

# A SEQUENTIAL QUADRATIC PROGRAMMING ALGORITHM THAT COMBINES MERIT FUNCTION AND FILTER IDEAS\*

FRANCISCO A. M. GOMES<sup>†</sup>

**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.

**Key words.** nonlinear programming, sequential quadratic programming, merit functions, filter methods.

**AMS subject classifications.** 65K05, 90C55, 90C30, 90C26

**1. Introduction.** In this paper we are concerned with the problem

$$\begin{aligned} (1.1a) \quad & \text{minimize } f(x) \\ (1.1b) \quad & \text{subject to } C(x) = 0 \\ (1.1c) \quad & l \leq x \leq u \end{aligned}$$

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 (1.1) may be slack variables generated when converting inequality constraints to this form.

Algorithms based on the sequential quadratic programming (SQP) approach are one of the most effective methods for solving (1.1). 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*. *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 work was supported by FAPESP grant 2004/05891-1

<sup>†</sup>University of Campinas – IMECC – Department of Applied Mathematics. CP 6065, 13081-970, Campinas, São Paulo, Brazil (chico@ime.unicamp.br).

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 (1.1c), so the merit function chosen was the *augmented Lagrangian*, written here (and in [10]) in an unusual way as

$$(1.2) \quad \mathcal{L}(x, \lambda, \theta) = \theta[f(x) + C(x)^T \lambda] + \frac{(1-\theta)}{2} \|C(x)\|_2^2.$$

In (1.2),  $\theta \in [0, 1]$  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 (1.1). The GMM algorithm approximates (1.1) by the quadratic programming problem

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

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

$$(1.3) \quad P_{red}(H, x, s, \theta) = \theta P_{red}^{opt}(H, x, s) + (1-\theta) P_{red}^{fsb}(x, s),$$

where

$$(1.4) \quad P_{red}^{fsb}(x, s) = M(x, 0) - M(x, s)$$

is the predicted reduction of the infeasibility and

$$(1.5) \quad P_{red}^{opt}(H, x, \lambda, s) = \overline{Q}(H, x, \lambda, 0) - \overline{Q}(H, x, \lambda, s)$$

is the predicted reduction of the Lagrangian.

The penalty parameter  $\theta$  plays a crucial role in the acceptance of the step. For an augmented Lagrangian of the form (1.2),  $(\theta - 1)/\theta$  can be viewed as the slope of the line that defines the *forbidden region* in the  $(\varphi, \ell)$ -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 1.1, where the forbidden region is highlighted for different values of  $\theta$ .

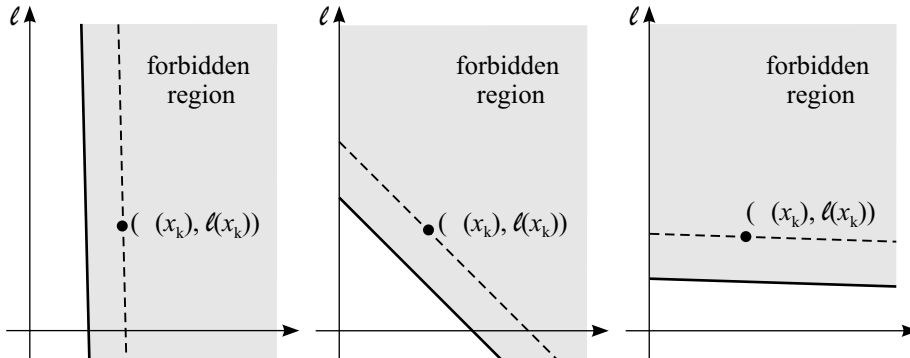


FIG. 1.1. Three merit functions on the  $(\varphi, \ell)$ -plane, showing the influence of the penalty parameter  $\theta$ . On the left,  $\theta = 1/50$ . In the middle,  $\theta = 1/2$ . On the right,  $\theta = 49/50$ .

Merit functions have been criticized for many reasons. First, it is not so easy to choose an initial value for  $\theta$ , since  $\ell(x, \lambda)$  and  $\varphi(x)$  usually have very different meanings and units. Besides, it is necessary to decrease  $\theta$  as the algorithm progresses to force it to find a feasible solution. If the initial penalty parameter used is near to 1 and  $\theta$  is decreased slowly, the algorithm may take too many iterations to reach a feasible point. On the other hand, starting from a small  $\theta$  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  $\theta$  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 the difficulties inherent to merit functions, Fletcher and Leyffer [8] introduced the idea of using a filter. Instead of combining infeasibility and optimality measures 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]).

