SIAM J. OPTIM.
Vol. 23, No. 2, pp. 1189–1213

INEXACT RESTORATION METHOD FOR DERIVATIVE-FREE OPTIMIZATION WITH SMOOTH CONSTRAINTS*

L. F. BUENO†, A. FRIEDLANDER‡, J. M. MARTÍNEZ‡, AND F. N. C. SOBRAL§

Abstract. A new method is introduced for solving constrained optimization problems in which the derivatives of the constraints are available but the derivatives of the objective function are not. The method is based on the inexact restoration framework, by means of which each iteration is divided in two phases. In the first phase one considers only the constraints, in order to improve feasibility. In the second phase one minimizes a suitable objective function subject to a linear approximation of the constraints. The second phase must be solved using derivative-free methods. An algorithm introduced recently by Kolda, Lewis, and Torczon for linearly constrained derivative-free optimization is employed for this purpose. Under usual assumptions, convergence to stationary points is proved. A computer implementation is described and numerical experiments are presented.

Key words. inexact restoration, derivative-free optimization, global convergence, numerical experiments

AMS subject classifications. 65K05, 65Y20, 90C56e

DOI. 10.1137/110856253

1. Introduction.

In this paper we address nonlinear programming problems in which the derivatives of the objective function are not available, whereas the derivatives of the constraints are [37]. Several methods take advantage of the simplicity of boxes and linear constraints [16, 27, 32, 33, 49] or need sufficient thickness of the feasible set [10, 11]. Here, we have in mind more general, perhaps highly nonlinear, constraints. We believe that approaches in which one evaluates function and constraints at the same points (for example, the augmented Lagrangian approaches of [17, 25, 34, 35]) are not fully satisfactory because sometimes the presence of topologically complex constraints causes the necessity of performing many evaluations. In these cases, the intrinsically expensive objective function could be unnecessarily computed, increasing the overall computational cost.

In this context, methods that separately evaluate constraints and objective function seem to be useful. The key point is that one should not evaluate the objective function when the partial (possibly difficult) goal is to improve feasibility. Inexact restoration (IR) methods are well suited for this purpose.

Restoration ideas have a long tradition in constrained optimization. Rosen [53, 54] introduced gradient projection methods for linear and nonlinear constraints, Miele and his coworkers [43, 44, 45] developed sequential gradient restoration ideas, whose rigorous convergence theory was proved by Rom and Avriel [51, 52]. Generalized

*Received by the editors November 22, 2011; accepted for publication (in revised form) January 29, 2013; published electronically June 20, 2013. This work was supported by PRONEX-CNPq/FAPERJ grant E-26/171.164/2003–APQ1, CEPID–Industrial Mathematics (FAPESP 2011-51305-0), FAPESP grants 03/09169-6, 06/53768-0, 07/06663-0, and 08/00468-4, and CNPq.
http://www.siam.org/journals/siopt/23-2/85625.html

†Department of Science and Technology, Federal University of São Paulo, São José dos Campos, São Paulo, Brazil (lfelipebueno@gmail.com).

‡Department of Applied Mathematics, Institute of Mathematics, Statistics and Scientific Computing, State University of Campinas, Campinas, São Paulo, Brazil (friedlan@ime.unicamp.br, martinez@ime.unicamp.br).

§Research and Development Division, Itaú-Unibanco, Praça Alfredo Egydio de Souza Aranha, 100, São Paulo, Brazil (chicosobral@gmail.com).

reduced gradient methods [1, 30] are still frequently used in engineering applications. In all these cases restoration is performed along specifically given directions (gradient-related in [53, 54, 43, 44, 45] and coordinate directions in [1, 30]). Methods with free restoration procedures have been recently introduced in [13, 21, 20, 39, 40], among others.

The main iteration of modern IR methods for smooth constrained optimization [15, 19, 21, 23, 39, 40] proceeds in two phases. In the first (restoration) phase, feasibility is improved without evaluations of the objective function at all. In the second (optimization) phase, one improves the objective function (or a Lagrangian, or an augmented Lagrangian with moderate penalty parameter) on a tangent approximation to the constraints. The resulting trial point is accepted, or not, according to trust region [39, 40], line search [19], or filter criteria [21, 23]. If the trial point is rejected, a new trial point is taken closer to the restored point obtained at the end of the first phase. Convergence to KKT points under regularity assumptions is usually obtained and superlinear convergence can be proved for local versions of the methods [15, 23]. A method that resembles IR ideas in the global optimization field was introduced in [41]. IR has been successfully applied to control problems [24, 12].

The IR approach is useful for the problems that we have in mind for two reasons. On one hand, difficulties associated with the fulfillment of the constraints are transferred to the first phase of the iterations, in which the objective function (whose derivatives are not available) plays no role. Therefore, we can take advantage of well established smooth optimization tools in the restoration phase. On the other hand, the optimization phase needs derivative-free minimization with linear constraints, a problem for which there exist adequate algorithms, in particular, the generating search set (GSS) method introduced by Kolda, Lewis, and Torczon [26, 28].

Although this paper deals with optimization problems in which constraint derivatives are available but function derivatives are not, it is worthwhile to mention that the IR philosophy applies to more general situations. For instance, consider the case in which function derivatives are available too, but evaluation is very expensive. In this case it is better to deal with infeasibility in a way that is independent of evaluating the objective function. Moreover, if some constraint evaluations are expensive and others are not, it is sensible to include the expensive constraints in the objective function in an augmented Lagrangian context with constrained subproblems [3]. In this way, the IR method turns out to be appropriate for solving the constrained augmented Lagrangian subproblems.

This paper is organized as follows. In section 2 we discuss some preliminaries. In section 3 we describe the derivative-free IR algorithm. In section 4 we analyze the algorithm's convergence. In section 5 we discuss implementation details. Numerical experiments are shown in section 6. Finally, section 7 is devoted to conclusions and lines for future research.

Notation. The symbol $\|\cdot\|$ will denote the Euclidean norm on $\mathbb{R}^n$.

We denote $\mathbb{N} = \{0, 1, 2, \ldots\}$.

The set of the nonnegative real numbers is denoted by $\mathbb{R}_+$.

$P_D(x)$ will be the Euclidean projection of x on the closed convex set D.

The Euclidean ball with center x and radius δ will be denoted $B(x, \delta)$.

2. Preliminary background. In this section we discuss sequential optimality conditions, constraint qualifications, and the GSS method. Our IR method will employ the GSS algorithm for solving optimization subproblems. Each GSS execution

finishes when local variations of small size Δ are unsuccessful. The theory of GSS guarantees a small gradient projection when this happens [28]. This property will help us to prove that limit points of sequences generated by the IR method satisfy the approximate gradient projection (AGP) property, which is a sequential necessary optimality condition. The AGP property, together with a weak constraint qualification, implies that the IR algorithm finds KKT points.

We consider the nonlinear programming problem in the form

$$\text{Minimize } f(x) \text{ subject to } h(x) = 0, g(x) \leq 0, \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^n \to \mathbb{R}^p$ are smooth.

Sequential optimality conditions [4, 8, 42] are motivated by the stopping convergence criteria usually employed in algorithms for solving (2.1). The execution of an algorithm generally stops, declaring "success" or "convergence," when some property $\mathcal{P}(\varepsilon)$ is fulfilled by the kth iterate, where $\varepsilon > 0$ is a small tolerance given by the user. For example, if $\mathcal{P}(0)$ represents the KKT conditions, the fulfillment of $\mathcal{P}(\varepsilon)$ indicates the satisfaction of KKT up to the tolerance ε. We say that the sequential optimality condition associated with the property $\mathcal{P}$ is fulfilled at a feasible point x^* if there exists sequences $x^k \to x^*$ and $\varepsilon_k \to 0$ such that $\mathcal{P}(\varepsilon_k)$ holds at x^k for all k. Usually, sequential optimality conditions are fulfilled at local minimizers of (2.1) independently of the satisfaction of regularity assumptions (constraint qualifications). For example, consider the problem of minimizing x_1 subject to $x_1^2 = 0$, whose unique minimizer is $x_1^* = 0$. The KKT conditions do not hold at the solution, but the associated approximate KKT (AKKT) sequential optimality condition does.

We are concerned with the AGP sequential optimality condition introduced in [42]. Given $\eta \in (0, \infty]$, a feasible point x^* is said to satisfy AGP (η) if there exist sequences $\{x^k\} \subset \mathbb{R}^n$ and $\{\varepsilon_k\} \subset \mathbb{R}_+$ such that $x^k \to x^*$, $\varepsilon_k \to 0$, and

$$\|P_{D_k}(x^k - \nabla f(x^k)) - x^k\| \leq \varepsilon_k, \tag{2.2}$$

where D_k is the set of points $x \in \mathbb{R}^n$ defined by

$$\nabla h_i(x^k)^T(x - x^k) = 0 \text{ for all } i = 1, \ldots, m,$$
$$\nabla g_j(x^k)^T(x - x^k) \leq 0 \text{ for all } j \text{ such that } g_j(x^k) \geq 0,$$

and

$$g_j(x^k) + \nabla g_j(x^k)^T(x - x^k) \leq 0 \text{ for all } j \text{ such that } -\eta < g_j(x^k) < 0.$$

It can be proved that AGP(η) is equivalent to AGP(η') for all $\eta, \eta' \in (0, \infty]$ [42]. For this reason we always write AGP instead of AGP(η).

It is interesting to observe that although the exact forms of AKKT and AGP are equivalent, the corresponding sequential conditions are not. (AGP is strictly stronger than AKKT [4].)

Constraint qualifications are properties of the constraints of nonlinear programming problems that when satisfied at a local minimizer x^*, independently of the objective function, imply that x^* fulfills the KKT conditions. In other words, if CQ is a constraint qualification, a necessary optimality condition is that the KKT conditions are fulfilled *or* the CQ condition does not hold. As a consequence, weak constraint qualifications produce strong optimality conditions. The best known constraint qualifications are LICQ (linear independence of the gradients of active constraints) and

Mangasarian–Fromovitz (MFCQ). The constant positive linear dependence condition (CPLD) is a weaker constraint qualification than LICQ and MFCQ, introduced in [50]. In [7] the status of CPLD with respect to other constraint qualifications was elucidated.

Assume that D is the feasible set of (2.1). We say that the CPLD condition holds at $x^* \in D$ if, whenever there exist $I_1 \subseteq \{1, \ldots, m\}$, $I_2 \subseteq \{1, \ldots, p\}$, $\lambda^* \in \mathbb{R}^m$, such that $\lambda_i^* = 0$ if $i \notin I_1$, and $\mu^* \in \mathbb{R}^p_+$ such that $\mu_i^* = 0$ if $i \notin I_2$, satisfying $\|\lambda^*\| + \|\mu^*\| > 0$ and

$$\sum_{i \in I_1} \lambda_i^* \nabla h_i(x^*) + \sum_{i \in I_2} \mu_i^* \nabla g_i(x^*) = 0,$$

we have that there exists $\delta > 0$ such that for all $x \in B(x^*, \delta)$, the gradients $\nabla h_i(x)$, $\nabla g_j(x), i \in I_1, j \in I_2$, are linearly dependent. (In simple words, the "positive linear dependence" of some gradients of active constraints imply that the same gradients remain linearly dependent in a neighborhood of x^*.)

This constraint qualification is weaker than MFCQ. Thus, results ensuring KKT under CPLD are stronger than results in which KKT is guaranteed subject to the fulfillment of MFCQ or LICQ. This attractive feature has already motivated the introduction of augmented Lagrangian derivative-free methods in which optimality in the limit is associated with CPLD [17, 35].

Weaker constraints qualifications that imply the KKT conditions when associated with sequential optimality conditions were recently introduced in [5, 6].

GSS is a derivative-free method for linearly constrained optimization [26, 28]. At each iteration of GSS a finite search set G^k is generated. The directions in G^k generate the tangent cone of the almost active constraints at the current iterate. The norms of these directions are bounded below and above by positive algorithmic parameters and satisfy a minimum angle condition given by another algorithmic parameter. A set of heuristics directions H^k can also be used for the search. The use of these directions does not interfere in the convergence analysis of the algorithm but may be important in practice.

Given a step size t_k, an iteration of GSS is deemed successful if the objective function f has a sufficient decrease along a feasible step $t_k d^k$ for some $d^k \in G^k \cup H^k$. In this case, the algorithm obtains the new iterate taking this step and moves on. The sufficient decrease is measured with a monotone function $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\lim \frac{\rho(t)}{t} = 0$. Usually (for example, at the implementation in [47]), the forcing function ρ has the form αt^2, where $\alpha > 0$ is an algorithmic parameter.

An iteration of the GSS algorithm is deemed unsuccessful if $f(x^k + t_k d^k) \geq f(x^k) - \rho(t_k)$ for all $d^k \in G^k \cup H^k$. In this case the algorithm stays at the current point and reduces the step size by a fixed ratio. The algorithm stops when $t_k < \Delta$, where $\Delta > 0$ is an algorithmic parameter that defines the convergence criterion. In [28] the authors show how to associate the projected gradient at unsuccessful iterates with the step size t_k. As a consequence, the size of the projected gradient at the final point is bounded by a multiple of Δ.

