

A sequential quadratic programming algorithm that combines merit function and filter ideas

FRANCISCO A. M. GOMES*

Department of Applied Mathematics, IMECC

Universidade Estadual de Campinas

13081-970, Campinas, São Paulo, Brasil.

E-mail: chico@ime.unicamp.br

Abstract

A sequential quadratic programming algorithm for solving nonlinear programming problems is presented. The new feature of the algorithm is related to the definition of the merit function. Instead of using one penalty parameter per iteration and increasing it as the algorithm progresses, we suggest that a new point is to be accepted if it stays sufficiently below the piecewise linear function defined by some previous iterates on the $(f, \|C\|_2^2)$ -space. Therefore, the penalty parameter is allowed to decrease between successive iterations. Besides, one need not to decide how to update the penalty parameter. This approach resembles the filter method introduced by Fletcher and Leyffer [*Math. Program.*, 91 (2001), pp. 239–269], but it is less tolerant since a merit function is still used. Numerical comparison with standard methods shows that this strategy is promising.

Mathematical subject classification: 65K05, 90C55, 90C30, 90C26.

Key words: sequential quadratic programming, merit functions, filter methods.

1 Introduction

In this paper we are concerned with the problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && C(x) = 0 \\ & && l \leq x \leq u \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a C^2 nonlinear function, $C : \mathbb{R}^n \rightarrow \mathbb{R}^m$ represents a set of C^2 nonlinear constraints and we suppose that $-\infty \leq l_i \leq u_i \leq \infty$, for $i = 1, \dots, n$. Naturally, some of the components of x in (1) may be slack variables generated when converting inequality constraints to this form.

Algorithms based on the sequential quadratic programming (SQP) approach are among the most effective methods for solving (1). Some interesting algorithms of this class are given, for example, in [3, 4, 10, 21]. A complete coverage of such methods can be found in [6, 20].

Since SQP algorithms do not require the iterates to be feasible, they have to concern with two conflicting objectives at each iteration: the reduction of the infeasibility and the reduction of function f . Both objectives must be taken in account when deciding if the new iterate is to be accepted or rejected. To make this choice, most algorithms combine optimality and feasibility into one single *merit function*.

*This work was supported by FAPESP grant 2004/05891-1

Filter methods, on the other hand, usually require only one of these two goals to be satisfied, avoiding the necessity of defining a weighted sum of them. Both approaches have advantages and drawbacks. In this paper, these approaches are mixed with the objective of reducing their disadvantages while keeping their best features. To motivate the new algorithm, both methods are briefly introduced below.

1.1 A prototypical SQP algorithm

The algorithm presented in this paper will inherit the structure of the merit function based SQP method proposed in [10], hereafter called the *GMM algorithm*.

This method divides each iteration into two components, a *normal* and a *tangential step*. The first of these components is used to reduce the infeasibility, while the aim of the second is to improve the objective function. A trust region approach is used as the globalization strategy. All of the iterates are required to satisfy the bound constraints $l \leq x \leq u$, so the merit function chosen was the *augmented Lagrangian*, written here (and in [10]) in an unusual way as

$$\mathcal{L}(x, \lambda, \theta) = \theta[f(x) + C(x)^T \lambda] + \frac{(1 - \theta)}{2} \|C(x)\|_2^2. \quad (2)$$

In (2), $\theta \in [0, 1]$ is a “penalty parameter” used as a weight to balance the Lagrangian function (for the equality constrained subproblem), defined as

$$\ell(x, \lambda) = f(x) + C(x)^T \lambda,$$

with a measure of the infeasibility, given by

$$\varphi(x) = \frac{1}{2} \|C(x)\|_2^2.$$

At iteration k , a new point $x_+ = x_k + s$ is accepted if the ratio between the actual and the predicted reduction of the merit function (when moving from x_k to x_+) is greater than a positive constant.

The actual reduction of the augmented Lagrangian at the candidate point x_+ is defined as

$$A_{red}(x_k, s, \theta) = \mathcal{L}(x_k, \theta) - \mathcal{L}(x_k + s, \theta).$$

The predicted reduction of the merit function depends on the strategy used to approximately solve (1). The GMM algorithm approximates (1) by the quadratic programming problem

$$\begin{aligned} \text{minimize} \quad & \bar{Q}(H, x, \lambda, s) = \frac{1}{2} s^T H s + \nabla \ell(x, \lambda)^T s + \ell(x, \lambda) \\ \text{subject to} \quad & A(x)s + C(x) = 0 \\ & l \leq x + s \leq u \end{aligned}$$

where H is a symmetric $n \times n$ matrix and $A(x) = (\nabla C_1(x), \dots, \nabla C_m(x))^T$ is the Jacobian of the constraints.

In this case, denoting

$$M(x, s) = \frac{1}{2} \|A(x)s + C(x)\|_2^2,$$

as the approximation of $\varphi(x)$, the predicted reduction of the augmented Lagrangian merit function is given by

$$P_{red}(H, x, s, \theta) = \theta P_{red}^{opt}(H, x, s) + (1 - \theta) P_{red}^{fsb}(x, s), \quad (3)$$

where

$$P_{red}^{fsb}(x, s) = M(x, 0) - M(x, s) \quad (4)$$

is the predicted reduction of the infeasibility and

$$P_{red}^{opt}(H, x, \lambda, s) = \overline{Q}(H, x, \lambda, 0) - \overline{Q}(H, x, \lambda, s) \quad (5)$$

is the predicted reduction of the Lagrangian.

The penalty parameter θ plays a crucial role in the acceptance of the step. For an augmented Lagrangian of the form (2), $(\theta - 1)/\theta$ can be viewed as the slope of the line that defines the *forbidden region* in the (φ, ℓ) -plane, that is, the semi-space that contains all of the points that are not acceptable at the current iteration. This is illustrated in Figure 1, where the forbidden region is highlighted for different values of θ .

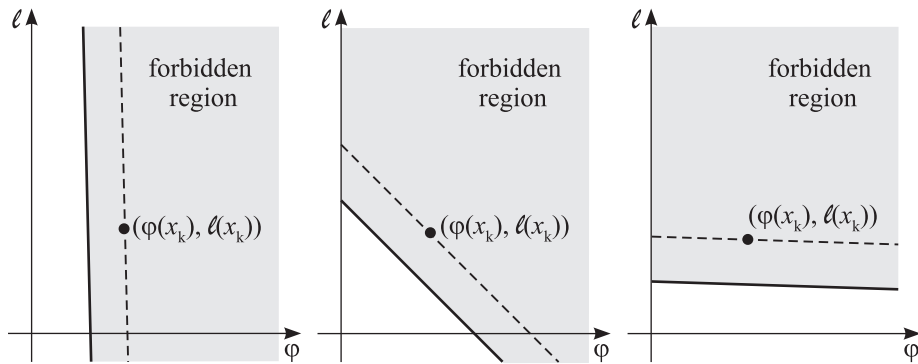


Figure 1: Three merit functions on the (φ, ℓ) -plane, showing the influence of the penalty parameter θ . On the left, $\theta = 1/50$. In the middle, $\theta = 1/2$. On the right, $\theta = 49/50$.

Merit functions have been criticized for many reasons. First, it is not so easy to choose an initial value for θ , since $\ell(x, \lambda)$ and $\varphi(x)$ usually have very different meanings and units. Besides, it is necessary to decrease θ as the algorithm progresses to force it to find a feasible solution. If the initial penalty parameter used is near to 1 and θ is decreased slowly, the algorithm may take too many iterations to reach a feasible point. On the other hand, starting from a small θ or decreasing this factor too quickly may force iterates to stay almost feasible, shortening the steps even when we are far from the optimal solution.

As shown in [10], the adoption of a non-monotone strategy for the reduction of θ is very effective to avoid this premature step shortening. However, it also allows the algorithm to cycle between small and large penalty parameters, inducing some zigzagging in many cases. This undesired behavior can be controlled if an efficient choice of the non-monotonicity parameter is adopted, but this choice is also very problem dependent, so the criticism of the method still applies.

1.2 The filter method

To overcome some of the difficulties inherent to merit functions, Fletcher and Leyffer [8] introduced the idea of using a filter. Instead of combining infeasibility and optimality in one single function, the filter method borrows the idea of nondominance from multi-criteria optimization theory. Thus, a trial point is refused only if it is dominated by some other point, generated in a previous iteration.

This approach was promptly followed by many authors, mainly in conjunction with SLP (sequential linear programming), SQP and interior-point type methods (see, for instance, [1, 5, 6, 7, 9, 11, 12, 15, 16, 17, 22, 23, 24, 25]).

The SQP-filter algorithm presented in [6] illustrates how this nondominance criterion works. In this

method, the objective function f is used to measure optimality, while infeasibility is measured by

$$\phi(x) = \max\{0, \max_{i=1, \dots, m} |C_i(x)|, \max_{i=1, \dots, n} [x_i - u_i], \max_{i=1, \dots, n} [l_i - x_i]\}.$$

Function $\phi(x)$ differs from $\varphi(x)$ in the norm used. Besides, the violation of the bound constraints is also taken in account here, since the filter method do not require the iterates to satisfy these constraints.

For each approximate solution x , we can plot a point $(\phi(x), f(x))$ in the (ϕ, f) -plane, just as we did with $(\varphi(x), \ell(x))$ in Figure 1. A third equivalent plane will be introduced in next section, as the algorithm presented in this paper works with (φ, f) pairs.

Let \mathcal{F} be a set of previously generated pairs in the form (ϕ_j, f_j) . An iterate x_k is accepted by the filter whenever it satisfies

$$\phi(x_k) < (1 - \gamma)\phi_j \quad \text{or} \quad f(x_k) < f_j - \gamma\phi(x_k) \quad \text{for all } (\phi_j, f_j) \in \mathcal{F}, \quad (6)$$

where $\gamma \in (0, 1)$ is a constant.

One advantage of this type of Pareto dominance criterion is that it does not require the redefinition of a penalty parameter at each iteration. However, to avoid the acceptance of iterates that are too close to the points in \mathcal{F} , it was necessary to add to the last inequality of (6) a term that depends on the infeasibility, giving a merit function flavor to this specific filter.

The main disadvantage of the SQP-filter based on (6) is that this acceptance criterion is too tolerant. In fact, requiring only the infeasibility *or* the optimality to be improved makes possible the acceptance of points that are only marginally less infeasible but have a large increase in $f(x)$ over the current iterate, or vice-versa.

Even though, the SQP-filter method can give us some good hints on how to improve the algorithms based on merit functions.

The first hint is that the same merit function that is reliable for points in the (φ, f) -plane that are near to $(\varphi(x_k), f(x_k))$ may be not so useful when the step is large, so the trial point is far from the current iterate. As illustrated in Fig.1, for values of θ near to 1, the acceptance criterion based on a merit function cuts off a significant portion of the feasible region, including, in many cases, the optimal solution of the problem.

The second good idea behind the SQP-filter method is that a restoration step should be used sometimes. The objective of a restoration is to obtain a point that is less infeasible than the current one and is also acceptable for the filter. In [6, sec. 15.5], a restoration step is computed when the trust region quadratic subproblem is incompatible (i.e. has an empty feasible set). In our method, this strategy will be used whenever staying away from feasibility seems not worthwhile. In other words, infeasible iterates are welcome only if they generate large reductions of the objective function. If the decrease in f is small and the current point is very infeasible, it is better to move off and find a more feasible point.

The last lesson we can take from the SQP-filter method is that feasible points could never be refused. In [6, sec. 15.5], feasible points are never added to \mathcal{F} , so a feasible iterate will always satisfy the first inequality of (6). This assures that the optimal solution will always be accepted by the algorithm and a restoration will always succeed.

1.3 Motivation and structure of the paper

The objective of this paper is to present an algorithm that takes advantages from both the merit function and the filter ideas. The algorithm introduced here is a merit function SQP method, in the sense that it still combines feasibility and optimality in one single function and it still uses penalty parameters. However, not one but several penalty parameters are defined per iteration, each one related to a portion of the (f, φ) -space. These parameters are also automatically computed from some (f, φ) -pairs collected at previous iterations, so no update scheme need to be defined for θ .

This paper is organized as follows. In the next section, we present the piecewise linear function we use to accept or reject points. Section 3 introduces the proposed algorithm. In section 4, we prove that

the algorithm is well defined. Sections 5 and 6 contain the main convergence results. Finally, in section 7 some conclusions are presented, along with lines for future work.