As an example of this nondominance criterion, in the SQP-filter method presented in [6], a point is accepted whenever it satisfies

$$(1.6) \quad \phi(x) < (1 - \gamma)\phi_j \text{ or } f(x) < f_j - \gamma\phi(x) \text{ for all } (\phi_j, f_j) \in \mathcal{F},$$

where

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

$\mathcal{F}$  is a set of previously generated points in the  $(\phi, f)$ -space and  $\gamma \in (0, 1)$  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 (1.6) 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 (1.6) is that this acceptance criterion is too tolerant. In fact, requiring only the infeasibility *or* the optimality to be improved may allow the acceptance of points that are only marginally less infeasible but are much less optimal (in the sense that  $f(x)$  is greater) than 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  $(\varphi, f)$ -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.1.1, for values of  $\theta$  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), while the algorithm of Gonzaga et al. [11] computes a restoration at every step. In our method, this strategy will be used whenever staying away from feasibility seems not to be worth. 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], feasible points are never added to  $\mathcal{F}$ , so a feasible iterate will always satisfy the first inequality of (1.6). 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  $(f, \varphi)$ -space. These parameters are also automatically computed from some  $(f, \varphi)$ -pairs collected at previous iterations, so no update scheme need to be defined for  $\theta$ .

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  $\ell_2$  merit function*, defined as:

$$(2.1) \quad \psi(x, \theta) = \theta f(x) + (1 - \theta)\varphi(x).$$

The actual reduction of the  $\ell_2$  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 (1.3), replacing (1.5) by

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

where

$$(2.2) \quad Q(H, x, s) = \frac{1}{2}s^T H s + \nabla f(x)^T s + f(x).$$

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 functions, let  $F$  be a set of  $p$  points  $(\varphi_i, f_i)$  in the  $(\varphi, f)$ -plane. Suppose that these pairs are ordered so that  $\varphi_1 < \varphi_2 < \dots < \varphi_p$ . Suppose also that each point  $(\varphi_i, f_i)$  in  $F$  is below the line segment joining  $(\varphi_{i-1}, f_{i-1})$  and  $(\varphi_{i+1}, f_{i+1})$ , 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  $(\varphi_i, f_i)$  in  $F$ , define another point  $(\bar{\varphi}_i, \bar{f}_i)$  by moving a little towards the southwest. 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

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

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

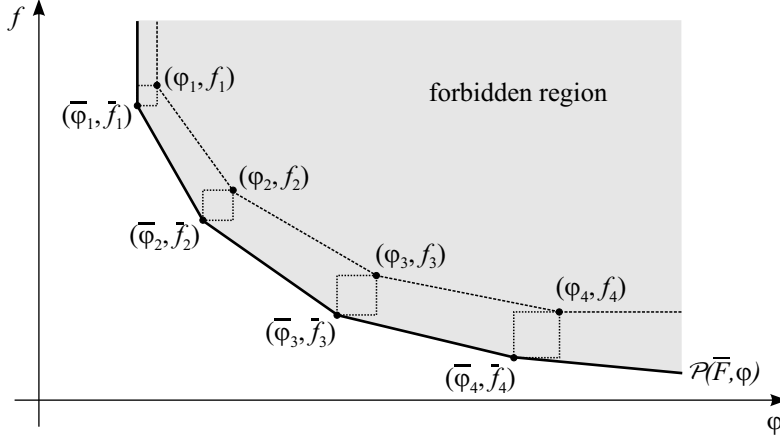


FIG. 2.1. The set  $F$  and the piecewise linear function  $\mathcal{P}(\bar{F}, \varphi)$ .

This new function, illustrated in Fig. 2.1, is formed by  $p + 1$  line segments that can be viewed as merit functions in the form (2.1). The  $i$ -th of these functions is defined by the penalty parameter

$$(2.3) \quad \bar{\theta}_i = \begin{cases} 0, & \text{if } i = 0; \\ \frac{\bar{\varphi}_{i+1} - \bar{\varphi}_i}{f_i - \bar{f}_{i+1} + \bar{\varphi}_{i+1} - \bar{\varphi}_i}, & \text{if } i < p; \\ 1/(1 + \gamma_s), & \text{if } i = p. \end{cases}$$

and a particular choice of  $\eta$  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,

$$(2.4) \quad \bar{\varphi}_i = \min\{\varphi_i - \gamma_c P_{red}^{f, sb}(x_k, s_c), (1 - \gamma_f)\varphi_i\},$$

and

$$(2.5) \quad \bar{f}_i = \min\{f_i - \gamma_f P_{red}^{opt}(H_k, x_k, s_c), f_i - (\varphi_i - \bar{\varphi}_i)\},$$

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.