The GSS version implemented in [47] will be used to solve the subproblems of our IR algorithm. We will use all the default parameters of [47], with the exception of the stopping criterion Δ, that will be determined at each iteration of the IR algorithm.

3. IR algorithm. The problem considered in the rest of this paper is

$$\text{Minimize } f(x) \quad \text{subject to} \quad h(x) = 0 \quad \text{and} \quad x \in \Omega, \tag{3.1}$$

where $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, $h : \Omega \to \mathbb{R}^m$, and Ω is a bounded polytope given by

$$\Omega = \{x \in \mathbb{R}^n \mid a_i^T x \leq b_i, i = 1, \ldots, p\}. \tag{3.2}$$

We assume that the polytope Ω is bounded because, in the convergence theory, we need to ensure boundedness of the generated sequence $\{x^k\} \subseteq \Omega$. Boundedness of Ω is the most natural sufficient condition on the problem that guarantees boundedness of the sequence. The algorithm is well defined and finds optimal points even without the boundedness assumption on Ω whenever the sequence $\{x^k\}$ is bounded.

We assume that h is smooth and its first derivatives are available.

For all $x \in \Omega$, $\theta \in (0, 1)$ we define the merit function $\Phi(x, \theta)$ by

$$\Phi(x, \theta) = \theta f(x) + (1 - \theta)\|h(x)\|. \tag{3.3}$$

The main algorithm considered in this paper is defined as follows.

ALGORITHM 1. *Let $r \in [0, 1)$, $\beta > 0$, $\bar{\mu} \geq \gamma > 0$, and $1 < \alpha_l \leq \alpha_u$ be algorithmic parameters. Choose $x^0 \in \Omega$, $\theta_0 \in (0, 1)$. Set $k \leftarrow 0$.*

Step 1. *Restoration.*

Compute $y^k \in \Omega$ such that

$$\|h(y^k)\| \leq r\|h(x^k)\| \tag{3.4}$$

and

$$\|y^k - x^k\| \leq \beta\|h(x^k)\|. \tag{3.5}$$

Step 2. *Penalty parameter.*

If

$$\Phi(y^k, \theta_k) - \Phi(x^k, \theta_k) \leq \frac{1 - r}{2}(\|h(y^k)\| - \|h(x^k)\|) \tag{3.6}$$

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

Else, compute

$$\theta_{k+1} = \frac{(1 + r)(\|h(x^k)\| - \|h(y^k)\|)}{2[f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\|]}. \tag{3.7}$$

Step 3. *Optimization and regularization.*

Choose $\mu \in [\gamma, \bar{\mu}]$.

Step 3.1. *Tangent set descent of the regularized objective function.*

Find $d \in \mathbb{R}^n$ such that $y^k + d \in \Omega$, $\nabla h(y^k)^T d = 0$, and

$$f(y^k + d) + \mu\|d\|^2 \leq f(y^k). \tag{3.8}$$

Step 3.2. *Descent condition for the merit function.*

If

$$\Phi(y^k + d, \theta_{k+1}) \leq \Phi(x^k, \theta_{k+1}) + \frac{1 - r}{2}(\|h(y^k)\| - \|h(x^k)\|) \tag{3.9}$$

define $d^k = d$, $\mu_k = \mu$, $x^{k+1} = y^k + d^k$, update $k \leftarrow k + 1$, and go to Step 1. *Else, update*

$$\mu \in [\alpha_l \mu, \alpha_u \mu] \tag{3.10}$$

and go to Step 3.1.

Remarks.

1. The conditions stated at Step 3.1 are not sufficient to prove convergence to points that satisfy optimality conditions. Note that even the trivial choice $d = 0$ satisfies these requirements. In this case we will still be able to prove feasibility of limit points, but not optimality. The proof of optimality will result from choosing d as an approximate solution of the subproblem

$$\begin{array}{ll} \text{Minimize} & f(y^k + d) + \mu\|d\|^2 \\ \text{subject to} & \nabla h(y^k)^T d = 0, \\ & y^k + d \in \Omega, \end{array} \tag{3.11}$$

   which should satisfy the stopping criterion of GSS with a small tolerance Δ_k. This requirement will be stated as a precise assumption later.
2. An interesting case is when $d = 0$ is the only feasible solution of (3.11). Since the constraints of the minimization subproblem are linear, the KKT conditions of the subproblem are exactly satisfied at $d = 0$. If the current point is feasible, y^k is a solution of the original problem. Otherwise, there will be new restoration phases and so new opportunities of improving feasibility. If $d = 0$ remains the only feasible point at all the iterations, convergence will occur according to the theory that will be presented in this paper.
3. In the present paper we adopted the restoration criterion (3.5), used in [21, 39, 40], instead of the one used by Fischer and Friedlander in [19]. The condition $\|y^k - x^k\| \leq \beta\|h(x^k)\|$ says that y^k should not be very far from x^k; otherwise the algorithm could choose y^k equal to the same feasible point at every restoration phase. Fischer and Friedlander use, instead of (3.5), the related condition $f(y^k) \leq f(x^k) + \beta\|h(x^k)\|$. Under a Lipschitz assumption on f, the latter is implied by (3.5) (with a different constant). However, we prefer to use (3.5) here because this condition does not involve f, which is supposed to be the expensive function. In other words, with the present requirement, the restoration phase does not need to evaluate f at all.
4. The merit function used in this paper is $\Phi(x, \theta) = \theta f(x) + (1-\theta)\|h(x)\|$. As in usual penalty algorithms, very small values of θ reduce the merit function to a mere almost-infeasibility measure. In order to keep a good balance between optimality and feasibility, we do not want very quick reductions of the penalty parameter θ. In [19] the new value of θ is computed as the maximum value of $\{\theta, \theta/2, \theta/4, \ldots\}$ such that (3.6) is satisfied. In the present paper, by means of formula (3.7), we compute the maximum possible θ such that (3.6) is fulfilled. Therefore, our explicit computation of the maximum possible θ yields a larger value of the penalty parameter than the one computed using [19].
5. The practical effectiveness of Algorithm 1 relies strongly on the efficiency of the method used to solve (3.11). Algorithm 1 defines a sequence of alternated restoration and linearly constrained minimization steps. The objective function at the minimization steps is the objective function of the original problem plus a regularization term. This formulation is adequate for the employment of derivative-free minimization on the tangent subspace. The IR formulation [19], which is based on line searches, is not appropriate since in that case, one needs to guarantee that descent is possible along the obtained direction without explicit gradient information.

4. Convergence. Let us first state some assumptions that will be used in this section. The first assumption is that the restoration step is well defined.

Assumption A1. For all $k = 0, 1, 2, \ldots$ it is possible to compute $y^k \in \Omega$ such that (3.4) and (3.5) are fulfilled.

Clearly, this assumption ceases to be satisfied when, for example, the feasible set is empty. Sufficient regularity conditions on the problem that guarantee that Assumption A1 holds have been given in [19, 39]. In [39] the LICQ is used to ensure that Assumption A1 holds in a neighborhood of the solution. In [19] it is shown that if the MFCQ holds at all feasible points of (3.1), then the restoration step is always well defined.

On the other hand, recall that the existence of an "error bound" related to the feasible set means that the distance between x and the feasible set is bounded by a multiple of $\|h(x)\|$. Therefore, if an "error bound" with factor β exists, then for each x^k there exists a feasible $\bar{y}^k$ such that $\|\bar{y}^k - x^k\| \leq \beta\|h(x^k)\|$. In other words, it is possible to find y^k that fulfills conditions (3.4) and (3.5). If β needs to be very large, this means that very large variations in x are necessary to produce small variations in h. This is typical of situations in which constraint qualifications do not hold.

The following assumption states that f satisfies a Lipschitz condition.

Assumption A2. There exists $L_f > 0$ such that

$$|f(x) - f(y)| \leq L_f\|x - y\| \tag{4.1}$$

for all $x, y \in \Omega$.

LEMMA 4.1. *Assume that* A1 *and* A2 *are fulfilled. Then, for all* $k \in \mathbb{N}$, *Steps* 1 *and* 2 *of Algorithm* 1 *are well defined. Moreover, the sequence* $\{\theta_k\}$ *is nonincreasing, the inequality*

$$\Phi(y^k, \theta_{k+1}) - \Phi(x^k, \theta_{k+1}) \leq \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|) \tag{4.2}$$

is fulfilled for all k, *and there exists* $\bar{\theta} > 0$ *such that*

$$\theta_k \downarrow \bar{\theta}. \tag{4.3}$$

Proof. Step 1 is well defined by Assumption A1.

Assuming that $\theta_k > 0$ we will first prove that Step 2 is well defined and that $0 < \theta_{k+1} \leq \theta_k$. Using Assumption A1, if $\|h(y^k)\| - \|h(x^k)\| = 0$, then by (3.4) we have that $\|h(y^k)\| = \|h(x^k)\| = 0$. Therefore by (3.5) $y^k = x^k$. So, $\Phi(x^k, \theta_k) = \Phi(y^k, \theta_k)$. Thus (3.6) holds in this case and consequently $\theta_{k+1} = \theta_k > 0$.

Therefore, it remains to consider only the case in which $\|h(y^k)\| < \|h(x^k)\|$. In this case we obtain that

$$\|h(x^k)\| - \|h(y^k)\| + \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|) = \frac{1+r}{2}(\|h(x^k)\| - \|h(y^k)\|) > 0. \tag{4.4}$$

By direct calculations, the inequality (3.6) is equivalent to

$$\begin{aligned}\theta_k[f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\|] \\ \leq \|h(x^k)\| - \|h(y^k)\| + \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|).\end{aligned} \tag{4.5}$$

Thus, by (4.4) and the fact that $\theta_k > 0$, the requirement (3.6) is fulfilled whenever $f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\| \leq 0$. In this case, the algorithm also chooses $\theta_{k+1} = \theta_k > 0$.

Therefore, we only need to consider the case in which

$$f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\| > 0.$$

In this case, both the numerator and the denominator of (3.7) are positive. So, it turns out that $\theta_{k+1} > 0$ whenever θ_{k+1} is equal to θ_k or it is defined by (3.7). Moreover, if (3.6) does not hold, then by (4.5) we have that

$$\Phi(y^k, \theta) - \Phi(x^k, \theta) > \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|)$$

for all $\theta \geq \theta_k$. Now, since the choice (3.7) obviously implies that

$$\begin{aligned} \theta_{k+1}[f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\|] \\ = \|h(x^k)\| - \|h(y^k)\| + \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|). \end{aligned} \tag{4.6}$$

we conclude that $0 < \theta_{k+1} \leq \theta_k$ in all cases. So, since $\theta_0 \in (0, 1)$ as a initial parameter of the algorithm, the sequence $\{\theta_k\}$ is positive and nonincreasing. Furthermore, by (3.6), (3.7), and (4.6), we have that

$$\Phi(y^k, \theta_{k+1}) - \Phi(x^k, \theta_{k+1}) \leq \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|).$$

It only remains to prove that the sequence $\{\theta_k\}$ is bounded away from zero. For this purpose, it suffices to show that θ_{k+1} is greater than a fixed positive number when it is defined by (3.7). In this case we have that

$$\begin{aligned} \frac{1}{\theta_{k+1}} &= \frac{2[f(y^k) - f(x^k) + \|h(x^k)\| - \|h(y^k)\|]}{(1+r)[\|h(x^k)\| - \|h(y^k)\|]} \\ &\leq \frac{2}{1+r}\left[\frac{|f(y^k) - f(x^k)|}{\|h(x^k)\| - \|h(y^k)\|} + 1\right]. \end{aligned}$$

Thus, by (4.1), (3.4), and (3.5),

$$\frac{1}{\theta_{k+1}} \leq \frac{2}{1+r}\left[\frac{L_f \beta}{1-r} + 1\right]. \tag{4.7}$$

This implies that the sequence $\{1/\theta_{k+1}\}$ is bounded. Therefore, the sequence $\{\theta_k\}$ is bounded away from zero, as we wanted to prove. ∎

Formula (4.7) shows that the penalty parameter θ can be close to zero only if $L_f \beta$ is big. Now, L_f is the Lipschitz constant of f (a characteristic of the problem) which measures the variation of the objective function. It is natural that objective functions with large variations need "bad" penalty parameters, since one needs to penalize the constraints to equilibrate functional variations. Moreover, by (4.7), the cause of θ being small could be the fact that (3.5) holds only for big values of β. This means that the variation of the constraints is small as a function of x, so $\|h(x)\|$ is small far from the solution and very small penalty parameters are necessary in order to compensate the variation of f.

From now on, we employ an additional smoothness condition on h. Essentially, we are going to assume that ∇h satisfies a Lipschitz condition. This is a sufficient condition for the fulfillment of Assumption A3 below.