Through the paper, we will omit some (or even all) of the arguments of a function, if this does not lead to confusion. Therefore, sometimes $Q(H, x, s)$ will be expressed as $Q(s)$, for example, if there is no ambiguity on H and x .

2 A merit function that uses several penalty parameters per iteration

As we have seen, a merit function deals with two different concepts: the infeasibility and the optimality of the current point.

In this paper, we will introduce a new merit function that uses information collected at previous iterations to accept new points. Since we compare points generated at different stages of the algorithm, this function cannot be based on the augmented Lagrangian, like in [10], as it would depend on the Lagrange multiplier estimates used and, obviously, these estimates change from one iteration to another. Therefore, we decided to adopt the so called *smooth ℓ_2 merit function*, defined as:

$$\psi(x, \theta) = \theta f(x) + (1 - \theta)\varphi(x). \quad (7)$$

The actual reduction of the ℓ_2 merit function will be given by

$$A_{red}(x, s, \theta) = \theta A_{red}^{opt}(x, s) + (1 - \theta)A_{red}^{fsb}(x, s),$$

where

$$A_{red}^{opt}(x, s) = f(x) - f(x + s) \quad \text{and} \quad A_{red}^{fsb}(x, s) = \varphi(x) - \varphi(x + s).$$

Similarly, the predicted reduction of the merit function will be defined as in (3), replacing (5) by

$$P_{red}^{opt}(H, x, s) = Q(H, x, 0) - Q(H, x, s),$$

where

$$Q(H, x, s) = \frac{1}{2}s^T H s + \nabla f(x)^T s + f(x). \quad (8)$$

Generally, for a trial point to be accepted, it is necessary that the actual reduction of the merit function satisfies

$$A_{red}(x, s, \theta) \geq \eta P_{red}(H, x, s, \theta),$$

where $\eta \in (0, 1)$ is a given parameter.

However, this scheme based on a linear merit function is usually unreliable for trial points that are far from the current iterate. Therefore, we suggest the use of a piecewise linear function to accept or reject new points, which correspond to use several merit functions per iteration.

In order to define the new merit function, let F be a set of p points (φ_i, f_i) in the (φ, f) -plane. Suppose that these pairs are ordered so that $\varphi_1 < \varphi_2 < \dots < \varphi_p$. Suppose also that each point (φ_i, f_i) in F is below the line segment joining (φ_{i-1}, f_{i-1}) and (φ_{i+1}, f_{i+1}) , for $i = 2, \dots, p-1$. Thus the piecewise linear function that passes through all of the points in F is convex.

For each point (φ_i, f_i) in F , define another point $(\bar{\varphi}_i, \bar{f}_i)$ by moving a little towards the southwest (a precise definition of $\bar{\varphi}_i$ and \bar{f}_i is given in (10) and (11) below). Let \bar{F} be the set of points $(\bar{\varphi}_i, \bar{f}_i)$. The convex piecewise linear function that connects the points in \bar{F} is defined by

$$\mathcal{P}(\bar{F}, \varphi) = \begin{cases} \infty, & \text{if } \varphi < \bar{\varphi}_1; \\ \frac{(\bar{f}_i - \bar{f}_{i-1})}{(\bar{\varphi}_i - \bar{\varphi}_{i-1})}\varphi + \frac{(\bar{f}_{i-1}\bar{\varphi}_i - \bar{f}_i\bar{\varphi}_{i-1})}{(\bar{\varphi}_i - \bar{\varphi}_{i-1})}, & \text{if } \bar{\varphi}_{i-1} \leq \varphi < \bar{\varphi}_i; \\ \bar{f}_p - \gamma_s(\varphi - \bar{\varphi}_p), & \text{if } \varphi \geq \bar{\varphi}_p. \end{cases}$$

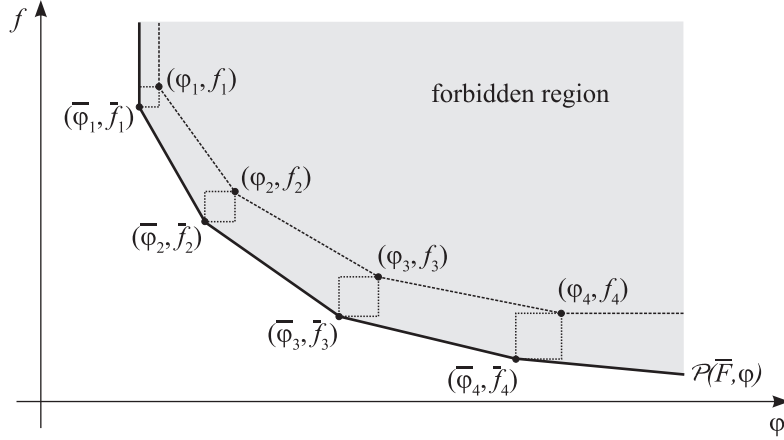


Figure 2: The set F and the piecewise linear function $\mathcal{P}(\bar{F}, \varphi)$.

where γ_s is a small positive constant, such as 10^{-4} .

This new function, illustrated in Fig. 2, is formed by $p + 1$ line segments that can be viewed as merit functions in the form (7). The i -th of these functions is defined by the penalty parameter

$$\bar{\theta}_i = \begin{cases} 0, & \text{if } i = 0; \\ \frac{\bar{\varphi}_{i+1} - \bar{\varphi}_i}{\bar{f}_i - \bar{f}_{i+1} + \bar{\varphi}_{i+1} - \bar{\varphi}_i}, & \text{if } i < p; \\ 1/(1 + \gamma_s), & \text{if } i = p. \end{cases} \quad (9)$$

and a particular choice of η that will be defined below.

The region between the piecewise linear function that passes through the points in F and the function defined by \bar{F} acts as a margin that prevents the acceptance of iterates that are not “sufficiently” better than the pairs in F .

At each iteration k , \bar{F}_k is generated defining, for each point $(\varphi_i, f_i) \in F_k$, another point $(\bar{\varphi}_i, \bar{f}_i)$ such that,

$$\bar{\varphi}_i = \min\{\varphi_i - \gamma_c P_{red}^{f_{sb}}(x_k, s_c), (1 - \gamma_f)\varphi_i\}, \quad (10)$$

and

$$\bar{f}_i = \min\{f_i - \gamma_f P_{red}^{opt}(H_k, x_k, s_c), f_i - (\varphi_i - \bar{\varphi}_i)\}, \quad (11)$$

for some $0 < \gamma_f < \gamma_c < 1$. Reasonable values for these constants are $\gamma_f = 10^{-4}$ and $\gamma_c = 10^{-3}$.

Our algorithm starts with $F_0 = \emptyset$. At the beginning of an iteration, say k , we define the temporary set \mathcal{F}_k as

$$\mathcal{F}_k = F_k \cup \{(f(x_k), \varphi(x_k))\}.$$

As it will become clear in the next section, depending on the behavior of the algorithm, the pair $(f(x_k), \varphi(x_k))$ may be permanently added to F_{k+1} at the end of the iteration. In this case, each pair $(f(x_i), \varphi(x_i))$ that is not below the line segment that joins $(f(x_{i-1}), \varphi(x_{i-1}))$ to $(f(x_{i+1}), \varphi(x_{i+1}))$ is removed from F_{k+1} , to keep the $\mathcal{P}(F, \varphi)$ function convex.

A new iterate $x_+ = x_k + s_c$ is rejected if $f(x_k + s_c)$ is above the piecewise-linear function $\mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c))$ or if we predict a good reduction for the merit function, but the real reduction is deceiving (in the sense that (14) occurs).

To express the first of these conditions in the SQP jargon, we say that x_+ is not accepted if

$$A_{red}(x_k, s_c, \theta_k) \leq \bar{\eta} P_{red}(H_k, x_k, s_c, \theta_k), \quad (12)$$

where

$$\theta_k = \begin{cases} \bar{\theta}_0, & \text{if } \varphi(x_+) < \bar{\varphi}_1; \\ \bar{\theta}_i, & \text{if } \bar{\varphi}_i \leq \varphi(x_+) < \bar{\varphi}_{i+1}; \\ \bar{\theta}_p, & \text{if } \varphi(x_+) \geq \bar{\varphi}_p. \end{cases} \quad (13)$$

The $\bar{\eta}$ parameter is used in (12) to define the region between F_k and \bar{F}_k . However, a formula for $\bar{\eta}$ cannot be written explicitly since we do not know in advance which of the terms in (10) and (11) will be used to define $\bar{\varphi}_i$ and \bar{f}_i . Fortunately, this formula will only be used in the proof of Lemma 15 and, in this case, a simple expression is known.

Some agreement between the model function and the objective function of (1) is an usual second requirement for accepting the step in filter methods. In the SQP-filter method presented in [6], for example, this extra condition is used whenever P_{red}^{opt} is sufficiently greater than a small fraction of the infeasibility at x_k . In this case, s_c is only accepted if $A_{red}^{opt}/P_{red}^{opt} > \gamma_g$ is satisfied. Most filter algorithms, such as those presented in [5, 7, 9, 16, 17, 22, 23] include similar tests.

The combination of infeasibility and optimality into a single merit function allow us to adopt a less stringent condition, rejecting x_+ only if

$$P_{red}^{opt}(H_k, x_k, s_c) \geq \kappa\varphi(x_k) \quad \text{and} \quad \theta_k^{sup} < \gamma_m, \quad (14)$$

where

$$\theta_k^{sup} = \sup\{\theta \in \mathbb{R} \mid A_{red}(x_k, s_c, \theta) \geq \gamma_g P_{red}(H_k, x_k, s_c, \theta)\}, \quad (15)$$

$\kappa > 0$, $\gamma_m \in (0, 1)$ and $\gamma_g \in (0, 1)$.

In words, (14) states that when we predict a good reduction for the optimality part of the merit function, the step is only accepted if there exists a penalty parameter $\theta_k^{sup} \in [\gamma_m, 1]$ such that $A_{red}/P_{red} \geq \gamma_g$.

This condition seems to be somewhat inelegant, since one should expect that all of the iterates that do not belong to the forbidden region shown in Fig. 2 are to be accepted. However, if we choose a small γ_m , say 10^{-4} , (14) becomes less stringent than (12) in most cases, so it is seldom used. In fact, we generally have $\bar{\theta}_0 (\equiv 0) < \gamma_m < \bar{\theta}_1$, so this condition only applies when x_k is the leftmost point in \mathcal{F} . And even in this case, the forbidden region is only slightly enlarged.

Finishing this section, Figure 3 illustrates how the new merit function reduces the region of acceptable points in comparison with the filter. The comparison was made for problem 7 from the Hock and Schittkowski test collection [18]. In the figure, the current iterate, $x_k = (-0.25, 2.27)$ is shown as a black dot. The points in the (φ, f) -space used to define the \mathcal{F} set were $(0.01, -1.5)$, $(2.28, -2.21)$ and $(20, -2.7)$. The forbidden region for the filter method, as defined in (6), is given in dark gray. The points that are acceptable for the filter but not for the new merit function are shown in light gray. The white region contains the points that are acceptable by both criteria. The contour lines of $f(x) = \log(1 + x_1^2) - x_2$ are concave up, while concave down contour lines are related to the constraint $C(x) = (1 + x_1^2)^2 + x_2^2 - 4$. The white concave down narrow region in the bottom of the graph corresponds to points that are almost feasible, i.e., points that satisfy $\varphi(x) < 0.01$.

3 An SQP algorithm

In the general framework of a trust region sequential quadratic programming algorithm, a step s_c is obtained approximating problem (1), in a neighborhood of an iterate x_k , by a quadratic programming (QP) problem.

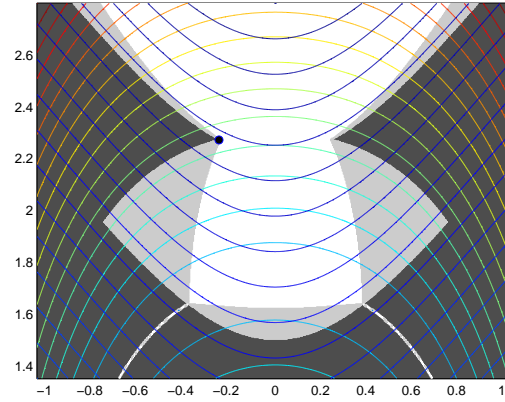


Figure 3: The difference between forbidden regions. Light gray points are accepted by the filter but not by the new merit function.