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

$$(2.6) \quad A_{red}(x_k, s_c, \theta_k) \leq \bar{\eta} P_{red}(H_k, x_k, s_c, \theta_k),$$

where

$$(2.7) \quad \theta_k = \begin{cases} \bar{\theta}_0, & \text{if } \varphi(x_+) < \bar{\varphi}_1; \\ \bar{\theta}_i, & \text{if } \bar{\varphi}_i \leq \varphi(x_+) < \bar{\varphi}_{i+1}; \\ \bar{\theta}_p, & \text{if } \varphi(x_+) \geq \bar{\varphi}_p. \end{cases}$$

The  $\bar{\eta}$  parameter is used in (2.6) 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 (2.4) and (2.5) will be used to define  $\bar{\varphi}_i$  and  $\bar{f}_i$ . Fortunately, this formula will only be used in the proof of Lemma 6.11 and, in this case, a simple expression is known.

Some agreement between the model function and the objective function of (1.1) 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

$$(2.8) \quad P_{red}^{opt}(H_k, x_k, s_c) \geq \kappa \varphi(x_k) \quad \text{and} \quad \theta_k^{sup} < \gamma_m,$$

where

$$(2.9) \quad \theta_k^{sup} = \sup\{\theta \in \mathbb{R} \mid A_{red}(x_k, s_c, \theta) \geq \gamma_g P_{red}(H_k, x_k, s_c, \theta)\},$$

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

In words, (2.8) 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. 2.1 are to be accepted. However, if we choose a small  $\gamma_m$ , say  $10^{-4}$ , (2.8) becomes less

stringent than (2.6) 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 2.2 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  $(\varphi, f)$ -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 (1.6), is given in red. The points that are acceptable for the filter but not for the new merit function are shown in yellow. 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  $\varphi(x) < 0.01$ .

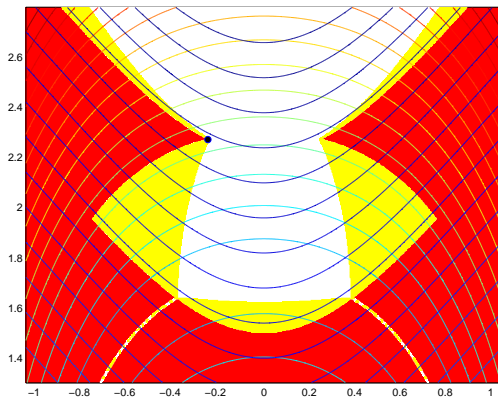


FIG. 2.2. The difference between forbidden regions. The yellow points are accepted by the filter but not by the new merit function.

**3. An SQP algorithm.** In the general framework of a trust region sequential quadratic programming algorithm, a step  $s_c$  is obtained approximating problem (1.1), in a neighborhood of an iterate  $x_k$ , by a quadratic programming (QP) problem.

In our case, this QP problem has the form

$$\begin{aligned}
 (3.1a) \quad & \text{minimize } Q(H_k, x_k, s) \\
 (3.1b) \quad & \text{subject to } A(x_k)s + C(x_k) = 0 \\
 (3.1c) \quad & l \leq x_k + s \leq u \\
 (3.1d) \quad & \|s\|_\infty \leq \Delta,
 \end{aligned}$$

where  $Q(H, x, s)$  is defined by (2.2),  $x_k$  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  $x_k$ . The infinity norm was chosen here so the constraints (3.1c) and (3.1d) can be grouped into one simple set of box constraints.



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

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

Unfortunately, if  $x_k$  is not  $\varphi$ -stationary, the constraints of (3.1) 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 (3.1) 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

$$(3.2) \quad \begin{aligned} & \text{reduce } M(x_k, s) \\ & \text{subject to } l \leq x_k + s \leq u \\ & \|s\|_\infty \leq 0.8\Delta. \end{aligned}$$

If  $M(x_k, s_n) = 0$ , then  $x_k$  can be substituted by  $x_k + s_n$  in (3.1) 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

$$(3.3) \quad \begin{aligned} & \text{reduce } Q(H_k, x_k, s) \\ & \text{subject to } A(x_k)s = A(x_k)s_n \\ & l \leq x_k + s \leq u \\ & \|s\|_\infty \leq \Delta. \end{aligned}$$

One should notice that the trust region radius was increased from (3.2) to (3.3). This is done to enlarge the feasible region when  $s_n$  is in the border of the trust region, so  $s_n$  is not the only solution of (3.3).

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  $\Omega$ . The solution of (3.2) is required to keep at least ninety percent of the reduction obtained by  $s_n^{dec}$ .