Assumption A3. There exists $L_h > 0$ such that for all $d \in \mathbb{R}^n$, $y \in \Omega$ such that $y + d \in \Omega$ and $\nabla h(y)^T d = 0$, one has

$$\|h(y+d)\| \leq \|h(y)\| + L_h \|d\|^2. \tag{4.8}$$

Let us prove now that if μ is large enough, then condition (3.8) implies that condition (3.9) is fulfilled.

LEMMA 4.2. *Assume that* A1, A2, *and* A3 *are fulfilled. Let $x^k \in \Omega$ be an iterate computed by Algorithm* 1. *Then, after a finite number of updates* (3.10), *condition* (3.9) *is satisfied. Moreover, there exists $\mu_{bound} > 0$ such that $\mu_k \leq \mu_{bound}$ for all $k \in \mathbb{N}$.*

Proof. By (3.10) and the initialization of μ at Step 3, $\mu \geq \gamma$ at all the IR iterations. Therefore, by (3.8),

$$f(y^k + d) + \gamma \|d\|^2 \leq f(y^k + d) + \mu \|d\|^2 \leq f(y^k).$$

Let us show now that condition (3.9) holds for μ large enough. By Lemma 4.1, there exists $\bar{\theta} > 0$ such that

$$\theta_k \geq \bar{\theta} \text{ for all } k. \tag{4.9}$$

We are going to prove now that if

$$\mu \geq \frac{1 - \bar{\theta}}{\bar{\theta}} L_h, \tag{4.10}$$

then inequality (3.9) is also fulfilled.

By (4.10) we have that

$$(1 - \bar{\theta}) L_h - \bar{\theta} \mu \leq 0. \tag{4.11}$$

By (4.2), Assumption A3, and the definition of Φ, we have

$$\begin{aligned} &\Phi(y^k + d, \theta_{k+1}) - \Phi(x^k, \theta_{k+1}) \\ &\quad = \Phi(y^k + d, \theta_{k+1}) - \Phi(y^k, \theta_{k+1}) + \Phi(y^k, \theta_{k+1}) - \Phi(x^k, \theta_{k+1}) \\ &\quad \leq \theta_{k+1}[f(y^k + d) - f(y^k)] + (1 - \theta_{k+1})(\|h(y^k + d)\| - \|h(y^k)\|) \\ &\qquad + \frac{1 - r}{2}(\|h(y^k)\| - \|h(x^k)\|) \\ &\quad \leq -\theta_{k+1} \mu \|d\|^2 + (1 - \theta_{k+1}) L_h \|d\|^2 + \frac{1 - r}{2}(\|h(y^k)\| - \|h(x^k)\|). \end{aligned} \tag{4.12}$$

But, by (4.9) and (4.11),

$$-\theta_{k+1} \mu \|d\|^2 + (1 - \theta_{k+1}) L_h \|d\|^2 \leq -\bar{\theta} \mu \|d\|^2 + (1 - \bar{\theta}) L_h \|d\|^2 \leq 0. \tag{4.13}$$

By (4.12) and (4.13), it follows that (3.9) holds whenever (4.10) takes place. Thus, after a finite number of updates (3.10), (3.9) is satisfied. By the boundedness of the initial μ, the update rule (3.10), and (4.10), the whole sequence $\{\mu_k\}$ is bounded independently of k. ∎

THEOREM 4.1. *Assume that $\{x^k\}$ and $\{y^k\}$ are generated by Algorithm* 1 *and that Assumptions* A1, A2, *and* A3 *hold. Then,*

$$\lim_{k \to \infty} \|h(x^k)\| = \lim_{k \to \infty} \|h(y^k)\| = \lim_{k \to \infty} \|d^k\| = 0. \tag{4.14}$$

Moreover, $\{x^k\}$ and $\{y^k\}$ admit the same limit points and every limit point is feasible.

Proof. By condition (3.9), for all $k \in \mathbb{N}$,

$$\Phi(x^{k+1}, \theta_{k+1}) \leq \Phi(x^k, \theta_{k+1}) + \frac{1-r}{2}(\|h(y^k)\| - \|h(x^k)\|).$$

Therefore, by (3.4),

$$\Phi(x^{k+1}, \theta_{k+1}) \leq \Phi(x^k, \theta_{k+1}) - \frac{(1-r)^2}{2}\|h(x^k)\|. \tag{4.15}$$

Let us define $\rho_k = (1-\theta_k)/\theta_k$ for all $k \in \mathbb{N}$. By Lemma 4.1 there exists $\bar{\theta} > 0$ such that $\theta_k \geq \bar{\theta}$ for all $k \in \mathbb{N}$. This implies that $\rho_k \leq 1/\bar{\theta} - 1$ for all $k \in \mathbb{N}$. Since $\{\rho_k\}$ is bounded and nondecreasing it follows that

$$\sum_{k=0}^{\infty}(\rho_{k+1} - \rho_k) = \lim_{k\to\infty} \rho_{k+1} - \rho_0 < \infty. \tag{4.16}$$

By compactness, the sequence $\{\|h(x^k)\|\}$ is bounded. Therefore, by (4.16), there exists $c > 0$ such that

$$\sum_{k=0}^{\infty}(\rho_{k+1} - \rho_k)\|h(x^k)\| \leq c < \infty. \tag{4.17}$$

Now, by (3.9),

$$f(x^{k+1}) + \frac{1-\theta_{k+1}}{\theta_{k+1}}\|h(x^{k+1})\| \leq f(x^k) + \frac{1-\theta_{k+1}}{\theta_{k+1}}\|h(x^k)\| - \frac{(1-r)^2}{2\theta_{k+1}}\|h(x^k)\|.$$

Since $\theta_{k+1} < 1$, we have that $\frac{(1-r)^2}{2\theta_{k+1}} > \frac{(1-r)^2}{2}$. So, by the definition of ρ_{k+1},

$$f(x^{k+1}) + \rho_{k+1}\|h(x^{k+1})\| \leq f(x^k) + \rho_{k+1}\|h(x^k)\| - \frac{(1-r)^2}{2}\|h(x^k)\|.$$

Therefore, for all $k \in \mathbb{N}$,

$$f(x^{k+1})+\rho_{k+1}\|h(x^{k+1})\| \leq f(x^k)+\rho_k\|h(x^k)\|+(\rho_{k+1}-\rho_k)\|h(x^k)\|-\frac{(1-r)^2}{2}\|h(x^k)\|.$$

Thus, for all $k \in \mathbb{N}$ we have

$$f(x^k)+\rho_k\|h(x^k)\| \leq f(x^0)+\rho_0\|h(x^0)\|+\sum_{j=0}^{k-1}(\rho_{j+1}-\rho_j)\|h(x^j)\|-\frac{(1-r)^2}{2}\sum_{j=0}^{k-1}\|h(x^j)\|.$$

Therefore, by (4.17),

$$f(x^k) + \rho_k\|h(x^k)\| \leq f(x^0) + \rho_0\|h(x^0)\| + c - \frac{(1-r)^2}{2}\sum_{j=0}^{k-1}\|h(x^j)\|.$$

Thus,

$$\frac{(1-r)^2}{2}\sum_{j=0}^{k-1}\|h(x^j)\| \leq -[f(x^k) + \rho_k\|h(x^k)\|] + f(x^0) + \rho_0\|h(x^0)\| + c.$$

Since $\{\rho_k\}$ is bounded, by the continuity of f and h and the compactness of Ω, it follows that the series $\sum_{k=0}^{\infty}\|h(x^k)\|$ is convergent. Therefore,

$$\lim_{k\to\infty} \|h(x^k)\| = 0.$$

Thus, by (3.5),

$$\lim_{k\to\infty} \|y^k - x^k\| = 0$$

and so the sequences $\{x^k\}$ and $\{y^k\}$ admit the same limit points.

Now, by (3.8) and the fact that $\mu \geq \gamma$, for all $k \in \mathbb{N}$ we have

$$f(x^{k+1}) - f(x^k) = f(x^{k+1}) - f(y^k) + f(y^k) - f(x^k) \leq -\gamma\|d^k\|^2 + f(y^k) - f(x^k).$$

Then, by (3.5) and (4.1),

$$f(x^{k+1}) - f(x^k) \leq -\gamma\|d^k\|^2 + L_f\beta\|h(x^k)\|$$

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

$$f(x^k) \leq f(x^0) - \gamma\sum_{\ell=0}^{k-1} \|d^\ell\|^2 + L_f\beta\sum_{\ell=0}^{k-1} \|h(x^\ell)\|.$$

Since the series $\sum_{k=0}^{\infty} \|h(x^k)\|$ is convergent, there exists $\bar{c} > 0$ such that for all $k \in \mathbb{N}$,

$$f(x^k) \leq f(x^0) + \bar{c} - \gamma\sum_{\ell=0}^{k-1} \|d^\ell\|^2.$$

Thus, since f is bounded below on Ω, the series $\sum_{k=0}^{\infty} \|d^k\|^2$ is convergent and $\|d^k\|$ tends to zero. ∎

Observe that up to now, no optimality condition has been assumed for the approximate solution of (3.11), which means that $d^k = 0$ could be a possible choice for all $k \in \mathbb{N}$. In this case, the algorithm would have performed only restoration steps. Moreover, the existence of derivatives of f has not been assumed at all. Employing a Lipschitz condition for f quarantees the algorithm is well-defined and the limit points are feasible but without additional smoothness assumptions. This feature opens the possibility of using efficient fully derivative-free algorithms for finding useful directions d by solving (3.11).

Assumption A4 imposes that the approximate solution d in (3.11) should be obtained by means of the algorithm GSS for derivative-free minimization with linear constraints [26, 28]. Since $d = 0$ is feasible for problem (3.11), this assumption is consistent with (3.8). This will be used to ensure optimality conditions of the limit points of the IR algorithm.

Assumption A4. For all $k \in \mathbb{N}$, the direction d at Step 3.1 of Algorithm 1 is obtained by solving (3.11) by means of algorithm GSS of [28], using a stopping criterion parameter $\Delta_k > 0$ such that $\lim_{k\to\infty} \Delta_k = 0$.

For deciding to stop the execution, GSS employs a fixed tolerance $\eta > 0$ such that the only linear constraints that are considered in the testing procedure are those that are "almost active" at $y^k + d$ with tolerance η. This includes, of course, the active constraints at $y^k + d$. Recalling the definition of Ω (3.2), it is useful to define

$$A(k,\eta) = \{i \in \{1,\ldots,p\} \mid -\eta \leq a_i^T(y^k + d) - b_i \leq 0\}$$

and

$$\Omega(k,\eta) = \{x \in \mathbb{R}^n \mid a_i^T x \leq b_i \text{ for all } i \in A(k,\eta)\}.$$

$A(k,\eta)$ is the set of indices of the linear constraints that are almost active at the point $y^k + d \in \Omega$ and $\Omega(k,\eta)$ is the polytope Ω considering only the almost active

constraints. Clearly, $\Omega \subseteq \Omega(k,\eta)$ and if $\eta = \infty$, one has that $\Omega = \Omega(k,\eta)$. Moreover, since $y^k \in \Omega$, for d small enough we have that

$$A(k,\eta) = \{i \in \{1,\dots,p\} \mid -2\eta \le a_i^T y^k - b_i \le 0\}.$$

Finally, we define

$$D_k = \{x \in \Omega(k,\eta) \mid \nabla h(y^k)^T (x - y^k) = 0\}.$$

Assumption A5. From now on we will assume that the derivatives of f, although not used in our algorithm at all, exist and are Lipschitz continuous, with constant L_g, for all x in a neighborhood of Ω.

We emphasize that Assumption A5 is necessary to prove Theorem 6.3 of [28]. This result ensures that when one uses the GSS algorithm to minimize a function F subject to linear constraints, then

$$\|P_D(x - \nabla F(x)) - x\| \le c\Delta,$$

where x is a point obtained after an unsuccessful iteration, Δ is the step size at the current GSS iteration, c is a constant that only depends on algorithmic parameters and problem-dependent magnitudes, and D is the subspace defined by the almost active constraints at x.

The termination of the GSS algorithm occurs after an unsuccessful iteration. By Assumption A4 we employ the GSS algorithm as a subproblem solver to minimize $F_k(x) \equiv f(x) + \mu_k \|x - y^k\|^2$ subject to $x - y^k \in D_k$. Therefore, Theorem 6.3 of [28] can be used to determine the optimality properties at x^{k+1}, namely,

$$\|P_{D_k}(x^{k+1} - \nabla f(x^{k+1}) - 2\mu_k(x^{k+1} - y^k)) - x^{k+1}\| \le c_k \Delta_k. \tag{4.18}$$

Moreover, since the algorithmic parameters and the problem magnitudes are naturally bounded, we may assume that $c_k \le c > 0$ for all $k \in \mathbb{N}$.

This property is enough to prove the main convergence result for the IR algorithm (with GSS).

THEOREM 4.2. *Assume that the sequence $\{x^k\}$ is generated by Algorithm* 1 *and that Assumptions* A1–A5 *hold. Let x^* be a limit point of $\{x^k\}$. Then, x^* is feasible and satisfies the AGP condition. Moreover, if the CPLD constraint qualification holds at x^*, then x^* fulfills the KKT conditions.*

Proof. The feasibility of x^* has been proved in Theorem 4.1.

