and that, every time x^{k+1} is computed by Algorithm 4.2, Assumption A4 holds. Given $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$, if Algorithm 4.5 does not stop by declaring failure of the restoration phase, then it computes an iterate such that either*

$$\begin{cases} x^k \in \Omega \\ \|h(x^k)\| \leq \varepsilon_{\text{feas}} \\ \|\nabla f(x^k) + J_h(x^k)^T\nu^k + A^T\mu^k + C^T\omega^k\| \leq \varepsilon_{\text{opt}} \\ \|\min\{d - Cx^k, \omega^k\}\| \leq \varepsilon_{\text{opt}} \end{cases} \tag{71}$$

or

$$\begin{cases} y^k \in \Omega \\ \|h(y^k)\| \leq \varepsilon_{\text{feas}} \\ \|\nabla f(y^k) + J_h(y^k)^T\nu^k + A^T\mu^k + C^T\omega^k\| \leq \varepsilon_{\text{opt}} \\ \|\min\{d - Cy^k, \omega^k\}\| \leq \varepsilon_{\text{opt}}. \end{cases} \tag{72}$$

To perform this task, the iteration k and the number of evaluations of f, ∇f, h, and J_h are bounded above by N_A, $N_A(N_O^{f,h} + K)$, $2N_A$, $N_A(N_R^h + N_O^{f,h} + K) + 1$, and $2N_A + 2$, respectively, where

$$N_A = \frac{\bar{h}}{\min\{\varepsilon_{\text{feas}}, \varepsilon_{\text{opt}}/(2\hat{\kappa}\beta)\}} + \frac{4\hat{\kappa}^2(2C_L + \tilde{\alpha}\bar{h})}{\alpha_L\varepsilon_{\text{opt}}^2} + 1 = O(\varepsilon_{\text{feas}}^{-1} + \varepsilon_{\text{opt}}^{-2}),$$

$N_O^{f,h} = O(1)$ is as defined in (59), $N_R^h = O(1)$ is as defined in (38), $\bar{h}$ satisfies (21), $\hat{\kappa}$ is defined in (68), C_L satisfies (15), and $\tilde{\alpha} = C_f\beta + (r+1)\bar{\lambda} + 3\alpha_L D_\Omega\beta$, C_f as in (8), D_Ω as in (10), $\bar{\lambda}$ as in (14) and β as in (41). Moreover, if $x^\star$ is a limit point of $\{x^k\}$ or $\{y^k\}$, then $x^\star$ is AKKT. Consequently, if $x^\star$ satisfies the AKKT-regularity condition, then x^* is a KKT point.

Proof: Note that, regardless of whether x^{k+1}, λ^{k+1} and θ_{k+1} calculated in iteration k of Algorithm 4.5 were obtained by the acceleration strategy (call to Algorithm 4.4 in Step 0) or by a two-phase iteration (call to Algorithms 4.1 and 4.2 in Steps 1 and 2, respectively), the conditions (3), (4), and (5) hold. Therefore, by Lemmas 2.1 and 2.2, we have that there exists an iterate x^{k+1} with $k \leq N_A$ such that

[1]The definition of ν^k, μ^k, and ω^k serves only to test the stopping criterion and for the theoretical analysis.

$\|x^{k+1} - x^k\| \leq \varepsilon_{\text{opt}}/(2\hat{\kappa})$ and $\|h(x^k)\| \leq \varepsilon_{\text{feas}}$ or $\|h(y^k)\| \leq \|h(x^k)\| \leq \min\{\varepsilon_{\text{feas}}, \varepsilon_{\text{opt}}/(2\hat{\kappa}\beta)\} \leq \varepsilon_{\text{feas}}$. If this x^{k+1} is a result of the acceleration strategy, then analogously to the proof of Theorem 4.3 (replacing y^k by x^k), we get (71). On the other hand, if x^{k+1} is computed in a two-phase iteration, we have that

$$\|x^{k+1} - y^k\| \leq \|x^{k+1} - x^k\| + \|x^k - y^k\| \leq \|x^{k+1} - x^k\| + \beta\|h(x^k)\| \leq \varepsilon_{\text{opt}}/\hat{\kappa}.$$

Thus, by Lemma 4.10, we obtain (72). The bounds on the number of evaluations follow trivially from the definition of the algorithm. Since $\varepsilon_{\text{feas}}$ and ε_{opt} are arbitrary, we have that $x^\star$ is AKKT and therefore, if the AKKT-regularity condition holds, it is KKT. □