In our case, this QP problem has the form

$$\begin{aligned}
 & \text{minimize} && Q(H_k, x_k, s) \\
 & \text{subject to} && A(x_k)s + C(x_k) = 0 \\
 & && l \leq x_k + s \leq u \\
 & && \|s\|_\infty \leq \Delta,
 \end{aligned} \tag{16}$$

where $Q(H, x, s)$ is defined by (8), x_k is supposed to belong to

$$\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$$

and H_k is an approximation of the Hessian of the Lagrangian at x_k . It should be noticed that H_k does not need to be positive definite, so this problem must be carefully handled. The infinity norm was chosen here so the bound constraints and the trust region constraint of (16) can be grouped.

We will use the term φ -stationary to say that a point \hat{x} satisfies the first order optimality conditions of

$$\begin{aligned}
 & \text{minimize} && \varphi(x) \\
 & \text{subject to} && x \in \Omega.
 \end{aligned} \tag{17}$$

Unfortunately, if x_k is not φ -stationary, the constraints of (16) may be inconsistent, so this problem may not have a solution. Some algorithms, such as the SQP-filter method presented in [6], include a *restoration* step, called to find a new point that makes (16) compatible. Another common practice to overcome this difficulty is to directly divide the step s_c into two components. The first of these components, called *normal step*, or simply s_n , is obtained as the solution of the feasibility problem

$$\begin{aligned}
 & \text{reduce} && M(x_k, s) \\
 & \text{subject to} && l \leq x_k + s \leq u \\
 & && \|s\|_\infty \leq \beta_d \Delta.
 \end{aligned} \tag{18}$$

where $\beta_d \in (0, 1]$ is a given constant. If $M(x_k, s_n) = 0$, then x_k can be substituted by $x_k + s_n$ in (16) to make this problem feasible, so it can be solved by any QP algorithm. Otherwise, the second component

of s_c , called the *tangential step*, or s_t , is computed so Q is reduced but the predicted reduction of the infeasibility obtained so far is retained. In other words, s_c is the solution of the (now consistent) problem

$$\begin{aligned} & \text{reduce} && Q(H_k, x_k, s) \\ & \text{subject to} && A(x_k)s = A(x_k)s_n \\ & && l \leq x_k + s \leq u \\ & && \|s\|_\infty \leq \Delta. \end{aligned} \tag{19}$$

Usually, β_d is set to some value around 0.8 so the trust region is enlarged from (18) to (19). This is done to prevent s_n from being the the only solution of (19) when this point is in the border of the trust region of (18).

To insure a sufficient decrease of M , a Cauchy point, s_n^{dec} , is computed. This Cauchy point is based on a decent direction for $\varphi(x)$ given by $P_\omega(x_k - \nabla\varphi(x_k))$, the orthogonal projection of $x_k - \nabla\varphi(x_k)$ on Ω . The solution of (18) is required to keep at least ninety percent of the reduction obtained by s_n^{dec} .

A similar procedure is adopted for (19). In this case, s_t^{dec} , the Cauchy point, is obtained from a descent direction for $f(x)$ on the tangent space, given by $P_x(-\nabla Q(s_n))$, the orthogonal projection of $-\nabla Q(s_n)$ on the set

$$\mathcal{T} = \{y \in \mathcal{N}(A(x_k)) \mid (x_k + s_n + y) \in \Omega\}.$$

Again, the decrease on Q obtained by the solution of (19) must not be less than a fixed percentage of the reduction supplied by the Cauchy point.

Besides using this two-step scheme, the algorithm presented here also performs a restoration whenever the step becomes too small and the current point is very infeasible. Although this feasibility reinforcement seems to be unnecessary, the restoration plays a very important role in accelerating the method and keeping the trust region radius sufficiently large.

This role is better explained by an example. Going back to Figure 2, let's suppose that the current iterate x_k is represented by $(\bar{\varphi}_4, \bar{f}_4)$ and that the optimal point has a function value greater than \bar{f}_1 . In this case, unless the trust region radius is sufficiently large and our quadratic model is really good (so the algorithm can jump over a large portion of the forbidden region), it will be necessary to perform a considerable number of short-step iterations to traverse from the southeast to the northwest part of the figure. To shorten the path between the current point and the desired solution, it is necessary to focus only on reducing the infeasibility, and this is exactly what the restoration step does.

One may notice that the use of a standard filter is not useful for circumventing this difficulty. In fact, the problem is aggravated in the presence of (almost) right angles in the frontier of the forbidden region. To see why this happens, let's suppose now that the restoration is used only to ensure that problem (16) is compatible and that x_k is in the vicinity of an almost horizontal segment of the filter envelope. In this case, to escape from this region, it may be necessary to severely reduce the trust region radius before (16) becomes inconsistent, so the restoration is called.

The main steps of the algorithm are given below. We start from $k = 0$ and take $F_0 = \emptyset$ as the initial set of points used to define the piecewise linear function $\mathcal{P}(F)$. An initial point $x_0 \in \Omega$, an initial trust-region radius $\Delta_0 \geq \Delta_{min}$ and an initial symmetric matrix H_0 need also to be given.

Algorithm 1 *A new SQP algorithm*

1. WHILE the stopping criteria are not satisfied
 - 1.1. $\mathcal{F}_k \leftarrow F_k \cup \{(f(x_k), \varphi(x_k))\}$;
 - 1.2. IF $\|C(x_k)\| = 0$ (x_k is feasible),
 - 1.2.1. $s_n \leftarrow 0$;

- 1.3. ELSE
- 1.3.1. Compute d_n (a descent direction for $\varphi(x)$):

$$d_n \leftarrow P_\omega(x_k - \gamma_n \nabla \varphi(x_k)) - x_k;$$
- 1.3.2. Determine s_n^{dec} (the decrease step for $\varphi(x)$), the solution of

$$\begin{aligned} & \text{minimize} && M(x_k, s) \\ & \text{subject to} && l \leq x_k + s \leq u \\ & && \|s\|_\infty \leq \beta_d \Delta_k \\ & && s = t d_n, t \geq 0; \end{aligned}$$
- 1.3.3. Compute s_n (the normal step) such that

$$\begin{aligned} & l \leq x_k + s_n \leq u, \\ & \|s_n\|_\infty \leq \beta_d \Delta_k, \text{ and} \\ & M(x_k, 0) - M(x_k, s_n) \geq \beta_m [M(x_k, 0) - M(x_k, s_n^{dec})]; \end{aligned}$$
- 1.4. Compute d_t (a descent direction for $f(x)$ on the tangent space):

$$d_t \leftarrow P_x(-\gamma_t \nabla Q(s_n));$$
- 1.5. Determine s_t^{dec} (the decrease step for $f(x)$), the solution of

$$\begin{aligned} & \text{minimize} && Q(s) \\ & \text{subject to} && l \leq x_k + s \leq u \\ & && \|s\|_\infty \leq \Delta_k \\ & && s = s_n + t d_t, t \geq 0; \end{aligned}$$
- 1.6. Compute a trial step s_c such that

$$\begin{aligned} & A(x_k) s_c = A(x_k) s_n, \\ & l \leq x_k + s_c \leq u, \\ & \|s_c\|_\infty \leq \Delta_k, \text{ and} \\ & Q(s_n) - Q(s_c) \geq \beta_q [Q(s_n) - Q(s_t^{dec})]; \end{aligned}$$
- 1.7. IF $(f(x_k + s_c) \geq \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c)))$ OR
 $(P_{red}^{opt}(H_k, x_k, s_c) \geq \kappa \varphi(x_k) \text{ AND } \theta_k^{sup} < \gamma_m),$
 $\Delta_k \leftarrow \alpha_R \min\{\Delta_k, \|s_c\|_\infty\};$ (reduce Δ)
- 1.7.1. $\Delta_k \leftarrow \alpha_R \min\{\Delta_k, \|s_c\|_\infty\};$ (reduce Δ)
- 1.8. ELSE
- 1.8.1. $\rho_k \leftarrow A_{red}^{opt}(x_k, s_c) / P_{red}^{opt}(H_k, x_k, s_c);$
- 1.8.2. IF $P_{red}^{opt}(H_k, x_k, s_c) < \kappa \varphi(x_k)$ OR $\rho_k < \gamma_r,$
- 1.8.2.1. $F_{k+1} \leftarrow \mathcal{F}_k;$ (include $(f(x_k), \varphi(x_k))$ in F)
- 1.8.3. ELSE $F_{k+1} \leftarrow F_k;$
- 1.8.4. Accept the trial point:

$$\begin{aligned} & x_{k+1} \leftarrow x_k + s_c; \\ & \tilde{\rho}_k \leftarrow A_{red}(x_k, s_c, \theta_k^{sup}) / P_{red}(H_k, x_k, s_c, \theta_k^{sup}); \\ & \Delta_{k+1} \leftarrow \begin{cases} \max\{\alpha_R \min\{\Delta_k, \|s_c\|_\infty\}, \Delta_{min}\}, & \text{if } \tilde{\rho}_k < \gamma_g, \\ \max\{\alpha_A \Delta_k, \Delta_{min}\}, & \text{if } \tilde{\rho}_k \geq \eta; \end{cases} \\ & \text{Determine } H_{k+1}; \\ & k \leftarrow k + 1; \end{aligned}$$
- 1.9. IF $\Delta_k < \Delta_{rest}$ AND $\varphi(x_k) > \epsilon_h \Delta_k^2,$
- 1.9.1. Compute a restoration step s_r so that
 $(\varphi(x_k + s_r) < \epsilon_h \Delta_k^2 \text{ AND } f(x_k + s_r) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_r)))$ OR
 $x_k + s_r$ is φ -stationary but infeasible;
- 1.9.2. $F_{k+1} \leftarrow \mathcal{F}_k;$ (include $(f(x_k), \varphi(x_k))$ in F)
- 1.9.3. Accept the new point:

$$\begin{aligned} & x_{k+1} \leftarrow x_k + s_r; \\ & \Delta_{k+1} \leftarrow \max\{\beta_r \Delta_{rest}, \Delta_{min}\}; \\ & \text{Determine } H_{k+1}; \\ & k \leftarrow k + 1; \end{aligned}$$

The constants used here must satisfy $0 < \beta_d \leq 1$, $0 < \beta_m < 1$, $0 < \beta_q < 1$, $\kappa > 0$, $0 < \gamma_r < \gamma_g < \eta < 1$, $\gamma_n > 0$, $\gamma_t > 0$, $0 < \Delta_{min} < \Delta_{rest}$, $0 < \alpha_R < 1$, $\alpha_A \geq 1$, $\epsilon_h > 0$ and $\beta_r > 0$. Parameters κ , γ_n , γ_t , Δ_{min} , Δ_{rest} , ϵ_h and β_r are problem dependent and must be chosen according with some measure of problem data. Reasonable values for the remaining parameters might be $\beta_d = 0.8$, $\beta_m = 0.9$, $\beta_q = 0.9$, $\gamma_r = 0.01$, $\gamma_g = 0.05$, $\eta = 0.9$, $\alpha_R = 0.25$ and $\alpha_A = 2.5$. The constant η should not be confused with the parameter $\bar{\eta}$ defined in (12).

Algorithm 1 seems to be somewhat inefficient, since it requires the solution of several quadratic programming problems per iteration. However, as it will become clear at section 7, only steps 1.3.3 and 1.6 usually require the solution of a quadratic problem. Besides, we do not need to solve these problems exactly, so the algorithm is competitive with modern nonlinear programming codes.

The restoration step is called whenever the trust region radius becomes too small compared to $\varphi(x_k)$. In this case, we need to find a point that is less infeasible than x_k and that is also acceptable for the piecewise linear merit function. To ensure that such a point can always be obtained, it would be necessary to use an algorithm for global minimization. Of course, this alternative is unaffordable, so the restoration is computed in practice by an algorithm for solving the box constrained problem (17). The only drawback of this approach is that we cannot guarantee that a stationary but infeasible point would not be reached. Therefore, we say that the main algorithm fails if this happens.

If x_k is feasible, then the condition $P_{red}^{opt} < \kappa\varphi(x_k)$ is never satisfied, since P_{red}^{opt} is always greater or equal to zero. Besides, the condition $A_{red}^{opt} < \gamma_r P_{red}^{opt}$ is also never satisfied when x_k is feasible and $f(x_k + s_c) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c))$. Therefore, all of the points in F_k are infeasible, although \mathcal{F}_k may contain a feasible point. This result is very important for two reasons. First, it prevents the optimal solution of problem 1 from being refused by the algorithm. Moreover, it also assures that the algorithm is well defined, as stated in the next section.