By the contraction property of projections, Assumption A5, and (4.18) we have that

$$\begin{aligned}
&\|P_{D_k}(y^k - \nabla f(y^k)) - y^k\| \\
&\quad \le \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1})) - x^{k+1}\| \\
&\qquad + \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1})) - P_{D_k}(y^k - \nabla f(y^k)) + y^k - x^{k+1}\| \\
&\quad \le \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1})) - x^{k+1}\| + (L_g + 2)\|x^{k+1} - y^k\| \\
&\quad \le \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1})) - P_{D_k}(x^{k+1} - \nabla f(x^{k+1}) - 2\mu_k(x^{k+1} - y^k))\| \\
&\qquad + \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1}) - 2\mu_k(x^{k+1} - y^k)) - x^{k+1}\| + (L_g + 2)\|d^k\| \\
&\quad \le \|2\mu_k(x^{k+1} - y^k)\| \\
&\qquad + \|P_{D_k}(x^{k+1} - \nabla f(x^{k+1}) - 2\mu_k(x^{k+1} - y^k)) - x^{k+1}\| + (L_g + 2)\|d^k\| \\
&\quad \le (2\mu_k + L_g + 2)\|d^k\| + c\Delta_k.
\end{aligned}$$

Therefore, by Lemma 4.2 (boundedness of μ_k), Theorem 4.1 ($\|d^k\| \to 0$), and Assumption A5, we have that

$$\lim_{k\to\infty} \|P_{D_k}(y^k - \nabla f(y^k)) - y^k\| = 0.$$

This implies that the limit point x^* satisfies the AGP optimality condition. Therefore, the desired result follows from Theorem 2.2 of [4]. ∎

5. Implementation. One of the most attractive characteristics of IR methods relies on the freedom to choose the methods for both restoration and optimization phases. This allows one to use suitable methods which exploit the structure of the problem. In Algorithm 1, although Assumption A4 requires the use of a specific algorithm for solving subproblem (3.11), any derivative-free method for linear constraints whose limit points satisfy (4.18) can be used.

The version of GSS used here is the one implemented in HOPSPACK. The sufficient descent condition is defined by $\rho(t) = \alpha t^2$ with $\alpha = 0.01$. The search directions have unitary norm and, after every unsuccessful GSS iteration, the search step is halved. A constraint is declared almost active when its distance to the current point is smaller than 2Δ, where $\Delta > 0$ is the tolerance used for the stopping criterion. See formula (5.4) below.

5.1. Restoration phase. In order to satisfy requirements (3.4) and (3.5), we consider the smooth nonlinear programming problem

$$\text{Minimize } \|y - x^k\|^2 \text{ subject to } h(y) = 0 \text{ and } y \in \Omega. \tag{5.1}$$

In [19] it is proved that if all the feasible points of (3.1) satisfy MFCQ, the global solution of (5.1) satisfies condition (3.5) for some problem-dependent value of β which does not depend on k. For solving (5.1) we employed ALGENCAN, an augmented Lagrangian method described in [3] with the implementation provided in [59]. At each (outer) iteration this method minimizes $\|y - x^k\|^2 + \frac{\rho_{k,j}}{2}\sum_{i=1}^m (h_i(y) + \lambda_i^{k,j}/\rho_{k,j})^2$ subject to $y \in \Omega$ using an active set approach. The approximate solution of each subproblem will be called $y^{k,j}$ and the initial approximation for solving (5.1) is $y^{k,0} = x^k$. The initial penalty parameter $\rho_{k,1}$ is chosen in order to equilibrate the values of the objective function and the constraints of (5.1), and the initial Lagrange multipliers $\lambda_i^{k,0}$ are chosen to be null. Lagrange multipliers are updated according to the classical first-order formula $\lambda^{k,j+1} = \lambda^{k,j} + \rho_{k,j} h(y^{k,j})$. The active set method at each outer iteration stops when an approximate solution of the subproblem is found, in such a way that the norm of the projected gradient of the objective function of the subproblem is less than 10^{-5} times the same norm at the initial point for the subproblem. The ALGENCAN process for solving (5.1) stops when optimality is fulfilled up to a precision ε_k^{opt} and feasibility is fulfilled up to a precision ε_k^{feas}. Since in our restoration case feasibility of (5.1) is much more important than optimality, we employed a loose value for ε_k^{opt} in our tests ($\varepsilon_k^{opt} = 0.1$ at all the IR iterations), which means that the KKT conditions of (5.1) aim to be satisfied up to that moderate precision. On the other hand, we used a rather strict criterion for ε_k^{feas}:

$$\varepsilon_k^{\text{feas}} = \max\{10^{-8}/\sqrt{n}, \min\{\varepsilon_{k-1}^{\text{feas}}, \|h(x^k)\|\}\Delta_k\},$$

where $\varepsilon_0^{\text{feas}} = 0.01$ and Δ_k is specified in the optimization phase.

Employing these criteria, one usually obtains an approximate solution $y^k \in \Omega$ of (5.1) that satisfies $\|h(y^k)\| \le r\|h(x^k)\|$ and $\|y^k - x^k\| \le \beta\|h(x^k)\|$ with $r = 0.99$

and $\beta = 1000$. However, if after a maximum of 20 augmented Lagrangian iterations, a point $y^k \in \Omega$ satisfying those requirements has not been found, the IR algorithm stops declaring failure at the restoration phase.

5.2. Optimization phase. For solving (3.11) in the optimization phase, we use the GSS algorithm [26, 28], which was implemented in the software HOPSPACK [47].

The optimization phase of the IR algorithm may be expensive because it involves objective function evaluations. While condition (3.9) is not satisfied, a linearly constrained derivative-free problem has to be solved for increasing values of μ. According to Lemma 4.2, sufficiently large values for μ produce directions that satisfy this condition, at the cost of taking small step sizes. On the other hand, small values of μ generate large step sizes, but subproblem (3.11) may need to be solved several times until condition (3.9) is fulfilled.

Inspired in Lemma 4.2 we use the following rule to set good values for μ. At the first iteration ($k = 0$), μ is initialized by the formula

$$\mu = 1.01 \min\{\max\{\gamma, \mu'\}, 10^{40}\gamma\},$$

where

$$\mu' = \begin{cases} \dfrac{1-\theta_1}{\theta_1} \dfrac{\|h(x^0)\| - \|h(y^0)\|}{\|x^0 - y^0\|^2} & \text{if } y^0 \neq x^0, \\ \gamma & \text{if } y^0 = x^0. \end{cases}$$

After solving subproblem (3.11), we compute

$$\mu'' = \begin{cases} \dfrac{1-\theta_{k+1}}{\theta_{k+1}} \dfrac{\|h(y^k + d)\| - \|h(y^k)\|}{\|d\|^2} & \text{if } d \neq 0 \\ \mu & \text{if } d = 0, \end{cases} \quad \text{and} \tag{5.2}$$

$$\mu' = 1.01 \min\{\max\{\gamma, \mu''\}, 10^{10}\mu, 10^{40}\gamma\},$$

where μ is the regularization parameter that has just been used in the objective function of (3.11) and d is the approximate solution found by GSS. If condition (3.9) is not satisfied, we update μ by the formula

$$\mu = \max\{\mu', 10\mu\} \tag{5.3}$$

and we solve (3.11) again; otherwise μ' is stored and used in the next iteration for the initialization of μ. At subsequent iterations ($k \geq 1$), μ is initialized by $\mu = \mu'$, where μ' was calculated in the previous iteration, and is updated by rules (5.2)–(5.3).

These implemented rules for initialization and updating of μ are compatible with the rules described in Algorithm 1, with the algorithmic parameters $\bar{\mu}$, α_l, and α_u implicitly determined. Other implementation details used in this phase are given below:

1. We set $\gamma = 2^{-20}$.
2. The initial point for subproblem (3.11) is $d = 0$. If the approximate solution d does not satisfy (3.9), we use this solution as initial point for solving (3.11) again.
   The advantages of using such initial points are that both are feasible solutions and, using the update given by (5.2), if the starting point is declared an approximate solution by GSS, then (3.9) is automatically satisfied.

3. We set $\Delta_0 = 0.5$ and

$$\Delta_{k+1} = \max\{10^{-16}, \min\{\delta_k, 0.1 \max\{\|h(y^k + d^k)\|, \|d^k\|\}\}, \tag{5.4}$$

where $\delta_k = 0.5/(1.1)^k$. Using (5.4) we require more effort in the GSS algorithm when both the infeasibility and the step size are decreasing or when a sufficient large number of iterations has been taken. Since $\lim_{k\to\infty} \delta_k = 0$ the updating formula (5.4) is (numerically) consistent with Assumption A4.

4. The execution of the IR algorithm was stopped whenever $x^{k+1} = y^k + d^k$ was found such that

$$\|h(y^k + d^k)\| \leq 10^{-8} \quad \text{and} \quad \Delta_k \leq 10^{-3}.$$

In addition, the algorithm stopped in the case of a failure of the restoration phase.

6. Numerical experiments. The IR algorithm was implemented in C/C++ language employing the GSS subroutine included in HOPSPACK [47], which is an augmented Lagrangian algorithm developed to deal with general derivative-free problems described in [25]. HOPSPACK also uses GSS to minimize the augmented Lagrangian function subject to linear and bound constraints. For using ALGENCAN in the restoration phase an interface between C++ and Fortran 77 was used. The code of ALGENCAN and its interface with C++ are available in [59]. In order to generate the executable file we used `gfortran-4.2`, `g++-4.2` and `cmake`. The last one was necessary for compiling HOPSPACK. The BLAS and LAPACK libraries (version 3.2.1) were also necessary.

The Hock and Schittkowski [22] test set is widely used to test derivative-free algorithms on moderate-size problems. For running the IR algorithm, the bound constraints of each problem define the set Ω. Linear (not bound) constraints of the original problem were processed in the same way as nonlinear ones. In the case of HOPSPACK linear constraints of the original problem were used in the definition of Ω. A subset of 105 constrained problems, from a set of 116 general nonlinear problems, was selected to test our algorithm. The box-constrained problems (9 problems) and the "badly coded" problems 67 and 85 were ignored. The initial point x^0 was computed as the Euclidean projection on Ω of the initial point x^0_{HS} given in [22]. Note that this process does not involve any objective function evaluation. For the application of the IR algorithm all the inequality constraints were converted to equality constraints by the addition of nonnegative slack variables.

We compared the number of function evaluations used in our method with the augmented Lagrangian method HOPSPACK [47], which implements the algorithm described in [25]. Table 6.1 shows the problems considered in this test. The column Prob. corresponds to the problem number from [22]. The columns Var., Ineq., and Eq. are the number of variables, inequality constraints, and equality constraints, respectively. The values between parentheses represent the number of linear constraints. In HOPSPACK we imposed, as a feasibility convergence criterion, that the norm of $h(x)$ should be smaller than 10^{-8}. For optimality we maintained the default tolerance parameter 10^{-3} which corresponds to the size of the grid in GSS. Note that these criteria are quite compatible with the ones used for IR.

Since both IR and HOPSPACK use GSS for solving subproblems, we believe that comparisons against HOPSPACK are the most relevant. However, for completeness, we also included COBYLA [48], NOMAD [10, 11, 31], and SDPEN [36] in our comparison. These three algorithms do not use GSS at all. We used Miller's Fortran 90

TABLE 6.1
Description of the test problems.