5 Numerical experiments

In this section we present numerical results with Algorithm 4.5 for the case where $\Omega = \{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$ with $\ell, u \in \mathbb{R}^n$, $\ell_i < u_i$ for $i = 1, \ldots, n$. On the one hand, we intend to show that the algorithm is implementable and can be used as a general-purpose tool for solving nonlinear programming problems with equality constraints and bounds. On the other hand, we wish to evaluate whether the optional SQP acceleration (Step 0) contributes to improve the performance of the method. In the experiments, we considered all 376 nonlinear programming problems from the CUTEst collection [40] (version 2.2.0) with only equality constraints and bounds. We considered the standard dimensions of each problem (number of variables n and number of constraints m) and used the provided initial estimates x^0 and λ^0.

We implemented Algorithm 4.5, which calls Algorithms 4.1, 4.2, and 4.4, and has Algorithm 4.3 as a particular, in Fortran 90. In the numerical experiments we considered $\Lambda = \{\lambda \in \mathbb{R}^m \mid \|\lambda\|_\infty \leq 10^{16}\}$, $\theta_0 = 0.9$, $r = 0.3$, $r_{\text{feas}} = 10^{-12}$, $\sigma_{\min} = 10^{-2}$, $\sigma_{\max} = 10^{16}$, $M = 10^{16}$, $\beta_c = 1$, $\alpha_R = \alpha_L = \alpha_\Phi = 10^{-8}$, $\kappa_R = \kappa_T = \kappa_P = 10^{-4}$, $\kappa_\varphi = 10^{16}$, and $\tau_1 = \tau_2 = 10$. All these values are rather standard in the literature of inexact restoration methods [15] and regularized methods [16] and Algorithm 4.5 is not very sensitive to small variations of these values.

In Algorithm 4.1, we choose $z^0 = x^k$ in Step 2 (and hence the parameter β_c plays no role) and $B_\ell = J(z^\ell)^T J(z^\ell)$ and $\sigma_{\ell,0} = 10^{-8} \approx \sqrt{\epsilon_{\text{mach}}}$ in Step 4, where $\epsilon_{\text{mach}} \approx 10^{-16}$ represents the machine precision. Thus, by the way of updating the regularization parameter, $\lambda_{\min}(B_\ell + \sigma_{\ell j} I) \geq 1/M$ is satisfied, for all ℓ and j, for any $M \geq 10^8$. The subproblem (32), that consists in minimizing a quadratic function subject to bound constraints, is solved by using Gencan [8, 17, 20, 21, 23]. In Algorithm 4.2, we choose $\sigma_0 = 0$, $\lambda' = \lambda^k$, and $H = \nabla^2 L(y^k, \lambda^k)$ in Step 1. The subproblem in Step 2, that consist in minimizing a quadratic function subject to linear equality constraints and bound constraints, is solved using Algencan [1, 2, 23, 24]. The computed approximate solution z^j satisfies (44), (45), and the optimality conditions of Assumption A4, i.e., (64) and (65). This means that, together with z^j, we get ν^j and ω^j such that

$$\|\nabla f(y^k) + J_h(y^k)^T \nu^j + (H + \sigma_j I)(z^j - y^k) + [-I \mid I]\,\omega^j\| \leq \kappa_P \|z^j - y^k\|$$

and

$$\|\min\{-[-I \mid I]^T z^j, \omega^j\}\| \leq \kappa_P \|z^j - y^k\|.$$

All this to say that, in Step 3, we choose $\lambda = P_\Lambda(\nu^j)$. In Algorithm 4.4, we only considered the cases $K = 0$ and $K = 1$. This means that Algorithm 4.4 is called by Algorithm 4.5 at most once per iteration. When called at iteration k, we choose $\lambda' = \lambda^k$, $\sigma = 0$, and $H = \nabla^2 L(x^k, \lambda^k)$ in Step 1. The subproblem in Step 2, that consist in minimizing a quadratic function subject to linear equality constraints and bound constraints, is solved using Algencan.