A similar procedure is adopted for (3.3). 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 (3.3) 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 2.1, 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 (3.1) 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 (3.1) 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 3.1. *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
 
$$\begin{aligned} & \text{minimize} && M(x_k, s) \\ & \text{subject to} && l \leq x_k + s \leq u \\ & && \|s\|_\infty \leq 0.8\Delta_k \\ & && s = td_n, t \geq 0; \end{aligned}$$
    - 1.3.3. Compute  $s_n$  (the normal step) such that
 
$$\begin{aligned} & l \leq x_k + s_n \leq u, \\ & \|s_n\|_\infty \leq 0.8\Delta_k, \text{ and} \\ & M(x_k, 0) - M(x_k, s_n) \geq 0.9[M(x_k, 0) - M(x_k, s_n^{dec})]; \end{aligned}$$
  - 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
 
$$\begin{aligned} & \text{minimize} && Q(s) \\ & \text{subject to} && l \leq x_k + s \leq u \\ & && \|s\|_\infty \leq \Delta_k \\ & && s = s_n + td_t, t \geq 0; \end{aligned}$$
  - 1.6. Compute a trial step  $s_c$  such that
 
$$\begin{aligned} & A(x_k)s_c = A(x_k)s_n, \\ & l \leq x_k + s_c \leq u, \\ & \|s_c\|_\infty \leq \Delta_k, \text{ and} \\ & Q(s_n) - Q(s_c) \geq 0.9[Q(s_n) - Q(s_t^{dec})]; \end{aligned}$$

- 1.7. IF  $(f(x_k + s_c) \geq \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_c)))$  OR  
 $(P_{red}^{opt}(H_k, x_k, s_c) \geq \kappa\varphi(x_k)$  AND  $\theta_k^{sup} < \gamma_m)$ ,
- 1.7.1.  $\Delta_k \leftarrow \alpha_R \min\{\Delta_k, \|s_c\|_\infty\}$ ; (reduce  $\Delta$ )
- 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:  
 $x_{k+1} \leftarrow x_k + s_c$ ;  
 $\tilde{\rho}_k \leftarrow A_{red}(x_k, s_c, \theta_k^{sup})/P_{red}(H_k, x_k, s_c, \theta_k^{sup})$ ;  
 $\Delta_{k+1} \leftarrow \begin{cases} \max\{\alpha_R \min\{\Delta_k, \|s_c\|_\infty\}, \Delta_{min}\}, & \text{if } \tilde{\rho}_k < \gamma_g, \\ \max\{\alpha_A \Delta_k, \Delta_{min}\}, & \text{if } \tilde{\rho}_k \geq \eta; \end{cases}$   
Determine  $H_{k+1}$ ;  
 $k \leftarrow k + 1$ ;
- 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  
 $(\varphi(x_k + s_r) < \epsilon_h \Delta_k^2$  AND  $f(x_k + s_r) < \mathcal{P}(\mathcal{F}_k, \varphi(x_k + s_r)))$  OR  
 $x_k + s_r$  is  $\varphi$ -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:  
 $x_{k+1} \leftarrow x_k + s_r$ ;  
 $\Delta_{k+1} \leftarrow \max\{\beta \Delta_{rest}, \Delta_{min}\}$ ;  
Determine  $H_{k+1}$ ;  
 $k \leftarrow k + 1$ ;

The constants used here must satisfy  $\kappa > 0$ ,  $0 < \gamma_r < \gamma_g < \eta < 1$ ,  $\gamma_n > 0$ ,  $\gamma_t > 0$ ,  $0 < \Delta_{min} < \Delta_{rest}$ ,  $0 < \alpha_R < 1$ ,  $\alpha_A \geq 1$ ,  $\epsilon_h > 0$  and  $\beta > 0$ . Parameters  $\kappa$ ,  $\gamma_n$ ,  $\gamma_t$ ,  $\Delta_{min}$ ,  $\Delta_{rest}$ ,  $\epsilon_h$  and  $\beta$  are problem dependent and must be chosen according with some measure of problem data. Reasonable values for the remaining parameters might be  $\gamma_r = 0.01$ ,  $\gamma_g = 0.05$ ,  $\eta = 0.9$ ,  $\alpha_R = 0.25$  and  $\alpha_A = 2.5$ . The constant  $\eta$  should not be confused with the parameter  $\bar{\eta}$  defined in (2.6).

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 1.1 to be refused by the algorithm. Moreover, it assures the algorithm is well defined, as stated in the next section.

**4. The algorithm is well defined.** An iteration of algorithm 3.1 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  $\varphi$ -stationary but infeasible point or a feasible but not regular point.