Prob.	Var.	Ineq.	Eq.	Prob.	Var.	Ineq.	Eq.	Prob.	Var.	Ineq.	Eq.
6	2	0	1	43	4	3	0	80	5	0	3
7	2	0	1	44	4	6(6)	0	81	5	0	3
8	2	0	2	46	5	0	2	83	5	6	0
9	2	0	1(1)	47	5	0	3	84	5	6	0
10	2	1	0	48	5	0	2(2)	86	5	10(10)	0
11	2	1	0	49	5	0	2(2)	87	6	0	4
12	2	1	0	50	5	0	3(3)	88	2	1	0
13	2	1	0	51	5	0	3(3)	89	3	1	0
14	2	1	1(1)	52	5	0	3(3)	90	4	1	0
15	2	2	0	53	5	0	3(3)	91	5	1	0
16	2	2	0	54	6	0	1(1)	92	6	1	0
17	2	2	0	55	6	0	6(6)	93	6	2	0
18	2	2	0	56	7	0	4	95	6	4	0
19	2	2	0	57	2	1	0	96	6	4	0
20	2	3	0	58	2	3	0	97	6	4	0
21	2	1(1)	0	59	2	3	0	98	6	4	0
22	2	2(1)	0	60	3	0	1	99	7	0	2
23	2	5(1)	0	61	3	0	2	100	7	4	0
24	2	3(3)	0	62	3	0	1(1)	101	7	6	0
26	3	0	1	63	3	0	2(1)	102	7	6	0
27	3	0	1	64	3	1	0	103	7	6	0
28	3	0	1(1)	65	3	1	0	104	8	6	0
29	3	1	0	66	3	2	0	105	8	1 (1)	0
30	3	1	0	68	4	0	2	106	8	6 (3)	0
31	3	1	0	69	4	0	2	107	9	0	6
32	3	1	1(1)	70	4	1	0	108	9	13	0
33	3	2	0	71	4	1	1	109	9	4 (2)	6
34	3	2	0	72	4	2	0	111	10	0	3
35	3	1(1)	0	73	4	2(1)	1(1)	112	10	0	3(3)
36	3	1(1)	0	74	4	2(2)	3	113	10	8 (3)	0
37	3	2(2)	0	75	4	2(2)	3	114	10	8 (4)	3(1)
39	4	0	2	76	4	3(3)	0	116	13	15 (5)	0
40	4	0	3	77	5	0	2	117	15	5	0
41	4	0	1(1)	78	5	0	3	118	15	29(29)	0
42	4	0	2	79	5	0	3	119	16	0	8(8)

implementation of COBYLA [61]. COBYLA is a sequential trust region algorithm that employs linear approximations of the objective function and the constraints. The linear models come from interpolation at the vertices of a simplex whose regularity is maintained throughout the process. We stopped COBYLA's executions when the size of the simplex became smaller than 10^{-6}. SDPEN is a sequential penalty derivative-free algorithm for nonlinear constrained optimization based on the methods described in [36]. We used the Fortran 90 implementation of SDPEN available at the DFL web page [60]. SDPEN executions were stopped when the steplength became smaller than 10^{-8}. NOMAD [31] is coded in C++ and implements the mesh adaptive direct search algorithm (MADS) from Audet and Dennis [10], a direct search method whose convergence theory is based on Clarke's nonsmooth calculus. MADS is an extension of the generalized pattern search algorithm [9, 58] in which globalization is achieved by simple decrease with integer lattices and constraints are handled using a progressive barrier technique [11]. NOMAD executions stopped when the default feasibility criterion took place and, in addition, the grid size was smaller than 10^{-3}.

One should be cautious in deriving conclusions from the comparison since the amount of problem information used by HOPSPACK, NOMAD, COBYLA, and

SDPEN is significantly less than that used by the IR algorithm. On the other hand, those algorithms have already been proved to be efficient for solving industrial problems and substantial coding effort has been invested in them.

In Tables 6.2 and 6.3 we show the detailed numerical results. In this table, P is the problem's number and, for each algorithm, f is the best functional value, $\|h\|$ is the norm of infeasibility, and #FE is the number of objective function evaluations. There is no information about HOPSPACK in problems 54, 111, and 114 because the execution was interrupted after 20 minutes of CPU time.

Using the results reported in Tables 6.2 and 6.3, we drew graphics of data and performance profiles [18, 46]. The performance measure was the number of objective function evaluations. A problem was considered solved by an algorithm if the obtained solution $\bar{x}$ was such that

$$\|h(\bar{x})\| \leq 10^{-8} \quad \text{and} \quad \frac{|f(\bar{x}) - f_{\mathrm{L}}|}{\max\{1, |f(\bar{x})|, |f_{\mathrm{L}}|\}} \leq 0.1, \tag{6.1}$$

where f_{L} is the lowest objective function value found among the compared algorithms. The results are displayed in Figure 6.1. Algorithmic executions were stopped after 20 minutes of CPU time. This happened only in problems 54, 111, and 114 with HOPSPACK (indicated with (–) in Tables 6.2 and 6.3).

We note that IR failed to solve 11 problems, while COBYLA failed in 14 problems. If only a small budget of function evaluations is tolerated, COBYLA is the most efficient of the methods tested. The reason is that, unlike the other methods, COBYLA uses a (linear) model of the objective function, which is more efficient than using pattern search procedures when the functions are relatively well behaved. In the next section we will see that when one combines GSS with a model-based method (Powell's quadratic BOBYQA) in two nonacademic problems, the robustness of IR dramatically increases.

The numerical results seem to indicate that the IR method with GSS is reliable. In some cases HOPSPACK performed a large number of function evaluations. This is because GSS is adversely affected by the Lipschitz constant associated with the augmented Lagrangian and this constant increases with the value of the penalty parameter, independently of the degree of nonlinearity of the constraints. On the other hand, IR is adversely affected by a big Lipschitz constant on *the gradients* of the constraints, because this is related to the reliability of the linearization. In the extreme case, if the constraints are nearly linear, IR will perform a moderate number of iterations with well-conditioned subproblems for the application of GSS, but HOPSPACK could need to deal with badly conditioned subproblems. The cases in which IR performs many function evaluations are generally due to poor linearization: In some optimization phases, many function evaluations are spent with the objective of minimizing f in a domain that does not represent well the true feasible set. Lack of fulfillment of constraint qualifications can also cause slow convergence. This is the case, for example, of problem 13, in which the CPLD condition does not hold at the minimizer.

6.1. Fitting simulation models. In order to illustrate the applicability of the IR algorithm, we will describe two nonacademic problems. In both cases we used IR with some adjustments, having in mind the specific problems. These problems have the form

$$\text{Minimize } f(x) \text{ subject to } \ h(x) = 0.$$

TABLE 6.2

Numerical results for the IR algorithm, NOMAD, HOPSPACK, COBYLA, and SDPEN in the first 53 *of* 105 *test problems from the Hock and Schittkowski test set. The entries marked with ⋆ correspond to nonsolved problems according to* (6.1).

	IR			NOMAD			HOPSPACK			COBYLA			SDPEN		
P	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE
6	5.57E−10	4E−09	366	⋆ 3.33E+00	⋆ 7E−07	⋆ 83	⋆ 4.84E+00	⋆5E−03	⋆ 151	1.56E−13	6E−12	57	⋆ 4.84E+00	⋆2E−08	⋆ 223
7	−1.73E+00	1E−09	153	⋆ −1.72E+00	⋆ 2E−07	⋆ 73	⋆ 6.93E−01	⋆0E+00	⋆ 325	−1.73E+00	4E−12	63	⋆ −1.31E+00	⋆8E−10	⋆ 233
8	−1.00E+00	2E−09	4	⋆ −1.00E+00	⋆ 1E−06	⋆ 117	⋆9E−03	⋆ 187	−1.00E+00	9E−10	24	⋆ −1.00E+00	⋆3E−08	⋆ 351
9	−5.00E−01	4E−15	117	0E+00	0E+00	186	4E−15	⋆ 26	−5.00E−01	9E−16	47	⋆ 0.00E+00	⋆0E+00	⋆ 111
10	−1.00E+00	7E−09	260	−1.00E+00	0E+00	205	⋆ −8.65E−01	⋆0E+00	⋆ 357	−1.00E+00	2E−12	101	⋆ −7.07E−01	⋆0E+00	⋆ 244
11	−8.50E+00	8E−09	126	−8.50E+00	0E+00	321	−8.48E+00	0E+00	523	−8.50E+00	4E−13	91	⋆ 5.55E+02	⋆0E+00	⋆ 256
12	−3.00E+01	7E−09	685	−3.00E+01	0E+00	117	−3.00E+01	0E+00	342	−3.00E+01	3E−12	57	−2.82E+01	0E+00	244
13	1.09E+00	6E−09	52326	1.00E+00	0E+00	110	−9.97E−01	5E−09	1183	1.00E+00	3E−23	58	−9.99E−01	4E−10	178
14	1.39E+00	3E−09	20	1.39E+00	0E+00	151	⋆ 1.39E+00	⋆4E−06	⋆ 202	⋆ 1.39E+00	1E−16	⋆ 79	⋆ 8.00E+00	⋆2E−11	⋆ 243
15	3.06E+00	7E−14	82	3.06E+00	0E+00	154	3.06E+00	0E+00	451	⋆ 3.60E+00	⋆9E−13	⋆ 52	3.06E+00	0E+00	182
16	2.58E−01	7E−10	546	⋆0E+00	⋆ 208	2.50E−01	0E+00	600	⋆ 3.98E+00	⋆0E+00	2.50E−01	⋆0E+00	0E+00	227
17	1.00E+00	8E−10	118	1.00E+00	3E−17	132	1.00E+00	0E+00	612	1.00E+00	9E−12	38	⋆ 4.00E+00	⋆0E+00	⋆ 163
18	5.00E+00	9E−10	39217	5.00E+00	0E+00	311	⋆ 1.07E+01	⋆0E+00	⋆ 263	5.00E+00	3E−13	95	⋆ 7.25E+00	⋆0E+00	⋆ 244
19	−6.96E+03	3E−09	138	−6.96E+03	0E+00	225	⋆ −7.47E+03	⋆3E−01	⋆ 1370	−6.96E+03	3E−12	46	−6.96E+03	0E+00	1385
20	4.02E+01	8E−10	88	3.82E+01	0E+00	190	3.82E+01	0E+00	393	3.82E+01	0E+00	18	−6.96E+03	0E+00	159
21	−1.00E+02	7E−15	153	−1.00E+02	0E+00	231	−1.00E+02	0E+00	32	−1.00E+02	0E+00	38	−1.00E+02	0E+00	159
22	1.00E+00	9E−09	35	1.00E+00	0E+00	153	1.00E+00	0E+00	276	1.00E+00	0E+00	⋆ 24	⋆ 2.50E+01	⋆0E+00	⋆ 225
23	2.00E+00	7E−09	37	2.00E+00	0E+00	194	2.00E+00	0E+00	466	2.00E+00	0E+00	31	2.00E+00	0E+00	221
24	−1.00E+00	1E−09	132	−1.00E+00	0E+00	119	−1.00E+00	4E−16	27	−1.00E+00	4E−16	20	−1.00E+00	0E+00	214
26	1.58E−07	9E−09	11112	0E+00	⋆0E+00	⋆ 300	⋆ 2.12E+01	⋆0E+00	⋆ 585	7.22E−09	8E−13	⋆ 2.12E+01	⋆4E−16	⋆ 325
27	4.00E+00	7E−09	4135	4.00E+00	0E+00	252	⋆ 4.00E+00	⋆3E−06	⋆ 1358	4.00E+00	1E−12	5041	4.00E+00	0E+00	347
28	2.05E−23	9E−16	510	1.11E−06	0E+00	508	7.70E−08	3E−15	264	6.56E−12	0E+00	414	⋆ 1.30E+01	⋆0E+00	⋆ 325
29	−2.26E+01	6E−10	647	−2.26E+01	0E+00	302	−2.25E+01	0E+00	327	−2.26E+01	4E−12	101	⋆ −1.68E+01	⋆0E+00	⋆ 328
30	1.00E+00	4E−09	790	1.00E+00	0E+00	79	1.00E+00	0E+00	55	1.00E+00	0E+00	86	1.00E+00	0E+00	265
31	6.00E+00	8E−10	526	6.00E+00	0E+00	277	6.00E+00	0E+00	921	6.00E+00	8E−14	84	⋆ 1.00E+01	⋆0E+00	⋆ 267
32	1.00E+00	1E−09	86	⋆ 1.99E+00	⋆6E−17	⋆ 599	1.00E+00	0E+00	51	1.00E+00	2E−16	31	⋆ 1.49E+00	⋆1E−10	⋆ 212
33	⋆ −4.00E+00	⋆5E−09	⋆ 54	−4.59E+00	0E+00	408	−4.59E+00	0E+00	381	−4.59E+00	0E+00	35	⋆ −4.00E+00	⋆0E+00	⋆ 212
34	−8.34E−01	2E−09	228	⋆ −1.12E−01	⋆0E+00	⋆ 463	⋆ −2.28E−01	⋆0E+00	⋆ 582	⋆ −8.34E−01	5E−08	⋆ 43	⋆ 4.88E−02	⋆0E+00	⋆ 325
35	1.11E−01	7E−10	289	1.11E−01	0E+00	221	1.11E−01	0E+00	340	1.11E−01	2.50E−01	⋆5E−08	⋆ 2.50E−01	⋆0E+00	⋆ 322
36	−3.30E+03	1E−12	273	−3.30E+03	0E+00	457	−3.30E+03	0E+00	60	−3.30E+03	0E+00	54	−3.30E+03	0E+00	231
37	−3.46E+03	1E−09	404	−3.46E+03	0E+00	638	−3.46E+03	5E−14	102	−3.46E+03	0E+00	111	−3.43E+03	0E+00	⋆ 350
39	−9.89E−01	4E−09	125	⋆ −1.01E+00	⋆6E−03	⋆ 997	−1.00E+00	0E+00	830	−1.00E+00	1E−12	146	⋆ −2.34E−05	⋆1E−14	⋆ 450
40	−2.50E−01	2E−09	133	⋆ −2.50E−01	⋆1E−06	⋆ 147	⋆ −2.51E−01	⋆2E−03	⋆ 897	−2.50E−01	3E−12	110	−2.33E−01	6E−09	529
41	1.93E+00	1E−09	430	1.93E+00	0E+00	600	1.93E+00	2E−15	292	1.93E+00	0E+00	121	2.00E+00	0E+00	213
42	1.39E+01	2E−09	831	1.40E+01	0E+00	276	1.40E+01	0E+00	779	1.39E+01	2E−12	102	1.40E+01	0E+00	435
43	−4.40E+01	8E−10	1878	−4.40E+01	0E+00	225	−4.40E+01	0E+00	1134	−4.40E+01	2E−12	125	⋆ −2.26E+01	⋆0E+00	⋆ 461
44	⋆ −1.30E+01	⋆2E−09	⋆ 277	−1.50E+01	0E+00	249	⋆ −1.30E+01	⋆0E+00	⋆ 57	−1.50E+01	1E−15	46	−1.50E+01	0E+00	311
46	1.42E−06	1E−09	1485	⋆ 3.34E+00	⋆2E−16	⋆ 246	⋆ 3.34E+00	⋆0E+00	⋆ 777	2.29E−08	4E−13	30580	⋆ 3.34E+00	⋆2E−16	⋆ 541
47	1.15E−08	9E−10	289	⋆ 1.25E+01	⋆0E+00	⋆ 343	⋆ 1.25E+01	⋆0E+00	⋆ 901	1.75E−12	8E−13	137	⋆ 1.25E+01	⋆3E−16	⋆ 537
48	1.07E−24	2E−15	861	⋆ 2.46E+00	⋆0E+00	⋆ 317	1.12E−06	2E−15	497	1.02E−12	2E−16	138	⋆ 8.40E+01	⋆0E+00	⋆ 541
49	1.34E−07	2E−14	20308	⋆ 2.66E+02	⋆0E+00	⋆ 408	1.43E−04	7E−15	1002	1.44E−05	2E−16	2000	⋆ 2.66E+02	⋆2E−16	⋆ 541
50	9.25E−27	6E−14	620	⋆ 1.74E+04	⋆0E+00	⋆ 369	5.29E−07	6E−14	290	3.56E−13	3E−16	295	⋆ 1.82E+04	⋆0E+00	⋆ 605
51	1.94E−27	9E−16	507	⋆ 6.41E−01	⋆0E+00	⋆ 384	⋆ 1.25E−06	9E−16	142	1.56E−13	0E+00	124	⋆ 8.50E+00	⋆0E+00	⋆ 539
52	5.33E+00	3E−09	307	5.34E+00	0E+00	449	⋆ 5.33E+00	⋆8E−06	⋆ 311	5.33E+00	3E−17	226	⋆ 4.50E+02	⋆5E−07	⋆ 44235
53	4.09E+00	3E−09	308	4.37E+00	0E+00	284	4.09E+00	1E−14	216	4.09E+00	4E−16	128	⋆ 7.00E+01	⋆0E+00	⋆ 547
⋆ 54	−1.54E−01	⋆5E−10	⋆ 447	−9.08E−01	0E+00	8074	⋆ −	⋆ −	⋆ 0	⋆ −1.54E−01	⋆1E−12	⋆ 3690	⋆ −8.06E−01	⋆0E+00	⋆ 9000001
55	6.67E+00	3E−09	18	6.71E+00	2E−16	301	⋆ 6.00E+00	⋆1E+00	⋆ 0	6.67E+00	2E−16	51	⋆ 6.81E+00	⋆8E−08	⋆ 2764
56	⋆ −1.06E−06	⋆5E−10	⋆ 21267	⋆ −1.00E+00	⋆9E−16	⋆ 452	⋆ −1.00E+00	⋆1E+00	⋆ 2075	−3.46E+00	8E−12	355	⋆ −1.00E+00	⋆5E−16	⋆ 757
57	3.06E−02	1E−09	287	2.85E−02	0E+00	193	3.06E−02	0E+00	74	3.06E−02	0E+00	40	2.85E−02	0E+00	214
58	3.19E+00	1E−09	102	3.19E+00	0E+00	197	⋆ −6.75E+00	⋆0E+00	817	⋆ 3.19E+00	⋆5E−07	⋆ 35	⋆ 4.00E+00	⋆0E+00	⋆ 217
59	−7.80E+00	4E−09	929	−7.80E+00	0E+00	335	⋆ 5.47E−02	⋆0E+00	⋆ 340	−6.75E+00	⋆0E+00	⋆ 178	⋆ −6.75E+00	⋆0E+00	⋆ 284
60	3.26E−02	8E−10	⋆ 596	⋆ 4.53E+00	⋆9E−07	⋆ 139	⋆ 5.47E−02	⋆2E−03	⋆ 465	3.26E−02	3E−12	83	⋆ 2.39E+01	⋆4E−10	⋆ 365
61	−1.44E+02	3E−09	182	−1.43E+02	0E+00	203	−1.43E+02	0E+00	621	⋆ −8.19E+01	⋆7E−13	⋆ 106	⋆ −1.01E+02	⋆2E−08	⋆ 372