4 The algorithm is well defined

An iteration of algorithm 1 ends only when a new point $x_k + s$ is below the piecewise linear function $\mathcal{P}(\mathcal{F}_k, \varphi(x_k + s))$, besides satisfying some other conditions stated at steps 1.7 or 1.9.1. While such a point is not found, the trust region radius is reduced and the iteration is repeated. It is not obvious that an acceptable point will be obtained, as we may generate a sequence of points that are always rejected by the algorithm. In this section, we prove that the algorithm is well defined, i.e. a new iterate x_{k+1} can always be obtained unless the algorithm stops by finding a φ -stationary but infeasible point or a feasible but not regular point.¹

In the following lemma, we consider the case where x_k is infeasible.

Lemma 1 *If x_k is not φ -stationary, then after a finite number of repetitions of steps 1.1 to 1.9, a new iterate x_{k+1} is obtained by the algorithm.*

Proof: At each iteration k , if $f(x_k + s_c) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c))$ and one of the conditions $P_{red}^{opt}(H_k, x_k, s_c) \leq \kappa\varphi(x_k)$ or $A_{red}(x_k, s_c, \theta_k^{sup}) \geq \gamma_g P_{red}(H_k, x_k, s_c, \theta_k^{sup})$ (for some $\theta_k^{sup} \geq \gamma_m$) is satisfied, then $x_k + s_c$ is accepted and we move to iteration $k + 1$. Otherwise, Δ_k is reduced and after some unfruitful steps, $\Delta_k < \Delta_{rest}$ and $\varphi(x_k) > \epsilon_h \Delta_k^2$, so a restoration is called.

Suppose that a φ -stationary but infeasible point is never reached (otherwise the algorithm fails). As the restoration generates a sequence of steps $\{s_j\}$ converging to feasibility, and since \mathcal{F}_k does not include feasible points (because x_k is infeasible and no feasible point is included in F_k), there must exist an iterate $x_k + s_r$ that satisfies $\varphi(x_k + s_r) < \min\{\bar{\varphi}_1, \epsilon_h \Delta_k^2\}$, so we can proceed to the next iteration. \square

Now, in order to prove that the algorithm is also well defined when x_k is feasible, we need to make the following assumptions.

A1. $f(x)$ and $C_i(x)$ are twice-continuously differentiable functions of x .

¹We say that a point is regular if the linear independence constraint qualification (LICQ) holds.

A2. The sequence of Hessian approximations $\{H_k\}$ is bounded.

As a consequence of A1 and A2, the difference between the actual and the predicted reduction of the merit function is proportional to Δ^2 , so the step is accepted for a sufficiently small trust region radius, as stated in the following lemma.

Lemma 2 *Suppose that A1 and A2 hold and that x_k is feasible and regular for problem 1 but the KKT conditions do not hold. Then, after a finite number of trust region reductions, the algorithm finds a new point $x_k + s_c$ that satisfies $f(x_k + s_c) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c))$ and $A_{red}(x_k, s_c, \theta_k^{sup}) \geq \gamma_g P_{red}(H_k, x_k, s_c, \theta_k^{sup})$ for some $\theta_k^{sup} \geq \gamma_m$.*

Proof: Since x_k is feasible, $s_n = 0$. Supposing that x_k is regular and non-stationary, there must exist a vector $d_t \neq 0$ satisfying

$$l \leq x_k + d_t \leq u, \quad A(x_k)d_t = 0, \quad \text{and} \quad d_t^T \nabla f(x_k) < 0.$$

Let us define, for all $\Delta > 0$,

$$p(\Delta) = t(\Delta)d_t,$$

where

$$t(\Delta) = \max\{t > 0 \mid [x_k, x_k + td_t] \subset \Omega, \text{ and } \|td_t\|_\infty \leq \Delta\}. \quad (20)$$

Clearly, $x + d_t \in \Omega$, so we have that $\|t(\Delta)d_t\|_\infty = \Delta$ whenever $\Delta \leq \|d_t\|_\infty$. Define, in this case,

$$c = -\frac{1}{2}d_t^T \nabla f(x_k) / \|d_t\|_\infty = -\frac{1}{2}d_t^T \nabla Q(0) / \|d_t\|_\infty > 0.$$

Since $Q(s_t^{dec}) \leq Q(p(\Delta))$, by elementary properties of one-dimensional quadratics, there exists $\Delta_1 \in (0, \|d_t\|_\infty]$ such that, for all $\Delta \in (0, \Delta_1)$,

$$Q(0) - Q(s_t^{dec}) \geq -\frac{1}{2}d_t^T \nabla Q(0)t(\Delta) = -\frac{1}{2} \frac{d_t^T \nabla Q(0)}{\|d_t\|_\infty} \Delta = c\Delta.$$

Moreover, since x_k is feasible and $As_n = 0$, we have that $M(x_k, 0) = M(x_k, s_c) = 0$, so

$$P_{red}^{fsb}(x_k, s_c) = 0, \quad \text{and}$$

$$P_{red}^{opt}(H_k, x_k, s_c(\Delta)) = Q(0) - Q(s_c(\Delta)) \geq \beta_q [Q(0) - Q(s_t^{dec})] \geq \beta_q c \Delta. \quad (21)$$

Once x_k is feasible, $(\varphi(x_k), f(x_k))$ is the first pair in \mathcal{F}_k . Thus, there exists $\bar{\Delta}_2 \in (0, \Delta_1]$ such that, for $\Delta < \bar{\Delta}_2$, we need to consider only the portion of $\mathcal{P}(\mathcal{F}_k, \varphi)$ defined on the interval $[0, \varphi_2]$. This linear function may be rewritten so the condition

$$f(x_k + s_c) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c))$$

is equivalent to

$$A_{red}(x_k, s_c(\Delta), \bar{\theta}_1) \geq \eta_1 P_{red}(H_k, x_k, s_c(\Delta), \bar{\theta}_1), \quad (22)$$

where,

$$P_{red}(H_k, x_k, s_c(\Delta), \bar{\theta}_1) = \bar{\theta}_1 P_{red}^{opt}(H_k, x_k, s_c(\Delta)) \geq \beta_q c \Delta \bar{\theta}_1, \quad (23)$$

$$A_{red}(x_k, s_c(\Delta), \bar{\theta}_1) = \bar{\theta}_1 [f(x_k) - f(x_k + s_c(\Delta))] + (1 - \bar{\theta}_1) \varphi(x_k + s_c(\Delta))$$

and $\bar{\theta}_1 > 0$ is given by (9).

Now, by A1, A2 and the definition of P_{red} , we have

$$A_{red}(x_k, s_c, \theta) = P_{red}(H_k, s_k, s_c, \theta) + \bar{c}_1 \|s_c\|^2. \quad (24)$$

So, using (23) and (24) we deduce that

$$\left| \frac{A_{red}(\Delta)}{P_{red}(\Delta)} - 1 \right| \leq \frac{|\bar{c}_1| \Delta}{\beta_q c \bar{\theta}_1}. \quad (25)$$

Thus, for $\Delta < \min\{(1 - \eta_1)\beta_q c \bar{\theta}_1 / |\bar{c}_1|, \bar{\Delta}_2\} = \bar{\Delta}_3$, the inequality (22) necessarily takes place.

Now, using the fact that $\theta_k^{sup} = 1$ for x_k feasible and replacing $\bar{\theta}_1$ by 1 in (25), we can conclude that, for

$$\Delta < \min\{(1 - \gamma_g)\beta_q c / |\bar{c}_1|, \bar{\Delta}_3\} = \bar{\Delta}_4, \quad (26)$$

the condition $A_{red}(x_k, s_c(\Delta), \theta_k^{sup}) \geq \gamma_g P_{red}(H_k, x_k, s_c(\Delta), \theta_k^{sup})$ is also satisfied and the step is accepted. \square

5 The algorithm converges to a feasible point

As mentioned in the last section, our algorithm can stop if a φ -stationary but infeasible point is found. Naturally, this unexpected behavior of the algorithm makes somewhat pretentious the title of this section.

Formally, what we will prove is that, supposing that a φ -stationary but infeasible point is never reached and that the restoration always succeeds, an infinite sequence of iterates converges to feasibility.

In the proofs of the lemmas presented here, we will suppose that A1 and the following assumption are satisfied.

A3. The sequence of iterates $\{x_k\}$ lies within a closed and bounded domain Ω_0 .

This requirement is easy to fulfill, as we usually can define finite lower and upper limits for the variables. Besides, as mentioned in [6, p.730], assumptions A1 and A3 together ensure that, for all k ,

$$f^{min} \leq f(x_k) \leq f^{max} \quad \text{and} \quad 0 \leq \varphi(x_k) \leq \varphi^{max}$$

for some constants f^{min} , f^{max} and $\varphi^{max} > 0$. Our analysis will be based on the fact that the rectangle $A_0 = [0, \varphi^{max}] \times [f^{min}, f^{max}]$ is covered by a finite number of rectangles with area greater than a small constant. Therefore, each time we expand the forbidden region (see fig (2)) by adding to it a small rectangle, we drive the iterates towards feasibility.

Let us start investigating what happens to $\varphi(x)$ when an infinite sequence of iterates is added to F , following the skeleton of Lemma 15.5.2 of [6].

Lemma 3 *Suppose that A1 and A3 hold and that $\{k_i\}$ is any infinite subsequence at which the iterate x_{k_i} is added to F . Then*

$$\lim_{i \rightarrow \infty} \varphi(x_{k_i}) = 0.$$

Proof: Let us suppose, for the purpose of obtaining a contradiction, that there exists an infinite subsequence $\{k_j\} \subseteq \{k_i\}$ for which

$$\varphi(x_{k_j}) \geq \epsilon, \quad (27)$$

where $\epsilon > 0$.

At iteration k_j , the (φ, f) -pair associate with x_{k_j} is included in F at position m , which means that $\varphi_{m-1} \leq \varphi_{k_j} (\equiv \varphi_m) \leq \varphi_{m+1}$ and $f_{m-1} \geq f_{k_j} (\equiv f_m) \geq f_{m+1}$. Thus, as long as the pair (φ_{k_j}, f_{k_j}) remains in F , no other (φ, f) -pair is accepted within the rectangle

$$r_m = \{(\varphi, f) \mid \bar{\varphi}_m \leq \varphi \leq \varphi_m, \bar{f}_m \leq f \leq f_m\}.$$

Notice that, by (10), (11) and (27), the area of this rectangle is, at least,

$$(\varphi_m - \bar{\varphi}_m)(f_m - \bar{f}_m) \geq (\varphi_m - \bar{\varphi}_m)^2 \geq (\gamma_f \varphi_{k_j})^2 \geq \gamma_f^2 \epsilon^2.$$

Assume now that (φ_{k_j}, f_{k_j}) is excluded from F by another pair (φ_{k_l}, f_{k_l}) , included in F at an iteration $k_l > k_j$. This case is illustrated in Fig. 4. Notice that (φ_{k_l}, f_{k_l}) cannot fall in regions I and V since, in this case, (φ_{k_j}, f_{k_j}) will not be excluded from F . It can be easily verified that the worst case occurs when (φ_{k_l}, f_{k_l}) lies on $p_1(\varphi)$ or $p_2(\varphi)$.

Suppose (φ_{k_l}, f_{k_l}) lies on $p_2(\varphi)$, as depicted in Fig. 4. In this case, the rectangle r_m will be entirely above \bar{p}_2 , the line that connects $(\bar{\varphi}_{k_l}, \bar{f}_{k_l})$ to $(\bar{\varphi}_{m+1}, \bar{f}_{m+1})$. Since \bar{p}_2 will be included in the new piecewise linear function $\mathcal{P}(\mathcal{F})$, no point within r_m can ever be reached by a new iterate.

The same idea can be applied in the case (φ_{k_l}, f_{k_l}) lies on $p_1(\varphi)$. Therefore, once (φ_{k_j}, f_{k_j}) is included in F , r_m will always be above $\mathcal{P}(\mathcal{F})$. Since the area of this rectangle is at least $\gamma_f^2 \epsilon^2$ and the set A_0 is completely covered by at most $\text{Surf}(A_0)/[\gamma_f^2 \epsilon^2]$ of such rectangles, it is impossible for an infinite subsequence of $\{k_i\}$ to satisfy (27), and the conclusion follows. \square

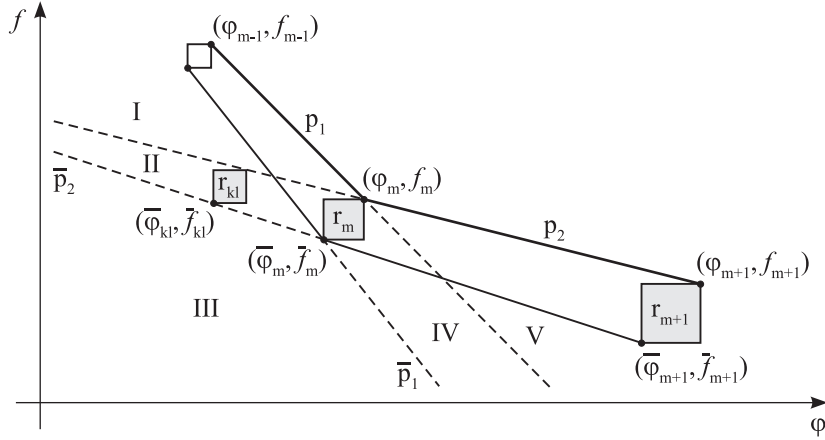


Figure 4: Adding a new iterate that excludes (φ_{k_j}, f_{k_j}) from F .