Given tolerances $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$ ($\varepsilon_{\text{feas}} = \varepsilon_{\text{opt}} = 10^{-8}$ in the numerical experiments), the stopping criterion

$$\|h(x)\| \leq \varepsilon_{\text{feas}} \tag{73}$$

and

$$\|P_\Omega(x - \nabla L(x, \lambda)) - x\| \leq \varepsilon_{\text{opt}}, \tag{74}$$

is tested before starting the algorithm with $x \equiv x^0$ and $\lambda \equiv \lambda^0$ and at the end of Steps 0 and 2 with $x \equiv x^{k+1}$ and $\lambda \equiv \lambda^{k+1}$. Note that the criterion chosen concerns the point x^{k+1} and not the points x^k or y^k, as would be suggested by Theorem 4.4. This is because it is natural that the point reached by the acceleration or optimization process is better than the point at which the iteration began. Therefore, it is reasonable to assume that we have obtained the desired accuracy in such an iteration, with the associated Lagrange multiplier.

Numerical experiments were conducted on a computer with an Apple M1 Max processor and a 64GB RAM memory, running Ventura 13.4.1. Codes were compiled by the GNU Fortran compiler of GCC (version 14.1.0) with the -O3 optimization directive enabled. We set a limit of 10 minutes of CPU time for each method/problem.

Hereafter, we will call IR to Algorithm 4.3 (or Algorithm 4.5 with $K = 0$) and accelerated IR to Algorithm 4.5 with $K = 1$. Detailed tables with the results of each method can be found in `http://www.ime.usp.br/~egbirgin/`. Regarding the stopping criterion, IR stopped satisfying (73,74) in 239 problems, while accelerated IR stopped satisfying (73,74) in 270 problems. In the remaining problems the methods stopped by hitting the CPU time limit or by an alternative criterion associated with lack of progress. Regardless of the stopping criterion, IR found a point satisfying (73) in 283 problems, while accelerated IR did the same in 296 problems. So far we can say that accelerated IR wins from IR if we use "finding KKT points" as a comparison criterion, since the former found nearly 10% more KKT points than the latter. Accelerated IR is still better if the comparison criterion is to find feasible points, but the advantage in that case drops to nearly 5%.

There are 277 problems for which both methods satisfied (73). Let's call f_1 and f_2 the function values found by the two methods in a given problem and say that f_1 and f_2 are equivalent if

$$f_i \leq f_{\min} + f_{\text{tol}} \max\{1, |f_{\min}|\} \text{ for } i = 1, 2,$$

where $f_{\min} = \min\{f_1, f_2\}$ and $f_{\text{tol}} = 0.1$. Among the 277 problems in which both methods found feasible points, the function values found by the methods are equivalent in 273. (This result remains virtually unchanged for different values of $f_{\text{tol}} \in \{10^{-2}, 10^{-3}, \ldots, 10^{-8}\}$.) This shows that the methods are tied if the criterion is to find a feasible point with the smallest possible objective function value, unless by the fact of accelerated IR having found feasible points in cases where the IR method did not, as described above.

It remains now to compare the efficiency of the methods in those $n_{\text{prob}} = 273$ problems in which they both found feasible points with equivalent objective function values. For this comparison we use performance profiles [30]. We use CPU time as a performance measure. Figure 1 shows on the abscissa $\kappa \geq 1$ and on the ordinate, for each method $i \in \{\text{IR}, \text{accelerated IR}\}$,

$$\Gamma_i(\kappa) = \frac{|\{j \in \{1, \ldots, n_{\text{prob}}\} \,|\, t_{ij} \leq \kappa \min_{\ell \in M}\{t_{\ell j}\}\}|}{n_{\text{prob}}},$$

where t_{ij} represents the CPU time used by method i in problem j. The figure shows that in 75% of the problems the two methods behave very similarly. In about half of the problems, the acceleration speeds up the satisfaction of the stopping criterion and, in the other half, either the acceleration ends up not speeding up the method or unsuccessful attempts to accelerate slow down the convergence of

the method slightly. But in all these cases, the slower method never takes more than twice as long as the faster method. On the other hand, in the remaining 25% of the problems, the acceleration speeds up the convergence of the non-accelerated method, which takes between 2 and 5,000 times longer than the accelerated method.