TABLE 6.3
Numerical results for the IR algorithm, NOMAD, HOPSPACK, COBYLA, and SDPEN in the last 52 *of* 105 *test problems from the Hock and Schittkowski test set. The entries marked with* ⋆ *correspond to nonsolved problems according to* (6.1).

	IR			NOMAD			HOPSPACK			COBYLA			SDPEN		
P	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE	f	$\|h\|$	#FE
62	−2.63E+04	2E−16	848	−2.63E+04	2E−15	⋆ 418	−2.63E+04	1E−16	⋆ 233	−2.63E+04	3E−17	455	−2.53E+04	3E−18	⋆ 308
63	9.62E+02	4E−09	171	⋆ 9.66E+02	⋆ 6E−06	⋆ 128	⋆ 5E−04	5E−04	⋆ 317	9.62E+02	2E−12	91	⋆ 9.72E+02	⋆ 5E−08	⋆ 963
64	6.30E+03	3B−09	1025	6.30E+03	0E+00	741	6.30E+03	0E+00	6680	6.30E+03	2E−17	678	6.39E+03	0E+00	577
65	9.54E−01	5D−09	1840	9.54E−01	0E+00	322	1.01E+00	0E+00	379	9.54E−01	4E−12	104	⋆ 5.22E+00	⋆ 0E+00	⋆ 219
66	5.18E−01	4E−09	406	5.32E−01	0E+00	511	5.33E−01	0E+00	566	5.18E−01	4E−13	81	5.32E−01	0E+00	323
68	−9.20E−01	9E−09	10424	⋆ −3.08E−01	⋆ 4E−17	⋆ 346	⋆ −8.44E−01	⋆ 2E−04	⋆ 1316	−9.20E−01	2E−13	3173	⋆ −3.72E−01	⋆ 4E−17	⋆ 392
69	−9.57E+02	2B−09	4130	⋆ −8.46E+02	⋆ 2E−17	⋆ 486	⋆ −9.57E+02	⋆ 1E−04	⋆ 2471	−9.57E+02	6E−14	16821	−9.49E+02	1E−17	380
70	⋆ 2.69E−01	⋆ 1E−09	⋆ 5563	7.50E−03	0E+00	1134	7.74E−03	0E+00	3766	7.50E−03	0E+00	155487	7.50E−03	0E+00	4075
71	1.70E+01	9E−09	4586	⋆ 1.82E+01	⋆ 6E−08	⋆ 107	⋆ 1.70E+01	⋆ 5E−07	⋆ 1939	1.70E+01	7E−14	101	1.60E+01	0E+00	217
72	7.28E+02	3B−09	58181	7.33E+02	1E−13	1604	7.28E+02	0E+00	18188	7.28E+02	6E−18	1782	⋆ 8.82E+02	⋆ 9E−12	⋆ 9000003
73	2.99E+01	1E−09	211	3.00E+01	0E+00	464	3.02E+01	0E+00	223	2.99E+01	7E−17	41	⋆ 1.27E+02	⋆ 7E−11	⋆ 445
74	5.13E+03	6B−10	392	⋆ 5.17E+03	⋆ 8E−07	⋆ 47347	⋆ 5.14E+03	⋆ 1E+00	⋆ 46145	5.13E+03	6E−14	3276	⋆ 1.26E+03	⋆ 6E−01	⋆ 406
75	5.17E+03	1E−09	139	5.13E+03	⋆ 3E−02	⋆ 9883	⋆ 5.23E+03	⋆ 2E−01	⋆ 22678	⋆ 5.27E+03	⋆ 2E−04	⋆ 3112	⋆ 1.26E+03	⋆ 5E−01	⋆ 406
76	−4.68E+00	6B−10	505	−4.68E+00	0E+00	472	−4.68E+00	0E+00	403	−4.68E+00	5E−23	96	−4.47E+00	0E+00	375
77	2.42E−01	2B−09	790	⋆ 2.40E+02	⋆ 9E−08	⋆ 358	⋆ 4.68E+00	⋆ 2E−04	⋆ 1904	2.42E−01	5E−13	158	⋆ 1.65E+03	⋆ 6E−09	⋆ 654
78	−2.92E+00	8D−09	566	⋆ −2.15E+00	⋆ 1E−05	⋆ 213	⋆ −2.89E+00	⋆ 4E−03	⋆ 869	−2.92E+00	4E−12	148	⋆ −8.52E−01	⋆ 2E−08	⋆ 600
79	7.88E−02	6B−09	362	⋆ 8.86E+01	⋆ 1E−06	⋆ 203	⋆ 2.42E−01	⋆ 1E−03	⋆ 1054	7.88E−02	7E−13	120	⋆ 1.00E+02	⋆ 9E−09	⋆ 607
80	5.39E−02	2E−09	658	⋆ 2.09E−01	⋆ 5E−06	⋆ 210	⋆ 1.00E+00	⋆ 5E−03	⋆ 557	5.39E−02	3E−12	138	⋆ 1.00E+00	⋆ 7E−09	⋆ 604
81	5.39E−02	6B−09	770	⋆ 3.32E−01	⋆ 3E−06	⋆ 196	⋆ 1.00E+00	⋆ 5E−03	⋆ 557	5.39E−02	6E−12	173	⋆ 1.00E+00	⋆ 7E−09	⋆ 604
83	−3.07E+04	2E−09	450	−3.07E+04	0E+00	1306	−3.07E+04	0E+00	1729	−3.07E+04	4E−14	106	−3.00E+04	0E+00	452
84	−5.28E+01	4E−10	338	−5.23E+01	0E+00	1734	−5.28E+01	0E+00	2842	−5.28E+01	1E−10	89	−5.07E+01	0E+00	402
86	−3.23E+01	3E−10	437	−3.23E+01	0E+00	534	−3.23E+01	0E+00	485	−3.23E+01	2E−16	111	⋆ −2.77E+01	⋆ 0E+00	⋆ 632
87	8.85E+03	1E−09	860	9.14E+03	4E−09	498	⋆ 9.33E+03	⋆ 7E−01	⋆ 16244	⋆ 9.22E+03	⋆ 8E−07	⋆ 87070	9.28E+03	7E−09	1020
88	1.37E+00	8D−12	105	1.36E+00	0E+00	280	1.36E+00	0E+00	1341	1.36E+00	3E−12	168	1.36E+00	0E+00	297
89	1.36E+00	4E−09	510	1.36E+00	0E+00	423	1.36E+00	0E+00	2208	1.36E+00	4E−12	312	1.36E+00	0E+00	344
90	1.37E+00	4E−09	1304	1.36E+00	0E+00	712	1.36E+00	0E+00	2885	1.36E+00	2E−12	1744	1.36E+00	0E+00	598
91	1.36E+00	5E−09	1305	1.37E+00	0E+00	847	1.36E+00	0E+00	4470	1.36E+00	2E−12	377	1.38E+00	0E+00	755
92	1.37E+00	3E−09	3235	1.36E+00	0E+00	1068	1.36E+00	0E+00	4672	1.36E+00	3E−12	3877	1.37E+00	0E+00	1000
93	1.48E+02	2E−09	2006	1.35E+02	0E+00	527	1.37E+02	0E+00	129	1.35E+02	7E−13	4958	1.37E+02	0E+00	644
95	1.56E−02	2E−09	741	1.56E−02	0E+00	148	1.71E−02	0E+00	156	1.56E−02	2E−16	44	⋆ 1.25E+00	⋆ 0E+00	⋆ 340
96	1.56E−02	5D−10	⋆ 668	1.56E−02	0E+00	148	⋆ 1.71E−02	⋆ 0E+00	156	1.56E−02	2E−16	44	⋆ 1.25E+00	⋆ 0E+00	⋆ 340
⋆ 97	4.07E+00	5D−10	⋆ 5654	3.14E+00	0E+00	395	⋆ 4.12E+00	⋆ 0E+00	⋆ 145	3.14E+00	5E−20	172	⋆ 4.07E+00	⋆ 0E+00	⋆ 348
98	3.14E+00	8D−10	1540	3.14E+00	0E+00	⋆ 362	⋆ 4.12E+00	⋆ 0E+00	⋆ 145	3.14E+00	3E−23	73	⋆ 4.07E+00	⋆ 0E+00	⋆ 347
⋆ 99	−8.31E+08	2D−09	4	⋆ −8.16E+08	⋆ 2E+00	⋆ 311	⋆ −7.46E+08	⋆ 3E+03	⋆ 729	⋆ −8.31E+08	⋆ 5E−07	⋆ 761	⋆ −7.85E+08	⋆ 2E+03	⋆ 720
⋆ 100	5.18E+03	⋆ 7D−10	1314	6.81E+02	0E+00	1336	6.84E+02	0E+00	873	6.81E+02	2E−11	380	6.85E+02	0E+00	771
101	1.81E+03	1E−09	267	⋆ 2.54E+03	⋆ 0E+00	⋆ 1595	1.82E+03	0E+00	14614	⋆ 7.91E+04	⋆ 8E+04	⋆ 186	1.97E+03	0E+00	1195
102	⋆ 1.13E+03	⋆ 9E−10	⋆ 57697	⋆ 1.42E+03	⋆ 0E+00	⋆ 1165	9.20E+02	0E+00	15222	9.12E+02	2E−12	3589554	⋆ 1.06E+03	⋆ 0E+00	⋆ 1497
103	1.55E+03	⋆ 3E−12	⋆ 22	⋆ 1.68E+03	⋆ 0E+00	⋆ 2047	5.44E+02	0E+00	14583	⋆ 3.00E+03	⋆ 2E−01	⋆ 529216	⋆ 7.56E+02	⋆ 0E+00	⋆ 1115
104	3.95E+00	6E−09	28794	4.08E+00	0E+00	681	3.95E+00	0E+00	9844	3.95E+00	2E−12	2567	4.14E+00	0E+00	870
105	1.14E+03	1E−09	25324	1.14E+03	0E+00	1787	1.14E+03	0E+00	10580	1.14E+03	0E+00	1635459	1.14E+03	0E+00	14730
106	7.05E+03	9E−09	68430	⋆ 6.84E+03	⋆ 3E−01	⋆ 861	⋆ 1.14E+04	⋆ 0E+00	⋆ 30424	7.05E+03	4E−11	2171406	⋆ 1.42E+04	⋆ 0E+00	⋆ 1010
107	5.06E+03	4E−09	265	⋆ 5.23E+03	⋆ 2E−05	⋆ 795	⋆ 5.06E+03	⋆ 3E−03	⋆ 7232	5.06E+03	2E−13	194	⋆ 6.19E+03	⋆ 4E−07	⋆ 1493
108	⋆ −5.00E−01	⋆ 1E−09	⋆ 139823	−8.44E−01	0E+00	2322	⋆ −5.00E−01	⋆ 0E+00	⋆ 99	−5.00E−01	⋆ 2E−12	⋆ 192	⋆ −4.29E−01	⋆ 0E+00	⋆ 885
109	5.39E+03	6B−10	⋆ 62792	⋆ 5.21E+03	⋆ 1E+01	⋆ 244247	⋆ 5.50E+03	⋆ 2E−02	⋆ 57551	5.63E+03	4E−11	671257	⋆ 1.15E+04	⋆ 6E+02	⋆ 1771
111	⋆ −4.28E+01	⋆ 2E−09	⋆ 2470	⋆ −4.33E+01	⋆ 4E−08	⋆ 3255	⋆ —	⋆ —	⋆ —	−4.78E+01	2E−14	9074	⋆ −4.24E+01	⋆ 1E−09	⋆ 908
112	−4.78E+01	3B−09	10907	⋆ −4.14E+01	⋆ 3E−06	⋆ 524	−4.78E+01	8E−16	730	−4.78E+01	5E−17	1688	⋆ −4.09E+01	⋆ 4E−10	⋆ 1077
113	2.43E+01	4E−10	4469	2.61E+01	0E+00	1062	2.54E+01	0E+00	1944	2.43E+01	1E−12	339	⋆ 2.83E+01	⋆ 0E+00	⋆ 1118
114	−1.76E+03	2E−09	108605	⋆ −1.06E+03	⋆ 2E−10	⋆ 2233	⋆ —	⋆ —	⋆ —	⋆ −1.37E+03	⋆ 4E−13	⋆ 13776	⋆ −7.89E+03	⋆ 6E−01	⋆ 1234
116	9.76E+01	5E−09	48605	⋆ 9.41E+01	⋆ 8E−03	⋆ 2387	⋆ 5.00E+01	⋆ 8E−02	⋆ 9131	9.76E+01	2E−13	1113532	⋆ 5.00E+01	⋆ 9E−02	⋆ 2484
117	3.23E+01	5E−10	48155	⋆ 1.71E+02	⋆ 0E+00	⋆ 70018	⋆ 5.40E+01	⋆ 0E+00	⋆ 7190	3.23E+01	3E−13	1566	⋆ 1.30E+03	⋆ 0E+00	⋆ 1219
118	6.65E+02	1E−09	7951	6.83E+02	0E+00	3370	6.65E+02	2E−14	3762	6.65E+02	2E−16	281	6.85E+02	0E+00	1746
119	2.45E+02	2B−09	1987	⋆ 7.91E+02	⋆ 9E−06	⋆ 1236	2.45E+02	5E−15	944	2.45E+02	3E−15	578	⋆ 5.79E+03	⋆ 4E−01	⋆ 2916