Finally, we are going to consider the case where no point is added to F_k for k sufficiently large.

Lemma 4 *Suppose that assumptions A1 and A3 hold. Suppose also that, for all $k > k_0$, x_k is never included in F_k . Then,*

$$\lim_{k \rightarrow \infty} \varphi(x_k) = 0. \quad (28)$$

Proof: Since x_k is not included in F_k , no restorations are made and both conditions stated at step 1.8.2 of algorithm 1 are never satisfied for $k > k_0$. Therefore, we have

$$f(x_k) - f(x_{k+1}) \geq \gamma_r P_{red}^{opt} \geq \gamma_r \kappa \varphi(x_k) > 0, \quad (29)$$

for all $k > k_0$, which means that the objective function always decrease between infeasible iterations. Since A1 and A3 imply $f^{min} \leq f(x_k) \leq f^{max}$, we must have

$$\lim_{k \rightarrow \infty} f(x_k) - f(x_{k+1}) = 0. \quad (30)$$

Then, (28) follows from (29) and (30). \square

6 The algorithm finds a critical point

Finally, we are able to prove the convergence of the algorithm to a stationary point for (1). In order to do that, we will need to make one additional assumption on the choice of the normal step s_n .

A4. There exist $\kappa_n, k_N > 0$ such that, if $\{x_{k_i}\}$ is a subsequence of iterates that converges to a feasible point, the choice of s_n at step 1.3.3 of algorithm 1 satisfies

$$\|s_n(x_k, \Delta_k)\| \leq \kappa_n \|C(x_k)\|_2$$

for $k_i > k_N$.

This requirement is also easy to fulfill since it only applies when the infeasibility is small and, in this case, it is reasonable to suppose that the normal step will also be small.

In the following lemma, derived from lemma 6.1 of [10], we show that in the neighborhood of a feasible, regular and non-stationary point, the directional derivative of the quadratic model (8) along d_t is bounded away from zero.

Lemma 5 *Suppose that A2 and A4 hold and that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1). Then there exists $k_1, c_1 > 0$ such that*

$$-\nabla Q(s_n(x, \Delta))^T d_t(H, x, \Delta) \geq c_1 \quad (31)$$

for all $x \in \{x_{k_i} \mid k \geq k_1\}$. Moreover, $\|d_t(H, x, \Delta)\|$ is bounded and bounded away from zero for all $x \in \{x_{k_i} \mid k \geq k_1\}$.

Proof: For all $x \in \{x_{k_i}\}$, we have that

$$d_t(H, x, \Delta) = P_x(-\gamma_t \nabla Q(s_n(x, \Delta))) = P_x(-\gamma_t [H s_n(x, \Delta) + \nabla f(x)]).$$

By the contractive property of the orthogonal projections,

$$\|P_x(-\gamma_t [H s_n(x, \Delta) + \nabla f(x)]) - P_x(-\gamma_t \nabla f(x))\|_2 \leq \gamma_t \|H\|_2 \|s_n(x, \Delta)\|_2.$$

So, by A2 and A4, we have that

$$\|d_t(H, x, \Delta) - P_x(-\gamma_t \nabla f(x))\|_2 \leq \gamma_1 \|C(x)\| \quad (32)$$

for $k > k_N$. By the continuity of $\nabla f(x)$ and the fact that $\{x_{k_i}\}$ converges, we deduce that

$$\|\nabla f(x_{k_i})^T P_x(-\gamma_t \nabla f(x_{k_i})) - \nabla f(x_{k_i})^T d_t(H_{k_i}, x_{k_i}, \Delta_{k_i})\|_2 \leq \gamma_2 \|C(x_{k_i})\|. \quad (33)$$

Notice that $P_x(-\gamma_t \nabla f(x_{k_i}))$ is the solution of

$$\begin{aligned} \text{minimize} \quad & \| -\gamma_t \nabla f(x_{k_i}) - z \|_2^2 \\ \text{subject to} \quad & A(x_{k_i})z = 0 \\ & l \leq x_{k_i} + s_n + z \leq u. \end{aligned}$$

Now, define $P_{x^*}(-\gamma_t \nabla f(x^*))$ as the solution of

$$\begin{aligned} \text{minimize} \quad & \| -\gamma_t \nabla f(x^*) - z \|_2^2 \\ \text{subject to} \quad & A(x^*)z = 0 \\ & l \leq x^* + z \leq u. \end{aligned} \quad (34)$$

Since x^* is regular but is not a stationary point for (1), it follows that $z = 0$ is not a solution for (34). So, $P_{x^*}(-\gamma_t \nabla f(x^*)) \neq 0$. Moreover, since $z = 0$ is feasible for (34), we have that

$$\| -\gamma_t \nabla f(x^*) - P_{x^*}(-\gamma_t \nabla f(x^*)) \|_2^2 < \| -\gamma_t \nabla f(x^*) \|_2^2,$$

which implies that $\nabla f(x^*)^T P_{x^*}(-\gamma_t \nabla f(x^*)) < 0$.

Using the fact that $P_x(-\gamma_t \nabla f(x))$ is a continuous function of x and s_n for all regular x (see [10]), we can define $\bar{c}_2, \bar{c}_3, \bar{c}_4 > 0$ and $\bar{k}_2 \in \mathbb{N}$ such that, for all $x \in \{x_{k_i} \mid k \geq \bar{k}_2\}$, we have

$$\bar{c}_2 \leq \|P_x(-\gamma_t \nabla f(x))\| \leq \bar{c}_3 \quad \text{and} \quad \nabla f(x)^T P_x(-\gamma_t \nabla f(x)) \leq -\bar{c}_4. \quad (35)$$

Now, from (32), (33) and (35), the continuity of $C(x)$ and the feasibility of x^* , there exists $\bar{k}_3 \geq \max\{\bar{k}_2, k_N\}$ such that, whenever $x \in \{x_{k_i} \mid k \geq \bar{k}_3\}$,

$$\frac{\bar{c}_2}{2} \leq \|d_t(H, x, \Delta)\| \leq 2\bar{c}_3 \quad \text{and} \quad \nabla f(x)^T d_t(H, x, \Delta) \leq \frac{-\bar{c}_4}{2}.$$

Therefore, $\|d_t(H, x, \Delta)\|$ is bounded and bounded away from zero for all $x \in \{x_{k_i} \mid k \geq \bar{k}_3\}$.

Finally, since $d_t \in \mathcal{N}(A(x))$, assumptions A2 and A4 hold, and $\|d_t\|$ is bounded, we have that, for all $x \in \{x_{k_i} \mid k \geq \bar{k}_3\}$,

$$\nabla Q(s_n)^T d_t = \nabla f(x)^T d_t + d_t^T H s_n \leq -\frac{\bar{c}_4}{2} + \gamma_3 \|C(x)\|,$$

where $\gamma_3 > 0$. Then, (31) follows defining $c_1 = \bar{c}_4/4$ and choosing $k_1 > \bar{k}_3$ such that $\|C(x)\| \leq \bar{c}_4/(4\gamma_3)$. \square

Using Lemma 5, we state in the next lemma that, in the neighborhood of a feasible, regular and non-stationary point, the decrease of the quadratic model (8) is proportional to the trust region radius Δ .

Lemma 6 *Suppose that A2 and A4 hold and that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1). Then there exists $c_2, k_2 > 0$ and $\Delta_1 \in (0, \Delta_{min})$ such that*

$$Q(x, s_n(x, \Delta)) - Q(x, s_c) \geq c_2 \min\{\Delta, \Delta_1\} \quad (36)$$

for all $x \in \{x_{k_i} \mid k \geq k_2\}$.

Proof: See Lemma 6.2 of [10]. \square

Now, we are able to present a crucial lemma, derived from Lemma 6.3 of [10], that relates P_{red}^{opt} to the trust region radius in the neighborhood of a feasible point.

Lemma 7 *Suppose that A1, A2 and A4 hold and that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1). Then there exists $\epsilon, c_3, k_3 > 0$ and $\Delta_1 \in (0, \Delta_{min})$ such that, for $k_i > k_3$, if*

$$\varphi(x_{k_i}) \leq \epsilon \Delta^2, \quad (37)$$

we have that

$$P_{red}^{opt}(x_{k_i}, s_c) = Q(x_{k_i}, 0) - Q(x_{k_i}, s_c) \geq c_3 \min\{\Delta, \Delta_1\}. \quad (38)$$

Proof: By Lemma 6, assumptions A1 and A4 and the convergence of $\{x_{k_i}\}$, we have that

$$Q(0) - Q(s_c) \geq Q(s_n) - Q(s_c) - |Q(0) - Q(s_n)| \geq c_2 \min\{\Delta, \Delta_1\} - \gamma_4 \|C(x)\|$$

for all $x \in \{x_{k_i} \mid k \geq k_2\}$, where c_2, k_2 and Δ_1 are defined as in Lemma 6 and $\gamma_4 > 0$. Therefore, (38) follows if we choose $c_3 < c_2$ and $k_3 \geq k_2$ such that $\epsilon \leq \alpha^2(c_2 - c_3)^2/(2\gamma_4^2)$, where $\alpha = \min\{1, \Delta_1/\Delta\}$. \square

We next examine what happens if Δ is bounded away from zero and an infinite subsequence of points is added to F .

Lemma 8 *Suppose that A1, A2, A3 and A4 hold and that $\{x_{k_j}\}$ is an infinite subsequence at which x_{k_j} is added to F . Suppose furthermore that the limit points of this sequence are feasible and regular, that the restoration always terminates successfully and that $\Delta_{k_j} \geq \Delta_2$, where Δ_2 is a positive scalar. Then there exists a limit point of this sequence that is a feasible, regular and stationary point for (1).*

Proof: From assumption A3, we know that $\{x_{k_j}\}$ has a convergent subsequence, say $\{x_{k_i}\}$. Let us suppose that the limit point of this subsequence is not stationary for (1).

From Lemma 3 we know that there exists $\bar{k}_5 \in \mathcal{N}$ such that, for $k_i > \bar{k}_5$,

$$\varphi(x_{k_i}) < \epsilon_h \Delta_2^2.$$

Thus, a restoration is never called for $k_i > \bar{k}_5$. So, the hypothesis that x_{k_i} is added to F_{k_i} implies that $A_{red}(x_{k_i}, s_c, \theta_{k_i}^{sup}) \geq \gamma_g P_{red}(x_{k_i}, s_c, \theta_{k_i}^{sup})$ for some $\theta_{k_i}^{sup} \geq \gamma_m$, and that one of the inequalities stated at step 1.8.2 of the algorithm is satisfied at iteration k_i .

Suppose, for the purpose of obtaining a contradiction, that $\{x_{k_i}\}$ converges to a point that is not stationary for (1). So, from Lemma 3 and (38), there exists $\bar{k}_6 \geq \bar{k}_5$ such that $\varphi(x_{k_i}) < \epsilon \Delta_{k_i}^2$ and

$$P_{red}^{opt}(x_{k_i}, s_c) \geq c_3 \min\{\Delta_1, \Delta_2\},$$

for all $k_i > \bar{k}_6$.

Using Lemma 3 again, we can deduce that there exists $\bar{k}_7 \geq \bar{k}_6$ such that $\varphi(x_{k_i}) < (c_3/\kappa) \min\{\Delta_1, \Delta_2\}$ and the condition $P_{red}^{opt} < \kappa \varphi(x_k)$ is never satisfied for $k_i > \bar{k}_7$.