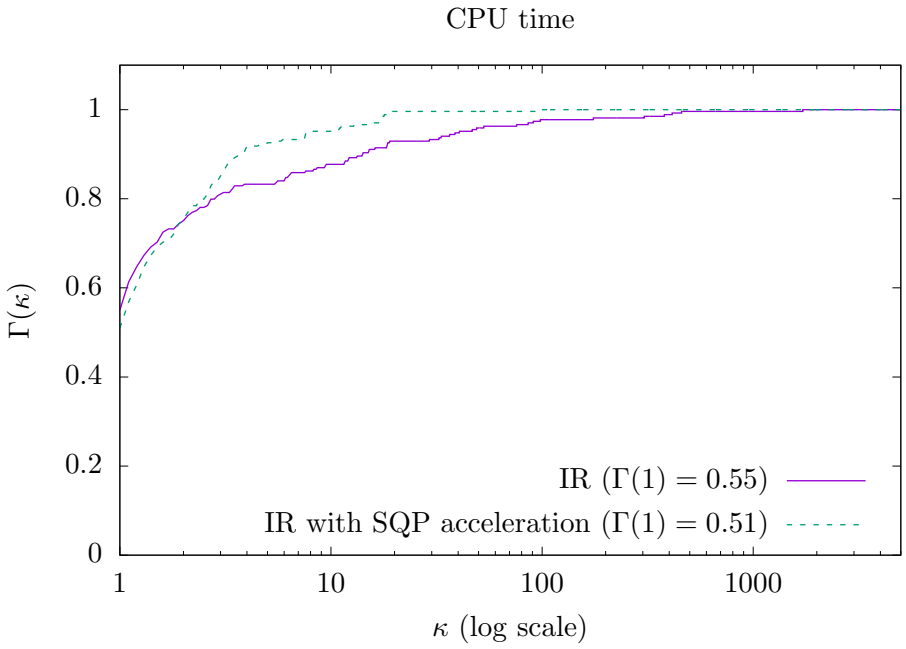

Figure 1: Performance profile using CPU time as the performance metric to compare the efficiency of IR and accelerated IR on the 273 (out of 376) problems where both methods found a feasible point with equivalent objective function value.

In the last decades several attempts were made to implement a general purpose inexact restoration method, see [16]. If on the one hand it is true that there is no efficient implementation that competes with efficient implementations of other paradigms such as interior points and augmented Lagrangian, there are efficient implementations for specific problems in which efficient and ad-hoc algorithms exist for the restoration phase. Regarding the implementation described in the present work, we can state that the amount of KKT points (270 out of 364) and feasible points (296 out of the same 364) found by accelerated IR are compatible with the proportions found by consolidated methods such as Algencan [2] and Ipopt [52], see [24]. A more robust and efficient implementation of an IR method should be based on the use of more specific algorithms for the resolution of each of the subproblems of Algorithms 4.1, 4.2, and 4.4.

6 Final remarks

In this article we have identified the main elements for globalizing a basic nonlinear programming framework using the sharp Lagrangian as a merit function. This analysis shows that the complexity with respect to the number of iterations of the algorithm is independent of the way the iterates are obtained, as long as the established descent conditions are satisfied. This analysis has been extended to a general Inexact Restoration algorithm, which turns out to be a special case of the basic algorithm. Using a quadratic regularization strategy, it was shown that the computational effort for evaluating

functions and their derivatives at a single iteration depends only on the algorithmic parameters and constants inherent to the problem, but not on the precision required as a stopping criterion. As a result, it was shown that the complexity in terms of iterations and number of evaluations is $O(\varepsilon_{\text{feas}}^{-1} + \varepsilon_{\text{opt}}^{-2})$ to obtain an $\varepsilon_{\text{feas}}$-feasible point that satisfies an optimality measure with tolerance ε_{opt}. Furthermore, the introduced framework could also be applied to alternatives using line search and trust regions. In this sense, we hope that the results of this study will facilitate the analysis of several other similar algorithms.

Another aspect to note is that the presented Inexact Restoration method allows greater flexibility in terms of parameters, inaccuracy in solving subproblems, and choices for obtaining iterates, compared to previous variants in the literature. In particular, we highlight the possibility to update the Lagrange multiplier together with the generation of x^{k+1} and not with y^k as in previous work. This is important because it is usually in the process of computing x^{k+1} that optimality information becomes available.