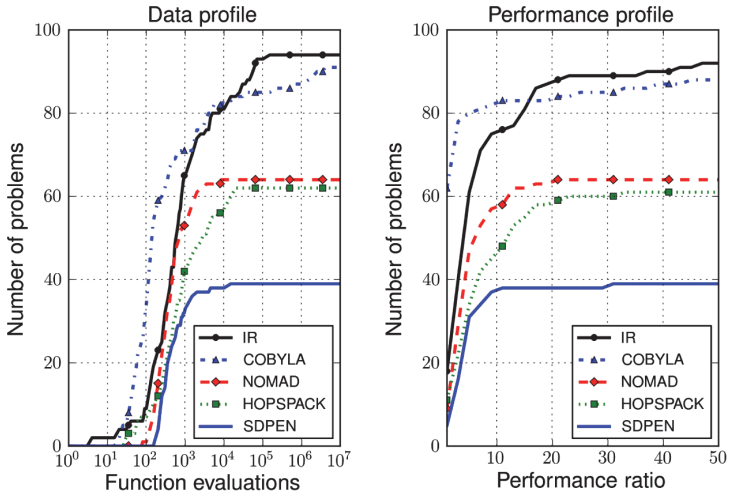

FIG. 6.1. *Data* (a) *and performance* (b) *profiles for the comparison between IR (Algorithm* 1*), COBYLA, NOMAD, HOPSPACK, and SDPEN.*

This allowed us to employ an acceleration of the GSS procedure. Since the subproblems involve minimization with linear equality constraints, they can be reduced to unconstrained subproblems having as variables the coordinates with respect to a null-space basis. We initiated the solution of each subproblem calling Powell's BOBYQA [49] for that unconstrained minimization and switching to GSS at the end of BOBYQA's calls. Since BOBYQA builds quadratic models of the objective function, it can be more efficient than pattern search methods in many cases. Since we switch to GSS at the end, the convergence theory proved in this paper remains valid. Other modifications concern scaling and specific parameters. The models have been simplified to provide clear examples without technicalities.

Asset pricing. A proprietary model defines the price of an asset P at time $t+1$ as a random variable that depends on the price at time t and three model parameters x_1, x_2, x_3. A different asset V depends on y_1, y_2, y_3 under a similar model. By means of simulation we obtain *nsim* trajectories for the prices of these assets on *ntime* periods of time. Given the true (historical) trajectories of P and V over that period we wish to estimate the parameters that maximize the probability of occurrence of those trajectories (or trajectories that are close to the historical ones with a tolerance *tol*). Related models may be found in [2, 29, 55].

The assets are related by phenomenological constraints: x_3 and y_3 must have the same sign, x_1 should be close to y_1, and x_2 should be close to y_2. These characteristics are modeled by the smooth constraints $x_3 y_3 - z_1^2 = 0$, $(x_1 - y_1)^2 + z_2^2 - 0.01 = 0$, and $(x_2 - y_2)^2 + z_3^2 - 0.01 = 0$. Clearly, the evaluation of the objective function is expensive and the derivatives are not available. On the other hand, we have three analytic nonlinear constraints. In a simplified version of the model with which we aim to illustrate the IR approach, the state variable $y(t+1)$ that corresponds to the price of P is defined by $\sigma_{sup} = \min\{0.1, \max\{x_1 + x_3 t, 0\}\}$, $\sigma_{inf} = \min\{0.1, \max\{x_2 + x_3 t, 0\}\}$, $c_{sup} = (1 + \sigma_{sup})y(t)$, and $c_{inf} = y(t)/(1 + \sigma_{inf})$, and $y(t+1)$ is uniformly random between c_{inf} and c_{sup}.

TABLE 6.4
Comparison of the asset pricing problem.

Method	Function	$\|h\|$	#FE
IR	6.34E−2	9.75E−12	551
HOPSPACK	9.86E−1	0.00E+00	2391
NOMAD	9.94E−1	0.00E+00	1026
COBYLA	9.89E−1	0.27E−11	187
SDPEN	5.54E−1	1.70E−11	459

Analogously, we define the random evolution of the prices of V (called $z(t)$). At the ith simulation run we compute $num(i)$, the number of times in which $y(t)$ or $z(t)$ differ from the historical prices in less than tol. The objective function that we minimize is given by $f(x) = 1/\sum_{i=1}^{nsim} num(i)$. The initial point was set to be random between 0 and 0.1. For each method we report the final infeasibility (sup-norm), the best functional value obtained at feasible points, and the number of functional evaluations. The results are given in Table 6.4. Note that all the methods find feasible points, IR finds the local minimizer with lowest functional value, while HOPSPACK, NOMAD, and COBYLA seem to find close local minimizers and the local minimizer found by SDPEN is better than the ones obtained by those methods.

Thin films. The transmission T of a thin absorbing film deposited on a thick transparent substrate (see [56, 57]) is given by

$$T = \frac{A\mathbf{x}}{B - C\mathbf{x} + D\mathbf{x}^2}, \tag{6.2}$$

where

$$A = 16s(\bar{n}^2 + \kappa^2), \tag{6.3}$$

$$B = [(\bar{n}+1)^2 + \kappa^2][(\bar{n}+1)(\bar{n}+s^2) + \kappa^2], \tag{6.4}$$

$$\begin{aligned} C = {} & [(\bar{n}^2 - 1 + \kappa^2)(\bar{n}^2 - s^2 + \kappa^2) - 2\kappa^2(s^2+1)]2\cos\varphi \\ & - \kappa[2(\bar{n}^2 - s^2 + \kappa^2) + (s^2+1)(\bar{n}^2 - 1 + \kappa^2)]2\sin\varphi, \end{aligned} \tag{6.5}$$

$$D = [(\bar{n}-1)^2 + \kappa^2][(\bar{n}-1)(\bar{n}-s^2) + \kappa^2], \tag{6.6}$$

$$\varphi = 4\pi\bar{n}d/\lambda, \quad \mathbf{x} = \exp(-\alpha d), \quad \alpha = 4\pi\kappa/\lambda. \tag{6.7}$$

In formulae (6.3)–(6.7) λ is the wavelength, $s = s(\lambda)$ is the refractive index of the transparent substrate (assumed to be known), $\bar{n} = \bar{n}(\lambda)$ is the refractive index of the film, $\kappa = \kappa(\lambda)$ is the attenuation coefficient of the film (α is the absorption coefficient), and d is the thickness of the film.

Typical transmission curves are given in [14]. We are given $ncurv$ empirical curves representing transmittances of the same material with different (given) thicknesses. Observations take place at wavelengths $\lambda_{min} = \lambda_1 < \ldots < \lambda_{nobs} = \lambda_{max}$. To each empirical transmittance curve $\gamma_i(\lambda)$ we associate the number of its local maximizers $M(\gamma_i)$ and the average transmittance $A(\gamma_i)$. We wish to estimate the functions $\bar{n}(\lambda)$ and $\kappa(\lambda)$ that best fit $M(\gamma_i), i = 1, \ldots, ncurv$. The objective function to be minimized will be $f(\bar{n}, \kappa) = \sum_i |A(\bar{n}, \kappa, d_i) - A(\gamma_i)|^2 + \sum_i |M(\bar{n}, \kappa, d_i) - M(\gamma_i)|^2$, where $M(\bar{n}, \kappa, d_i)$ is the number of local maximizers of the theoretical curve with thickness d_i, refractive index $\bar{n}$, and attenuation coefficient κ, and $A(\bar{n}, \kappa, d_i)$ is the average transmittance of same theoretical curve. The following constraints are imposed to the functions $\bar{n}(\lambda)$ and $\kappa(\lambda)$ [14]:

TABLE 6.5
Comparison on the thin films problem.