Therefore, $\rho_k < \gamma_r$ must hold. To show that this is not possible, let us write the inequality $A_{red}(x_{k_i}, s_c, \theta_{k_i}^{sup}) \geq \gamma_g P_{red}(x_{k_i}, s_c, \theta_{k_i}^{sup})$, as

$$\begin{aligned} \theta_{k_i}^{sup} A_{red}^{opt}(x_{k_i}, s_c) + (1 - \theta_{k_i}^{sup})(\varphi(x_{k_i}) - \varphi(x_{k_i} + s_c)) &\geq \\ \gamma_g \theta_{k_i}^{sup} P_{red}^{opt}(x_{k_i}, s_c) + \gamma_g (1 - \theta_{k_i}^{sup}) P_{red}^{fsb}(x_{k_i}, s_c). \end{aligned}$$

Using the hypothesis that $\rho_k < \gamma_r$ and the fact that $P_{red}^{fsb}(x_{k_i}, s_c) \geq 0$, we have

$$\theta_{k_i}^{sup} \gamma_r P_{red}^{opt}(x_{k_i}, s_c) + (1 - \theta_{k_i}^{sup})(\varphi(x_k) - \varphi(x_k + s_c)) \geq \gamma_g \theta_{k_i}^{sup} P_{red}^{opt}(x_{k_i}, s_c).$$

Then, taking $k_4 > k_3$ (defined in Lemma 7), we deduce from (38) that, for $k_i > k_4$,

$$(1 - \theta_{k_i}^{sup})(\varphi(x_{k_i}) - \varphi(x_{k_i} + s_c)) \geq (\gamma_g - \gamma_r) \theta_{k_i}^{sup} c_3 \min\{\Delta_1, \Delta_2\}.$$

But, since, $\gamma_g > \gamma_r$ and $\lim_{i \rightarrow \infty} \varphi_{k_i} = 0$, we must have

$$\lim_{i \rightarrow \infty} \theta_{k_i}^{sup} = 0,$$

which contradicts the fact that $\theta_{k_i}^{sup} \geq \gamma_m$. Therefore, $\{x_{k_i}\}$ must converge to a stationary point for (1). \square

Supposing again that Δ is bounded away from zero, we will now complete our analysis investigating what happens when no iterates are added to F for k sufficiently large.

Lemma 9 *Suppose that A1, A2, A3 and A4 hold, that x_k is always accepted but F_k remains unchanged for $k > k_5$ and that $\Delta_k \geq \Delta_3$, for some positive Δ_3 . Suppose also that the limit points of the infinite sequence $\{x_k\}$ are feasible and regular. Then there exists a limit point of $\{x_k\}$ that is a feasible, regular and stationary point of (1).*

Proof: Assumption A3 implies that there exists a convergent subsequence $\{x_{k_i}\}$. If the limit point of this subsequence is not stationary for (1), then from Lemma 7, we have

$$P_{red}^{opt} \geq c_3 \min\{\Delta_3, \Delta_1\} \tag{39}$$

for all $k_i > \max\{k_5, k_3\}$. Moreover, since x_{k_i} is always accepted and F_k is not changed, we deduce, from step 1.8.2 of the algorithm, that $\rho_k \geq \gamma_r$. Therefore,

$$f(x_{k_i}) - f(x_{k_i} + s_c) \geq \gamma_r P_{red}^{opt}. \tag{40}$$

From (39) and (40) we conclude that $f(x_{k_i}) - f(x_{k_i} + s_c) \geq \gamma_r c_3 \min\{\Delta_1, \Delta_3\}$ for all k_i sufficiently large, which contradicts the compactness assumption A3. Thus, the limit point of $\{x_{k_i}\}$ must be stationary. \square

In the last part of this section, we will discuss the behavior of the algorithm when $\Delta \rightarrow 0$. We will start showing that the predicted reduction of the quadratic model is sufficiently large when Δ is small.

Lemma 10 *Suppose that A2 and A4 hold and that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1). Suppose also that φ_k satisfies (37) and that*

$$\Delta < \min\{c_3/(\kappa\epsilon), \Delta_1\} = \Delta_5 \quad (41)$$

for $k_i > k_6$, where c_3 , ϵ and Δ_1 are defined as in Lemma 7. Then $P_{red}^{opt} > \kappa\varphi(x_{k_i})$.

Proof: Suppose, for the purpose of obtaining a contradiction, that $P_{red}^{opt} \leq \kappa\varphi(x_k)$ for some $k_i > k_6$. Then, from (38), we have

$$c_3 \min\{\Delta, \Delta_1\} \leq P_{red}^{opt} \leq \kappa\varphi(x_{k_i}) \leq \kappa\epsilon\Delta^2,$$

which is impossible because of (41). Thus $P_{red}^{opt} > \kappa\varphi(x_{k_i})$ must hold. \square

The purpose of the next four Lemmas is to prove that there exists a sufficiently small trust region radius so the step is always accepted and Δ is not reduced further at step 1.7.1 of algorithm 1.

The first lemma shows the relation between the predicted reduction of the infeasibility and Δ .

Lemma 11 *Suppose that assumption A1 holds and that x_k is not φ -stationary. Then there exists $\Delta_6, c_4 > 0$ such that*

$$P_{red}^{fsb}(x_k, s_c) \geq c_4\Delta_k, \quad (42)$$

if $\Delta_k \in (0, \Delta_6)$.

Proof: The proof of this Lemma is based on the same arguments used to obtain (21). However, here we deal with the reduction of the infeasibility, instead of the reduction of the objective function, so some modifications need to be done.

Firstly, we must notice that $d_n \neq 0$, since we suppose that x_k is not φ -stationary. Thus, we can redefine (20) as

$$t(\Delta) = \max\{t > 0 \mid [x_k, x_k + td_n] \in \Omega \text{ and } \|td_n\| \leq \beta_d\Delta\}.$$

Now, using the fact that for Δ sufficiently small we have $\|t(\Delta)d_n\| = \beta_d\Delta$ and defining

$$c = -\frac{1}{2}d_n^T \nabla \varphi(x_k) / \|d_n\| > 0,$$

there must exist $\Delta_6 \in (0, \|d_n\|]$ such that

$$M(0) - M(t(\Delta)) \geq -\frac{1}{2}d_n^T \nabla \varphi(x_k)t(\Delta) = c\|d_n\|t(\Delta) = \beta_d c \Delta$$

for all $\Delta \in (0, \Delta_6)$. Therefore, for the normal step s_n computed at step 1.3.3 of algorithm 1, we have

$$M(0) - M(s_n) \geq \beta_d \beta_m c \Delta_k.$$

But, since $A(x_k)s_c = A(x_k)s_n$, we deduce from (4) that

$$P_{red}^{fsb}(x_k, s_c) \geq \beta_d \beta_m c \Delta_k$$

and the desired inequality follows. \square

In order to prove that $x_k + s_c$ will be accepted, we need to consider how $\bar{\varphi}$ and \bar{f} are computed. Let us begin using the previous lemma to show that, for a small Δ , $\bar{\varphi}_i$, defined in (10), will depend on the predicted reduction of the infeasibility.

Lemma 12 *Suppose that A1 holds and that x_k is not φ -stationary. Then there exists $\Delta_7 > 0$ such that*

$$\gamma_c P_{red}^{fsb}(x_k, s_c) > \gamma_f \varphi(x_k),$$

if $\varphi(x_k) < \epsilon_h \Delta_k^2$ and $\Delta_k \in (0, \Delta_7)$.

Proof: Lemma 11 ensures that

$$\frac{\gamma_c}{\gamma_f} P_{red}^{fsb}(x_k, s_c) \geq \frac{\gamma_c}{\gamma_f} c_4 \Delta_k > 0.$$

Defining $\Delta_7 = \min\{\gamma_c c_4 / (\gamma_f \epsilon_h), \Delta_6\}$, where Δ_6 is given in Lemma 11, we have that

$$\frac{\gamma_c}{\gamma_f} P_{red}^{fsb}(x_k, s_c) \geq \epsilon_h \Delta_7 \Delta > \epsilon_h \Delta^2 \geq \varphi(x_k),$$

for all $\Delta \in (0, \Delta_7)$, so the desired result follows. \square

Using Lemma 11 again, we can also show that \bar{f} , defined in (11), will depend on P_{red}^{opt} if Δ is sufficiently small.

Lemma 13 *Suppose that A1, A2 and A4 hold, that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1), and that $\bar{\varphi}_k$ is given by (10). Then there exists $\Delta_8 > 0$ such that*

$$\gamma_f P_{red}^{opt}(H_k, x_k, s_c) \geq (\varphi(x_k) - \bar{\varphi}_k),$$

if $\varphi(x_k) < \min\{\epsilon_h, \epsilon\} \Delta_k^2$ and $\Delta_k \in (0, \Delta_8)$, where ϵ is defined as in Lemma 7.

Proof: From Lemma 7 we deduce that, if $\Delta_k \in (0, \Delta_1]$, then

$$\gamma_f P_{red}^{opt}(H_k, x_k, s_c) \geq \gamma_f c_3 \Delta_k.$$

Now, defining $\Delta_8 = \min\{\gamma_f c_3 / \epsilon_h, \Delta_1\}$, we have

$$\gamma_f P_{red}^{opt}(H_k, x_k, s_c) \geq \epsilon_h \Delta_8 \Delta \geq \epsilon_h \Delta_k^2 \geq \varphi(x_k) \geq (\varphi(x_k) - \bar{\varphi}_k)$$

and the desired conclusion follows. \square

Let us prove now that, for a infeasible x , the ratio between the actual and the predicted reduction of the merit function is sufficiently large if Δ is small, not matter how the penalty parameter θ is chosen.

Lemma 14 *Suppose that A1, A2 and A4 hold, that $\theta \in [0, 1]$ and $\gamma \in (0, 1)$ are given parameters and that $\{x_{k_i}\}$ is an infinite subsequence that converges to the feasible and regular point $x^* \in \Omega$, which is not stationary for (1). Then there exists $\Delta_9 > 0$ such that, for k_i sufficiently large,*

$$A_{red}(x_{k_i}, s_c, \theta) \geq \gamma P_{red}(H_{k_i}, x_{k_i}, s_c, \theta)$$

for all $\Delta_{k_i} \in (0, \Delta_9)$, if x_{k_i} is infeasible.

Proof: Since $\lim_{i \rightarrow \infty} \varphi(x_{k_i}) = 0$ and, at the beginning of iteration k , the trust region radius satisfies $\Delta_{k_i} \geq \Delta_{min}$, there must exist $k_7 \geq k_3$ (defined in Lemma 7) such that, for $k_i > k_7$, the condition $\varphi(x_{k_i}) \leq \epsilon \Delta_{k_i}^2$ is satisfied, so (38) holds. Besides, since x_{k_i} is infeasible, (42) also holds if we take $\Delta_9 \leq \Delta_6$ (defined in Lemma 11). Therefore, from the definition of P_{red} , we have that

$$P_{red}(H_{k_i}, x_{k_i}, s_c, \theta) \geq \theta c_3 \min\{\Delta_{k_i}, \Delta_1\} + (1 - \theta) c_4 \Delta_{k_i}.$$

where Δ_1 , c_3 and c_4 are defined as in Lemmas 7 and Lemma 11.

For any $\theta \in [0, 1]$, the above inequality implies that

$$P_{red}(H_{k_i}, x_{k_i}, s_c, \theta) \geq \min\{\Delta_{k_i}, \Delta_1\} \min\{c_3, c_4\} > 0.$$

But, from A1 and A4, we also have that

$$|A_{red}(\Delta_{k_i}) - P_{red}(\Delta_{k_i})| \leq c_5 \Delta_{k_i}^2.$$

for some $c_5 > 0$. From the last two inequalities, we deduce that, for $\Delta_{k_i} \leq \Delta_1$,

$$\frac{|A_{red}(\Delta_{k_i}) - P_{red}(\Delta_{k_i})|}{P_{red}(\Delta_{k_i})} = \left| \frac{A_{red}(\Delta_{k_i})}{P_{red}(\Delta_{k_i})} - 1 \right| \leq \frac{c_5}{\min\{c_3, c_4\}} \Delta_k. \quad (43)$$

Therefore, defining $\Delta_9 = \min\{(1 - \gamma) \min\{c_3, c_4\}/c_5, \Delta_1, \Delta_6\}$, we obtain the required result. \square

In our last lemma, we will use the previous results to prove that, if $\Delta \rightarrow 0$, there is no infinite subsequence that converges to a point that is not stationary for (1).

Lemma 15 *Suppose that A1, A2, A3 and A4 hold. Suppose also that the limit points of the infinite sequence $\{x_k\}$ are feasible and regular and that $\lim_{k \rightarrow \infty} \Delta_k = 0$. Then there exists a limit point of $\{x_k\}$ that is a stationary point of (1).*

Proof: Assumption A3 implies that there exists a convergent subsequence $\{x_{k_i}\}$. Let us suppose, for the purpose of obtaining a contradiction, that the limit point of this subsequence is not stationary for (1).