Finally, we would like to emphasize that the general framework makes it possible to use acceleration techniques while maintaining a well-established theory. In this paper, we have shown an alternative way to globalize the SQP method using a two-phase technique that also exploits the use of quadratic subproblems. With this strategy, it was possible to speed up convergence in some of the problems used in our numerical tests. We believe that this general framework can efficiently incorporate speedups developed for specific problems, many of which lack rigorous convergence theory. Exploring these possibilities in topological optimization and data science problems is of particular interest to us for future research.

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. Augmented Lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111(1–2):5–32, 2006.

[2] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18:1286–1309, 2007.

[3] R. Andreani, S. L. C. Castro, J. L. Chela, A. Friedlander, and S. A. Santos. An inexact-restoration method for nonlinear bilevel programming problems. *Computational Optimization and Applications*, 43(3):307–328, 2007.

[4] R. Andreani, G. Haeser, and J. M. Martínez. On sequential optimality conditions for smooth constrained optimization. *Optimization*, 60(6):627–641, 2011.

[5] R. Andreani, J. M. Marínez, A. Ramos, and P. J. S. Silva. Strict constraint qualifications and sequential optimality conditions for constrained optimization. *Mathematics of Operations Research*, 3(43):693–717, 2018.

[6] R. Andreani, J. M. Martínez, A. Ramos, and P. J. S. Silva. A cone-continuity constraint qualification and algorithmic consequences. *SIAM Journal on Optimization*, 26(1):96–110, 2016.

[7] R. Andreani, V. A. Ramirez, S. A. Santos, and L. D. Secchin. Bilevel optimization with a multiobjective problem in the lower level. *Numerical Algorithms*, 81(3):915–946, 2019.

[8] M. Andretta, E. G. Birgin, and J. M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization. *Optimization*, 54(3):305–325, 2005.

[9] M. B. Arouxét, N. E. Echebest, and E. A. Pilotta. Inexact restoration method for nonlinear optimization without derivatives. *Journal of Computational and Applied Mathematics*, 290:26–43, 2015.

[10] N. Banihashemi and C. Y. Kaya. Inexact restoration for euler discretization of box-constrained optimal control problems. *Journal of Optimization Theory and Applications*, 156(3):726–760, 2012.

[11] N. Banihashemi and C. Y. Kaya. Inexact restoration and adaptive mesh refinement for optimal control. *Journal of Industrial & Management Optimization*, 10(2):521–542, 2014.

[12] S. Bellavia, N. Krejić, and B. Morini. Inexact restoration with subsampled trust-region methods for finite-sum minimization. *Computational Optimization and Applications*, 76(3):701–736, 2020.

[13] S. Bellavia, N. Krejić, B. Morini, and S. Rebegoldi. A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. *Computational Optimization and Applications*, 84(1):53–84, 2022.

[14] R. H. Bielschowsky and F. A. M. Gomes. Dynamic control of infeasibility in equality constrained optimization. *SIAM Journal on Optimization*, 19(3):1299–1325, 2008.

[15] 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*, 282:1–16, 2015.

[16] 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(4):815–838, 2019.

[17] E. G. Birgin and J. M. Gentil. Evaluating bound-constrained minimization software. *Computational Optimization and Applications*, 53(2):347–373, 2012.

[18] 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(311):1307–1326, 2017.

[19] E. G. Birgin, N. Krejić, and J. M. Martínez. Iteration and evaluation complexity for the minimization of functions whose computation is intrinsically inexact. *Mathematics of Computation*, 89(321):253–278, 2019.

[20] E. G. Birgin and J. M. Martínez. A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients. In G. Alefeld and X. Chen, editors, *Topics in Numerical Analysis*, Computing Supplementa (Computing, vol. 15), pages 49–60. Springer Vienna, 2001.

[21] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002.

[22] E. G. Birgin and J. M. Martínez. Local convergence of an inexact-restoration method and numerical experiments. *Journal of Optimization Theory and Applications*, 127(2):229–247, 2005.

[23] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2014.

[24] E. G. Birgin and J. M. Martínez. Complexity and performance of an augmented Lagrangian algorithm. *Optimization Methods and Software*, 35:885–920, 2020.

[25] 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(2):1189–1213, 2013.

[26] L. F. Bueno, G. Haeser, and J. M. Martínez. A flexible inexact-restoration method for constrained optimization. *Journal of Optimization Theory and Applications*, 165(1):188–208, 2014.

[27] L. F. Bueno, G. Haeser, and J. M. Martínez. An inexact restoration approach to optimization problems with multiobjective constraints under weighted-sum scalarization. *Optimization Letters*, 10(6):1315–1325, 2016.