Method	Function	$\|h\|$	#FE
IR	1.078E−04	2.120E−05	1668
HOPSPACK	–	–	–
NOMAD	–	–	–
COBYLA	1.700E+01	1.587E−03	3225
SDPEN	3.265E−03	3.400E−07	11579

There exists $\lambda_{infl} \in [\lambda_{min}, \lambda_{max}]$ *such that* $\kappa(\lambda)/\lambda$ *is convex if* $\lambda \leq \lambda_{infl}$ *and concave if* $\lambda > \lambda_{infl}$.

The unknowns of the problem are the values of $\kappa(\lambda)$ at the observed grid wavelengths $\lambda_{min} < \cdots < \lambda_{max}$ and the inflection wavelength λ_{infl}. The nonlinear constraints are represented by

$$[\kappa(\lambda_{i+1})/\lambda_{i+1} - 2\kappa(\lambda_i)/\lambda_i + \kappa(\lambda_{i-1})/\lambda_{i-1}][\lambda_i - \lambda_{infl}] + z_i^2 = 0$$

for all $i = 2, \ldots, nobs-1$. Therefore, the problem has $2\,nobs-1$ variables and $nobs-2$ nonlinear constraints. In the numerical tests we considered $nobs = 101$ (201 variables and 99 nonlinear constraints) and $ncurv = 10$. The initial approximations for the unknowns κ were chosen as in [14], the initial refraction coefficients were null, and the initial inflection point was set to be 0.95 times the maximal wavelength. For the IR algorithm, COBYLA, and SDPEN we report the final infeasibility (sup-norm), the best functional value obtained at feasible points, and the number of functional evaluations. The execution of NOMAD and HOPSPACK was interrupted after 2 hours of execution time. The results are given in Table 6.5.

7. Final remarks. We presented an IR approach for constrained derivative-free optimization. The derivatives of the constraints were supposed to be available. Global convergence to stationary points was proved and an implementation of the proposed algorithm was tested against the derivative-free augmented Lagrangian algorithm HOPSPACK, the Powell's derivative-free algorithm COBYLA, which uses a linear approximation approach, the MADS method NOMAD, and the sequential penalty derivative-free algorithm SDPEN.

The availability of constraint derivatives is an important assumption since in the optimization phase of the algorithm, we solve a linearly constrained derivative-free optimization problem defined by the Jacobian of the constraints. On the other hand, objective function derivatives are not used at all. Moreover, the gradient of f plays a modest role in the convergence theory since it is only associated with the convergence properties of the internal GSS solver [28]. Clearly, the efficiency of this solver is essential to support the effectiveness of the overall algorithm. Nevertheless, the independence of important properties of the IR algorithm with respect to the smoothness of the objective function suggests that variations of the algorithm in which smoothness of f would not be assumed at all should be useful. Further research may be expected on this subject.

IR methods for constrained optimization are closely related to the block-generalized Brown–Brent methods for solving nonlinear systems of equations defined in [38]. This relation was emphasized in [15] in connection with local convergence proofs. In Brown–Brent methods one obtains local quadratic convergence without necessarily using derivatives of the components of the system and employing a different number of evaluations for each component. If the derivatives of all the components except

one are available, it is sensible to divide the system into two blocks, the first containing the derivable components and the second with only one function, for which the gradient is not available. With such decomposition that the Brown–Brent method is quadratically convergent, we take full advantage of the available derivatives, and we need to evaluate the complicated component only twice per iteration. This procedure suggests plausible ideas for considering a natural generalization of problem (3.1). We have in mind the situation in which only the derivatives of some constraints are available, whereas the remaining constraints must be addressed using derivative-free tools. This will be the subject of forthcoming research.

Acknowledgments. We are indebted to associate editor Prof. Margaret Wright and two anonymous referees for many useful comments and remarks that led to significant improvement of this paper.

REFERENCES

[1] J. Abadie and J. Carpentier, *Generalization of the Wolfe reduced-gradient method to the case of nonlinear constraints*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1968, pp. 37–47.

[2] A. Agapie and C. Bratianu, *Repetitive stochastic guesstimation for estimating parameters in a GARCH model*, J. Econom. Forecast., 2 (2010), pp. 213–222.

[3] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *On Augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim., 18 (2007), pp. 1286–1309.

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

[5] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva, *A relaxed constant positive linear dependence constraint qualification and applications*, Math. Program., 135 (2012), pp. 255–273.

[6] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. S. Silva, Two new constraint qualifications and applications, SIAM J. Optim., 22 (2012), pp. 1109–1135.

[7] R. Andreani, J. M. Martínez, and M. L. Schuverdt, *On the relation between the constant positive linear dependence condition and quasinormality constraint qualification*, J. Optim. Theory Appl., 125 (2005), pp. 473–485.

[8] R. Andreani, J. M. Martínez, and B. F. Svaiter, *A new sequential optimality condition for constrained optimization and algorithmic consequences*, SIAM J. Optim., 20 (2011), pp. 3533–3554.

[9] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[10] C. Audet and J. E. Dennis, Jr., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.

[11] C. Audet and J. E. Dennis, Jr., *A progressive barrier for derivative-free nonlinear programming*, SIAM J. Optim., 20 (2009), pp. 445–472.

[12] N. Banihashemi and C. Y. Kaya, *Inexact restoration for Euler discretization of box-constrained optimal control problems*, J. Optim. Theory Appl., 156 (2013), pp. 726–760.

[13] R. H. Bielschowsky and F. A. M. Gomes, *Dynamic control of infeasibility in equality constrained optimization*, SIAM J. Optim., 19 (2008), pp. 1299–1325.

[14] E. G. Birgin, I. Chambouleyron, and J. M. Martínez, *Estimation of the optical constants and the thickness of thin films using unconstrained optimization*, J. Comput. Phys., 151 (1999), pp. 862–880.

[15] E. G. Birgin and J. M. Martínez, *Local convergence of an inexact-restoration method and numerical experiments*, J. Optim. Theory Appl., 127 (2005), pp. 229–247.

[16] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS/SIAM Ser. Optim., SIAM, Philadelphia, 2009.

[17] M. A. Diniz-Ehrhardt, J. M. Martínez, and L. G. Pedroso, *Derivative-free methods for nonlinear programming with general lower-level constraints*, Comput. Appl. Math., 30 (2011), pp. 19–52.

[18] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[19] A. Fischer and A. Friedlander, *A new line search inexact restoration approach for nonlinear programming*, Comput. Optim. Appl., 46 (2010), pp. 333–346.
[20] N. I. M. Gould and Ph. L. Toint, *Nonlinear programming without a penalty function or a filter*, Math. Program., 122 (2010), pp. 155–196.
[21] C. C. Gonzaga, E. Karas, and M. Vanti, *A globally convergent filter method for nonlinear programming*, SIAM J. Optim., 14 (2003), pp. 646–669.
[22] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econom. and Math. Systems 187, Springer, New York, 1981.
[23] E. W. Karas, C. C. Gonzaga, and A. A. Ribeiro, *Local convergence of filter methods for equality constrained non-linear programming*, Optimization, 59 (2010), pp. 1153–1171.
[24] C. Y. Kaya, *Inexact restoration for Runge-Kutta discretization of optimal control problems*, SIAM J. Numer. Anal., 48 (2010), pp. 1492–1517.
[25] T. G. Kolda and J. D. Griffin, *Nonlinearly-Constrained Optimization Using Asynchronous Parallel Generating Set Search*, Technical report SAND2007-3257, Sandia National Laboratories, 2007.
[26] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.
[27] T. G. Kolda, R. M. Lewis, and V. Torczon, *A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints*, Technical report SAND2006-5315, Sandia National Laboratories, 2006.
[28] T. G. Kolda, R. M. Lewis, and V. Torczon, *Stationarity results for generating set search for linearly constrained optimization*, SIAM J. Optim., 17 (2006), pp. 943–968.
[29] R. Kollmann, S. Maliar, B. A. Malin, and P. Pichler, *Comparison of solutions to the multi-country real business cycle model*, J. Econom. Dynam. Control, 35 (2011), pp. 186–202.
[30] L. S. Lasdon, *Reduced gradient methods*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York, 1982, pp. 235–242.
[31] S. Le Digabel, *Algorithm* 909*: NOMAD: Nonlinear optimization with the MADS algorithm*, ACM Trans. Math. Software, 37 (2011), pp. 1–15.
[32] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.
[33] R. M. Lewis and V. Torczon, *Pattern search algorithms for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.
[34] R. M. Lewis and V. Torczon, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM J. Optim., 12 (2002), pp. 1075–1089.
[35] R. M. Lewis and V. Torczon, *A Direct Search Approach to Nonlinear Programming Problems Using an Augmented Lagrangian Method with Explicit Treatment of Linear Constraints*, Technical report WM-CS-2010-01, College of William and Mary, 2010.
[36] G. Liuzzi, S. Lucidi, and M. Sciandrone, *Sequential penalty derivative-free methods for nonlinear constrained optimization*, SIAM J. Optim., 20 (2010), pp. 2614–2635.
[37] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Math. Program., 92 (2002), pp. 37–59.
[38] J. M. Martínez, *Generalization of the methods of Brent and Brown for solving nonlinear simultaneous equations*, SIAM J. Numer. Anal., 16 (1979), pp. 434–448.
[39] J. M. Martínez, *Inexact restoration method with Lagrangian tangent decrease and new merit function for nonlinear programming*, J. Optim. Theory Appl., 111 (2001), pp. 39–58.
[40] J. M. Martínez and E. A. Pilotta, *Inexact restoration algorithms for constrained optimization*, J. Optim. Theory Appl., 104 (2000), pp. 135–163.
[41] J. M. Martínez and F. N. C. Sobral, *Derivative-free constrained optimization on thin domains*, J. Global Optim., DOI: 10.1007/s10898-012-9944-x.
[42] J. M. Martínez and B. F. Svaiter, *A practical optimality condition without constraint qualifications for nonlinear programming*, J. Optim. Theory Appl., 118 (2003), pp. 117–133.
[43] A. Miele, H. Y. Huang, and J. C. Heideman, *Sequential gradient-restoration algorithm for the minimization of constrained functions, ordinary and conjugate gradient version*, J. Optim. Theory Appl., 4 (1969), pp. 213–246.
[44] A. Miele, A. V. Levy, and E. E. Cragg, *Modifications and extensions of the conjugate-gradient restoration algorithm for mathematical programming problems*, J. Optim. Theory Appl., 7 (1971), pp. 450–472.
[45] A. Miele, E. M. Sims, and V. K. Basapur, *Sequential Gradient-Restoration Algorithm for Mathematical Programming Problems with Inequality Constraints, Part* 1*, Theory*, Aero-Astronautics Report 168, Rice University, 1983.

[46] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 20 (2009), pp. 172–191.
[47] T. D. Plantenga, *HOPSPACK* 2.0 *User Manual*, Sandia National Laboratories, Albuquerque, NM, 2009.
[48] M. J. D. Powell, *A direct search optimization method that models the objective and constraint functions by linear interpolation*, in Advances in Optimization and Numerical Analysis, S. Gomez and J.-P. Hennart, eds., Kluwer Academic, Dordrecht, Netherlands, 1994, pp. 51–67.
[49] M. J. D. Powell, *The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives*, Report NA2009/06, University of Cambridge, Cambridge, UK, 2009.
[50] L. Qi and Z. Wei, *On the constant positive linear dependence condition and its application to SQP methods*, SIAM J. Optim., 10 (2000), pp. 963–981.
[51] M. Rom and M. Avriel, *Properties of the sequential gradient-restoration algorithm (SGRA), Part* 1*: Introduction and comparison with related methods*, J. Optim. Theory Appl., 62 (1989), pp. 77–98.
[52] M. Rom and M. Avriel, *Properties of the sequential gradient-restoration algorithm (SGRA), Part* 2*: Convergence analysis*, J. Optim. Theory Appl., 62 (1989), pp. 99–126.
[53] J. B. Rosen, *The gradient projection method for nonlinear programming, Part* 1*, Linear Constraints*, SIAM J. Appl. Math., 8 (1960), pp. 181–217.
[54] J. B. Rosen, *The gradient projection method for nonlinear programming, Part* 2*, Nonlinear Constraints*, SIAM J. Appl. Math., 9 (1961), pp. 514–532.
[55] W. Semmler and G. Gong, *Estimating parameters of real business cycle models*, J. Econom. Behav. Organ., 30 (1996), pp. 301–325.
[56] R. Swanepoel, *Determination of the thickness and optical constants of amorphous silicon*, J. Phys. E, 16 (1983), pp. 1214–1222.
[57] R. Swanepoel, *Determination of surface roughness and optical constants of inhomogeneous amorphous silicon films*, J. Phys. E, 17 (1984), pp. 896–903.
[58] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim. 7 (1997), pp. 1–25.
[59] TANGO (Trustable Algorithms for Nonlinear General Optimization), http://www.ime.usp.br/∼egbirgin/tango/ (2005).
[60] DFL (Derivative-Free Library), http://www.dis.uniroma1.it/∼lucidi/DFL/.
[61] AMFS (Alan Miller's Fortran Software), http://jblevins.org/mirror/amiller/.