Now, we need to consider separately two mutually exclusive situations. First, let us suppose that x_{k_i} is feasible. In this case, Lemma 2 assures that, for $\Delta < \bar{\Delta}_4$ (defined in (26)), the step is accepted and the trust region radius need not to be reduced further.

On the other hand, if x_{k_i} is not φ -stationary, Lemmas 12 and 13 assure that, for $\Delta_{k_i} \leq \min\{\Delta_7, \Delta_8\} = \Delta_{10}$, we have $P_{red}^{fsb} \geq \gamma_f \varphi_i / \gamma_c$ and $P_{red}^{opt} \geq (\varphi_i - \bar{\varphi}_i) / \gamma_f$, so the parameter $\bar{\eta}$ used in (12) is given by

$$\bar{\eta} = \frac{(\bar{f}_{i-1} - \bar{f}_i) \gamma_c P_{red}^{fsb} + (\bar{\varphi}_i - \bar{\varphi}_{i-1}) \gamma_f P_{red}^{opt}}{(\bar{f}_{i-1} - \bar{f}_i) P_{red}^{fsb} + (\bar{\varphi}_i - \bar{\varphi}_{i-1}) P_{red}^{opt}},$$

where i is defined in such a manner that $\bar{\varphi}_{i-1} \leq \varphi(x_+) \leq \bar{\varphi}_i$. Thus, $0 < \bar{\eta} < 1$ and $f(x_{k_i} + s_c) > \mathcal{P}(\mathcal{F}_{k_i}, \varphi(x_{k_i} + s_c))$ is equivalent to (12).

Now, using Lemmas 10 and 14, we can deduce that, for $\Delta_{k_i} < \Delta_9$, the step is always accepted. Consequently, $\Delta_{k_i} \geq \alpha_R \min\{\bar{\Delta}_4, \Delta_9, \Delta_{10}\}$, which contradicts the hypothesis that $\lim_{k \rightarrow \infty} \Delta_k = 0$, so we conclude that the limit point of the subsequence $\{x_{k_i}\}$ is a stationary point of (1). \square

Finally, let us state a theorem that puts together all of the results presented so far.

Theorem 1 *Suppose that A1, A2, A3 and A4 hold and that $\{x_k\}$ is an infinite sequence generated by Algorithm 1. Then either the restoration converges to a φ -stationary but infeasible point of (1), or $\lim_{k \rightarrow \infty} \varphi(x_k) = 0$. Moreover, if the restoration always succeeds and all of the limit points of $\{x_k\}$ are regular, there exists a limit point x^* that is a regular and stationary point for (1). In particular, if all of the φ -stationary points are feasible and regular, then there exists a subsequence of $\{x_k\}$ that converges to a feasible, regular and stationary point of (1).*

Proof: This result is a direct consequence of Lemmas 3, 4, 8, 9 and 15. \square

7 Numerical experience

The description of some steps of Algorithm 1 was intentionally left vague to suggest the reader that several alternative implementations are available. In this section, we describe one of such possible implementations and present the numerical results obtained by applying the algorithm to some problems from the CUTER collection [13].

7.1 Algorithmic details

The computational effort of algorithm 1 may be decomposed into three main parts. The first is related to the reduction of the infeasibility and includes steps 1.3.1 to 1.3.3. The aim of the second part, that comprises steps 1.4 to 1.6, is to improve the objective function. Finally, a restoration is called, at step 1.9.1, if the infeasibility needs to be drastically reduced. Each one of these parts is briefly described below.

Taking a closer look at steps 1.3.1 and 1.3.2, one may notice that vectors d_n and s_n^{dec} can be easily determined, since the first involves computing a projection on a box and the second requires only the solution of a one-dimensional problem.

The normal step, s_n , can be obtained by solving the bound-constrained least squares problem (18), replacing the word *reduce* by *minimize*. In our experiments, the Quacon package [2] was used for this purpose. The computation of s_n is declared successful when both conditions

$$M(x_k, 0) - M(x_k, s_n) \geq \beta_m [M(x_k, 0) - M(x_k, s_n^{dec})]$$

and

$$\|g_p(s_n)\|_2 \leq 0.001 \|g_p(s_n^{dec})\|_2$$

are satisfied, where $\beta_m = 0.9$ and g_p is the projected gradient of the quadratic function minimized. Otherwise, the limit of $\max\{1000, 6n\}$ iterations is reached.

Vector s_c is computed by applying the MINOS 5.4 [19] solver to the quadratic problem (19), stopping the algorithm when both

$$\|g_p(s_c)\|_\infty \leq 10^{-6} \left\{ \frac{\|\lambda\|_1}{\sqrt{m}}, 1 \right\}, \quad (44)$$

and condition (36) are satisfied, where, again, g_p is the projected gradient and λ is the vector of Lagrange multipliers. A limit of $\max\{1000, 3n\}$ iterations also applies.

Vector d_t is obtained, at step 1.4, using the MINOS package to approximately solve the problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|d + \gamma_t \nabla Q(s_n)\|_2^2 \\ \text{subject to} \quad & A(x_k)d = 0 \\ & l \leq x_k + s_n + d \leq u, \end{aligned}$$

where $\gamma_t = 0.001$. Since d_t is much more difficult to obtain than d_n , we decided to compute this vector only if s_c fails to satisfy conditions (36) and (44). In this case, after determining d_n and s_t^{dec} , s_c is computed again.

To determine the restored point s_r , successive normal steps were computed until the conditions stated at step 1.9.1 were satisfied. A more sophisticated restoration procedure could be devised, but this very simple scheme seemed to be satisfactory.

In all of the experiments, exact Hessians were computed, using the Lagrange multipliers supplied by MINOS as the approximate dual variables for problem (1).

Algorithm 1 terminates when $\|C(x_k)\|_\infty \leq 10^{-6}$ and one of the conditions

1. $\|\nabla \ell(x_k, \lambda_k)\|_\infty \leq \epsilon_g \max\{1, \|\nabla \ell(x_{typ}, \lambda_{typ})\|_\infty\}$,
2. $\|s_c\|_\infty \leq \epsilon_x \max\{1, \|x_k\|_\infty\}$, or
3. $P_{red}^{fsb} \leq \epsilon_s$ and $P_{red}^{opt} \leq \epsilon_o$ (for three successive iterations),

is satisfied, where $\epsilon_g = 10^{-6}$, $\epsilon_x = 10^{-18}$, $\epsilon_s = \epsilon_o = 10^{-8}$, and x_{typ} and λ_{typ} are typical primal and dual steps, determined at the first iterations of the algorithm.

The algorithm also stops if, after a restoration step, $\nabla \varphi(x_k + s) \leq 10^{-10}$. In this case, it is more than likely that a φ -stationary but infeasible point was found.

The remaining parameters used are $\Delta_0 = 10^5$, $\Delta_{min} = 10^{-5}$, $\gamma_n = 1.0$, $\gamma_t = 10^{-8}$, $\gamma_r = 10^{-2}$, $\gamma_g = 5 \cdot 10^{-2}$, $\kappa = 10^{-4}$, $\eta = 0.9$, $\alpha_A = 2.5$, $\alpha_R = 0.25$, $\beta_d = 0.8$, $\beta_q = 0.9$, $\beta_r = 16.0$, $\gamma_s = 10^{-4}$, $\gamma_f = 10^{-6}$, $\gamma_c = 10^{-3}$, $\gamma_m = 10^{-4}$, $\Delta_{rest} = 10^{-1}$ and $\epsilon_h = 10^2$.

7.2 Algorithm performance

To analyze the behavior of the algorithm just described, a set of 82 small to medium-size problems was extracted from the CUTER collection [13]. The selected problems are presented in Tables 1, 2 and 3. The number of variables of the original CUTER problem is given by n , while s is the number of slack variables required to convert the problem to the form shown in (1). The number of constraints is given by m .

Table 1: Selected tiny problems from the CUTER collection.

| Problem | n | s | m | Problem | n | s | m |
|----------|-----|----|----|----------|----|----|----|
| AIRPORT | 84 | 42 | 42 | HS116 | 13 | 14 | 14 |
| AVION2 | 49 | | 15 | LAKES | 90 | | 78 |
| CORE1 | 65 | 18 | 59 | LAUNCH | 25 | 19 | 28 |
| DEMBO7 | 16 | 20 | 20 | MINMAXBD | 5 | 20 | 20 |
| DIXCHLNG | 10 | | 5 | MRIBASIS | 36 | 46 | 55 |
| DIXCHLNV | 100 | | 50 | PRODPL1 | 60 | 9 | 29 |
| ERRINBAR | 18 | 1 | 9 | SWOPF | 83 | 14 | 92 |
| HIMMELBI | 100 | | 12 | TENBARS3 | 18 | | 8 |
| HIMMELBK | 24 | | 14 | TRUSPYR2 | 11 | 8 | 11 |

Table 2: Selected small problems from the CUTER collection.

| Problem | n | s | m | Problem | n | s | m |
|----------|-----|-----|-----|----------|-----|-----|-----|
| BRITGAS | 450 | | 360 | LUBRIF | 751 | | 500 |
| CATENARY | 501 | | 166 | MADSSCHJ | 201 | 398 | 398 |
| DALLASL | 906 | | 667 | MINC44 | 583 | | 519 |
| EIGENACO | 110 | | 55 | NET3 | 464 | 127 | 521 |
| EIGMINB | 101 | | 101 | QPCBLEND | 83 | 31 | 74 |
| ELATTAR | 7 | 102 | 109 | SMBANK | 117 | | 64 |
| EXPFITC | 5 | 502 | 502 | SMMPFS | 720 | 23 | 263 |
| HAIFAM | 99 | 150 | 150 | SSEBNLN | 194 | 24 | 96 |
| HALDMADS | 6 | 42 | 42 | STEENBRC | 540 | | 126 |
| HET-Z | 2 | 202 | 202 | YORKNET | 312 | | 256 |
| LEAKNET | 156 | | 153 | | | | |

The algorithm, hereafter called GMM99, was implemented in FORTRAN 77 and the executable program was generated using the ifort 8.1 compiler, under the Fedora 3 Linux operating system. To evaluate its performance, the algorithm was compared to Lancelot (release B), the well known nonlinear programming package distributed along with the Galahad library [14]. The tests were performed on a Dell Optiplex GX280 computer, using an Intel Pentium 4 540 processor, with a clock speed of 3.2GHz, 1MB of cache memory, a 800MHz front side bus and the Intel 915G chipset. The Lancelot default parameters were used, except for the maximum number of iterations that was increased to 10000. Exact first and second derivatives were computed by both methods.

The new algorithm failed to obtain a feasible point for problems LUBRIF, ROTDISC and YORKNET. Besides, it also could not find an objective function as good as Lancelot for problems BRAINPC5, CATENARY, DIXCHLNG, DTOC2, EG3, ELATTAR, NGONE and OPTCTRL3. These 11 occurrences were classified as *GMM99 failures*.

Table 3: Selected medium-size problems from the CUTEr collection.

| Problem | n | s | m | Problem | n | s | m |
|----------|------|------|------|----------|------|------|------|
| AUG2D | 3280 | | 1600 | HYDROELL | 1009 | 1008 | 1008 |
| BIGBANK | 2230 | | 1112 | JUNKTURN | 5010 | | 3500 |
| BRAINPC5 | 6907 | | 6900 | MINPERM | 1113 | | 1033 |
| BRIDGEND | 2734 | | 2727 | MOSARQP1 | 2500 | 700 | 700 |
| CLNLBEAM | 3003 | | 2000 | NGONE | 100 | 1273 | 1273 |
| CORKSCRW | 906 | 100 | 700 | OET2 | 3 | 1002 | 1002 |
| COSHFUN | 6001 | 2000 | 2000 | OPTCDEG3 | 1202 | | 800 |
| CRESC132 | 6 | 2654 | 2654 | OPTCTRL3 | 4502 | | 3000 |
| DITTERT | 1133 | | 1034 | OPTMASS | 3010 | 501 | 2505 |
| DRUGDIS | 3004 | | 2000 | ORTHREGD | 1003 | | 500 |
| DTOC2 | 5998 | | 3996 | READING1 | 4002 | | 2000 |
| DTOC4 | 2999 | | 1998 | READING3 | 4002 | | 2001 |
| EG3 | 1001 | 1999 | 2000 | ROTDISC | 905 | 361 | 1081 |
| GAUSSELM | 1240 | 2002 | 3017 | SINROSNB | 1000 | 999 | 999 |
| GILBERT | 1000 | | 1 | SSNLBEAM | 3003 | | 2000 |
| GRIDNETD | 3444 | | 1764 | SVANBERG | 1000 | 1000 | 1000 |
| HADAMARD | 401 | 800 | 1010 | TFI1 | 3 | 1001 | 1001 |
| HAGER1 | 2001 | | 1000 | TRAINF | 4008 | | 2002 |
| HAGER2 | 2001 | | 1000 | UBH1 | 9009 | | 6000 |
| HANGING | 1800 | 1150 | 1150 | ZAMB2 | 3966 | | 1440 |
| HELSEBY | 1408 | | 1399 | ZIGZAG | 3004 | 500 | 2500 |
| HVYCRASH | 4004 | | 3000 | | | | |

Lancelot attained the limit of 10000 iterations without finding a feasible point for problems CRESC132, DITTERT, LAUNCH, NET3, OPTMASS and SMMPSF. For problems DRUGDIS, GAUSSELM, HALDMADS, HVYCRASH, READING3 and SINROSNB, the new algorithm found a better objective function. Finally, for problems BRIDGEND and ROTDISC Lancelot exceeded the limit of 3000 seconds of running time. These 14 cases were treated as *Lancelot failures*.