[28] L. F. Bueno, F. Larreal, and J. M. Martínez. Inexact restoration for minimization with inexact evaluation both of the objective function and the constraints. *Mathematics of Computation*, 93(345):293–326, 2024.

[29] L. F. Bueno and J. M. Martínez. On the complexity of an inexact restoration method for constrained optimization. *SIAM Journal on Optimization*, 30(1):80–101, 2020.

[30] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[31] N. Echebest, M. L. Schuverdt, and R. P. Vignau. An inexact restoration derivative-free filter method for nonlinear programming. *Computational and Applied Mathematics*, 36(1):693–718, 2015.

[32] A. Fischer and A. Friedlander. A new line search inexact restoration approach for nonlinear programming. *Computational Optimization and Applications*, 46(2):333–346, 2009.

[33] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter. Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.

[34] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[35] J. B. Francisco, D. S. Gonçalves, F. S. V. Bazán, and L. L. T. Paredes. Non-monotone inexact restoration method for nonlinear programming. *Computational Optimization and Applications*, 76(3):867–888, 2019.

[36] J. B. Francisco, D. S. Gonçalves, F. S. V. Bazán, and L. L. T. Paredes. Nonmonotone inexact restoration approach for minimization with orthogonality constraints. *Numerical Algorithms*, 86(4):1651–1684, 2020.

[37] J. B. Francisco, J. M. Martínez, L. Martínez, and F. Pisnitchenko. Inexact restoration method for minimization problems arising in electronic structure calculations. *Computational Optimization and Applications*, 50(3):555–590, 2010.

[38] F. A. M. Gomes and T. A. Senne. An SLP algorithm and its application to topology optimization. *Computational & Applied Mathematics*, 30(1):53–89, 2011.

[39] M. A. Gomes-Ruggiero, J. M. Martínez, and S. A. Santos. Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints. *SIAM Journal on Scientific Computing*, 31(3):1628–1652, 2009.

[40] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2014.

[41] E. W. Karas, E. A. Pilotta, and A. A. Ribeiro. Numerical comparison of merit function with filter criterion in inexact restoration algorithms using hard-spheres problems. *Computational Optimization and Applications*, 44(3):427–441, 2008.

[42] C. Y. Kaya and J. M. Martínez. Euler discretization and inexact restoration for optimal control. *Journal of Optimization Theory and Applications*, 134(2):191–206, 2007.

[43] N. Krejić and J. M. Martínez. Inexact restoration approach for minimization with inexact evaluation of the objective function. *Mathematics of Computation*, 85(300):1775–1791, 2015.

[44] J. M. Martínez. Inexact-restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming. *Journal of Optimization Theory and Applications*, 111(1):39–58, 2001.

[45] J. M. Martínez and E. A. Pilotta. Inexact-restoration algorithm for constrained optimization. *Journal of Optimization Theory and Applications*, 104(1):135–163, 2000.

[46] A. Miele, H. Y. Huang, and J. C. Heideman. Sequential gradient-restoration algorithm for the minimization of constrained functions – ordinary and conjugate gradient versions. *Journal of Optimization Theory and Applications*, 4:213–243, 1969.

[47] M. Rom and M. Avriel. Properties of the sequential gradient-restoration algorithm (SGRA), part 1: Introduction and comparison with related methods. *Journal of Optimization Theory and Applications*, 62(1):77–98, 1989.

[48] M. Rom and M. Avriel. Properties of the sequential gradient-restoration algorithm (SGRA), part 2: Convergence analysis. *Journal of Optimization Theory and Applications*, 62(1):99–125, 1989.

[49] J. B. Rosen. The gradient projection method for nonlinear programming. Part I. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[50] J. B. Rosen. The gradient projection method for nonlinear programming. Part II. Nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

[51] C. E. P. Silva and M. T. T. Monteiro. A filter inexact-restoration method for nonlinear programming. *TOP*, 16(1):126–146, 2008.

[52] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2005.

[53] J. Walpen, P. A. Lotito, E. M. Mancinelli, and L. Parente. The demand adjustment problem via inexact restoration method. *Computational and Applied Mathematics*, 39(3):204, 2020.

[54] P. Wolfe. Methods of nonlinear programming. In R. L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 67–86. McGraw-Hill, New York, 1963.