Table 4 shows a summary of the results obtained comparing the new algorithm to Lancelot. According to this table, GMM99 outperformed Lancelot in 47 problems (out of 82), while Lancelot achieved the best performance in 32 cases (including GMM99 failures). As mentioned above, both algorithms failed to solve problem ROTDISC.

Table 4: Comparison between GMM99 and Lancelot.

| Compared performance | GMM99 | Lancelot |
|-----------------------------|-------|----------|
| Almost indistinguishable | 2 | 2 |
| From 1.01 to 2 times better | 2 | 5 |
| From 2 to 10 times better | 18 | 12 |
| From 10 to 100 times better | 10 | 3 |
| More than 100 times better | 4 | 2 |
| Failures | 11 | 14 |

Table 5 shows the number of times GMM99 and Lancelot obtained the best solution for each type of objective function and constraint. In this table, the problems are also divided according to their size.

For this specific selection of problems, GMM99 performed better for problems with a linear objective function, while Lancelot was the best option for solving problems with a quadratic function and nonlinear constraints. These results are probably related to the fact that MINOS was used to compute the tangential step in GMM99. In general, GMM99 had a slightly better performance.

The new algorithm was also compared with GMM, the standard trust-region SQP method presented in [10]. The results are shown in Tables 6 and 7.

Table 5: Number of times GMM99 and Lancelot outperformed one another as a function of problem type.

| Objective type | Constraint type | Algorithm | Problem size | | | Total |
|----------------|-----------------|-----------|--------------|-------|--------|-------|
| | | | tiny | small | medium | |
| Linear | Nonlinear | Lancelot | 0 | 3 | 2 | 5 |
| | | GMM99 | 9 | 8 | 8 | 25 |
| Quadratic | Linear | Lancelot | 0 | 0 | 3 | 3 |
| | | GMM99 | 0 | 1 | 1 | 2 |
| | Nonlinear | Lancelot | 3 | 2 | 10 | 15 |
| | | GMM99 | 2 | 1 | 3 | 6 |
| Nonlinear | Linear | Lancelot | 0 | 0 | 2 | 2 |
| | | GMM99 | 2 | 3 | 2 | 7 |
| | Nonlinear | Lancelot | 0 | 1 | 6 | 7 |
| | | GMM99 | 1 | 1 | 5 | 7 |

Table 6: Comparison between GMM99 and GMM.

| Compared performance | GMM99 | GMM |
|-----------------------------|-------|-----|
| Almost indistinguishable | 15 | 15 |
| From 1.01 to 2 times better | 18 | 12 |
| From 2 to 10 times better | 9 | 3 |
| From 10 to 100 times better | 3 | 1 |
| Failures | 9 | 15 |

The GMM algorithm was not able to find a feasible solution for problems CATENARY, CRESC132, LAKES, LUBRIF, ROTDISC and STEENBRC. Moreover, the objective function at the solution was worse than the value obtained by GMM99 for problems AUG2D, COSHFUN, EIGMINB, HAGER1, HAGER2, HIMMELBI, NGONE and SSNLBEAM. For the OPTCTRL3 problem, the algorithm found a solution too far from the one supplied by Lancelot. These problems were classified as *GMM failures*.

The 9 GMM99 failures cited in Table 6 include problems LUBRIF, ROTDISC and YORKNET, for which a feasible point was not found, problem OPTCTRL3, because Lancelot obtained a much better solution, and also problems BRAINPC5, ELATTAR, GAUSSELM, HVYCRASH and SINROSNB, for which GMM has found a lower objective function.

Naturally, the definition of failure used here is arguable, since the fact that an algorithm has found a better solution does not imply that the other method failed to find a local optimum. Besides, problems CATENARY, DIXCHLNG, DTOC2, EG3 and NGONE, classified as GMM99 failures when comparing this algorithm to Lancelot, are not considered as failures in Table 6, since GMM was not able to find a better solution in all of these cases. Even though, we decided to use the term *failure* to stress that an algorithm that finds a lower solution performs better than the rival, no matter the time spent to find the solution.

Finally, Table 7 shows how the good performance of GMM and GMM99 is distributed among different types of problems.

Tables 6 and 7 clearly show the superiority of GMM99 over GMM. The new algorithm is more robust and attained the best performance for all of the problem classes. Taking into consideration that GMM also uses Minos to compute the horizontal step and Quacon to compute the vertical step, so the core of the algorithm is the same of GMM99, these results suggest that the use of the new acceptance criterion, combined with a restoration procedure, can improve the performance of the SQP algorithm.

For some problems, no points were included in F , so the behavior of GMM and GMM99 were very similar. Considering the whole set of 82 problems, the mean number of points in F at the last iteration of GMM99 was 5. However, for 7 problems, the cardinality of the F set was greater or equal to 20.

Table 7: Number of times GMM and GMM99 outperformed one another as a function of problem type.

| Objective type | Constraint type | Algorithm | Problem size | | | Total |
|----------------|-----------------|-----------|--------------|-------|--------|-------|
| | | | tiny | small | medium | |
| Linear | Nonlinear | GMM | 2 | 4 | 3 | 9 |
| | | GMM99 | 3 | 5 | 7 | 15 |
| Quadratic | Linear | GMM | 0 | 0 | 0 | 0 |
| | | GMM99 | 0 | 1 | 3 | 4 |
| | Nonlinear | GMM | 0 | 1 | 5 | 6 |
| | | GMM99 | 3 | 0 | 7 | 10 |
| Nonlinear | Linear | GMM | 0 | 1 | 1 | 2 |
| | | GMM99 | 1 | 1 | 2 | 4 |
| | Nonlinear | GMM | 0 | 2 | 3 | 5 |
| | | GMM99 | 1 | 0 | 8 | 9 |

The restoration was called at least once for 24 problems. The purpose of the restoration is to avoid the excessive reduction of the step size when x_k is very infeasible. This strategy has shown to be effective for 18 problems, while the performance was not significantly altered for two problems and slightly worsened for 4 problems.

8 Conclusions

In this paper, we depict the general framework of a new SQP algorithm that combines ideas from both merit functions and the filter introduced by Fletcher and Leyffer in [8]. The use of several penalty parameters defined automatically by the previous iterates avoids the premature reduction of θ as well as the zigzagging that can occur when a non-monotone strategy is used to update this parameter. The new method is also less tolerant than the filter method, since we do not accept points that marginally reduce the infeasibility or the objective function.

The algorithm presented here is unique in the sense that it combines

- The use of a piecewise-linear function to accept or reject iterates. This function can be interpreted as a merit function that selects the penalty parameter according to the position of the predicted point in the (φ, f) -plane, but also as a convex piecewise filter.
- The effective use of the points in F to define the forbidden region in the (φ, f) -plane. In addition to dominated points, filter-SQP methods cut off the entire northwestern part of the allowed region when the predicted reduction of the objective function is positive. The GMM99 algorithm rejects points if they belong to the forbidden region depicted in Figure 2 or if condition (14) is satisfied. However, this second condition is so mild that it is seldom used in practice.
- The use of the restoration to prevent the step shortening inherent to methods that use points in the (φ, f) -plane to define the forbidden region.

The results shown in the previous section suggest that the algorithm, as implemented, is superior to the method presented in [10] and slightly better than Lancelot for small to medium-size problems. As a future work, we intend to test other strategies to compute the restoration, as well as other solvers to compute the normal and tangential steps, in order to deal with larger problems. Perhaps, the library used to solve the quadratic subproblems can be selected dynamically, after knowing the size of the problem and the type of the objective function and constraints.

Acknowledgments

The author would like to thank Prof. José Mario Martínez for his insightful comments and Tadao Ando Jr. for the revision of the text.

References

- [1] C. Audet and J. E. Dennis, *A pattern search filter method for nonlinear programming without derivatives*, SIAM J. Optimization, 14 (2004), pp.980–1010.
- [2] R. H. Bielschowsky, A. Friedlander, F. A. M. Gomes, J. M. Martínez and M. Raydan, *An adaptative algorithm for bound constrained quadratic minimization*, Investigación Operativa, 7 (1997), pp. 67–102.
- [3] R. H. Byrd, J. C. Gilbert and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Programming, 89 (2000), pp. 149–185.
- [4] R. H. Byrd, M. E. Hribar and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optimization, 9 (2000), pp. 877–900.
- [5] C. M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Programming, 96 (2003), pp. 161–177.
- [6] A. Conn, N. I. M. Gould and P. L. Toint, *Trust Region Methods*, SIAM, Philadelphia (2000).
- [7] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM J. Optimization, 13 (2002), pp. 635–659.
- [8] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Programming, 91 (2002), pp. 239–269.
- [9] R. Fletcher, S. Leyffer and P. L. Toint, *On the global convergence of a filter-SQP algorithm*, SIAM J. Optimization, 13 (2002), pp. 44–59.
- [10] F. A. M. Gomes, M. C. Maciel and J. M. Martínez, *Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters*, Math. Programming, 84 (1999), pp. 161–200.
- [11] C. Gonzaga, E. Karas and M. Vanti, *A globally convergent filter method for nonlinear programming*, SIAM J. Optimization, 14 (2003), pp. 646–669.
- [12] N. I. M. Gould, S. Leyffer and P. L. Toint, *A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares*, SIAM J. Optimization, 15 (2005), pp. 17–38.
- [13] N. I. M. Gould, D. Orban and P. L. Toint, *CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
- [14] N. I. M. Gould, D. Orban and P. L. Toint, *GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization*, ACM Trans. Math. Software, 29 (2003), pp. 353–372.
- [15] N. I. M. Gould, C. Sainvitu and P. L. Toint, *A filter-trust-region method for unconstrained optimization*, SIAM J. Optimization, 16 (2006), pp. 341–357.

- [16] N. I. M. Gould and P. L. Toint, *Global convergence of a hybrid trust-region SQP-filter algorithm for general nonlinear programming*, in System Modeling and Optimization XX, E. Sachs and R. Tichatschke (eds.), Kluwer, Dordrecht (2003), pp. 23–54.
- [17] N. I. M. Gould and P. L. Toint, *Global convergence of a non-monotone trust-region filter algorithm for non-linear programming*, Tech. report RAL-TR-2003-003, CLRC Rutherford Appleton Laboratory, Oxfordshire, UK (2003).
- [18] W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, *Lecture Notes in Economics and Mathematical Systems*, 187 (1981), Springer, Heidelberg.
- [19] R. B. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, *Math. Programming*, 14 (1978), pp. 41–72.
- [20] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York (1999).
- [21] T. D. Plantenga, *A trust-region method for nonlinear programming based on primal interior point techniques*, *SIAM J. Sci. Computing*, 20 (1999), pp. 282–305.
- [22] M. Ulbrich, S. Ulbrich and L. N. Vicente, *A globally convergent primal-dual interior point filter method for nonconvex nonlinear programming*, *Math. Programming*, 100 (2004), pp. 379–410.
- [23] S. Ulbrich, *On the superlinear local convergence of a filter-SQP method*, *Math. Programming*, 100 (2004), pp. 217–245.
- [24] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence*, *SIAM J. Optimization*, 16 (2005), pp. 1–31.
- [25] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, *Math. Programming*, 106 (2006), pp. 25–57.