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

LEMMA 4.1. *If  $x_k$  is not  $\varphi$ -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,  $\Delta_k$  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  $\varphi$ -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  $\varphi(x_k + s_r) < \min\{\overline{\varphi}_1, \epsilon_h \Delta_k^2\}$ , 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  $x$ .

**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  $\Delta^2$ , so the step is accepted for a sufficiently small trust region radius, as stated in the following lemma.

LEMMA 4.2. *Suppose that A1 and A2 hold and that  $x_k$  is feasible and regular for problem 1.1 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

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

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

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

$$P_{red}^{opt}(H_k, x_k, s_c(\Delta)) = Q(0) - Q(s_c(\Delta)) \geq 0.9[Q(0) - Q(s_t^{dec})] \geq 0.9c\Delta.$$

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

$$(4.1) \quad A_{red}(x_k, s_c(\Delta), \bar{\theta}_1) \geq \eta_1 P_{red}(H_k, x_k, s_c(\Delta), \bar{\theta}_1),$$

where,

$$(4.2) \quad P_{red}(H_k, x_k, s_c(\Delta), \bar{\theta}_1) = \bar{\theta}_1 P_{red}^{opt}(H_k, x_k, s_c(\Delta)) \geq 0.9c\Delta\bar{\theta}_1,$$

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

and  $\bar{\theta}_1 > 0$  is given by (2.3).

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

$$(4.3) \quad A_{red}(x_k, s_c, \theta) = P_{red}(H_k, s_k, s_c, \theta) + \bar{c}_1 \|s_c\|^2.$$

So, using (4.2) and (4.3) we deduce that

$$(4.4) \quad \left| \frac{A_{red}(\Delta)}{P_{red}(\Delta)} - 1 \right| \leq \frac{\bar{c}_1 \Delta}{0.9c\bar{\theta}_1}.$$

Thus, for  $\Delta < \min\{(1 - \eta_1)0.9c\bar{\theta}_1/\bar{c}_1, \bar{\Delta}_2\} = \bar{\Delta}_3$ , the inequality (4.1) necessarily takes place.

Now, using the fact that  $\theta_k^{sup} = 1$  for  $x_k$  feasible and replacing  $\bar{\theta}_1$  by 1 in (4.4), we can conclude that, for

$$(4.5) \quad \Delta < \min\{(1 - \gamma_g)0.9c/\bar{c}_1, \bar{\Delta}_3\} = \bar{\Delta}_4,$$

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  $\varphi$ -stationary but infeasible point is found. Moreover, the restoration procedure can also fail to obtain a more feasible point. Naturally, this unexpected behavior of the algorithm makes somewhat pretentious the title of this section.

Formally, what we will prove in this section is that, supposing that a  $\varphi$ -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  $\Omega_0$ .

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 (2.1)) 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 5.1. *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

$$(5.1) \quad \varphi(x_{k_j}) \geq \epsilon,$$

where  $\epsilon > 0$ .

At iteration  $k_j$ , the  $(\varphi, f)$ -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  $(\varphi_{k_j}, f_{k_j})$  remains in  $F$ , no other  $(\varphi, f)$ -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 (2.4), (2.5) and (5.1), 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  $(\varphi_{k_j}, f_{k_j})$  is excluded from  $F$  by another pair  $(\varphi_{k_l}, f_{k_l})$ , included in  $F$  at an iteration  $k_l > k_j$ . This case is illustrated in Fig. 5.1. Notice that  $(\varphi_{k_l}, f_{k_l})$  cannot fall in regions I and V since, in this case,  $(\varphi_{k_j}, f_{k_j})$  will not be excluded from  $F$ . It can be easily verified that the worst case occurs when  $(\varphi_{k_l}, f_{k_l})$  lies on  $\ell_1(\varphi)$  or  $\ell_2(\varphi)$ .

Suppose  $(\varphi_{k_l}, f_{k_l})$  lies on  $\ell_2(\varphi)$ , as depicted in Fig. 5.1. In this case, the rectangle  $r_m$  will be entirely above  $\bar{\ell}_2$ , the line that connects  $(\bar{\varphi}_{k_l}, \bar{f}_{k_l})$  to  $(\bar{\varphi}_{m+1}, \bar{f}_{m+1})$ . Since  $\bar{\ell}_2$  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  $(\varphi_{k_l}, f_{k_l})$  lies on  $\ell_1(\varphi)$ . Therefore, once  $(\varphi_{k_j}, f_{k_j})$  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 (5.1), and the conclusion follows.  $\square$

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

LEMMA 5.2. *Suppose that assumptions A1 and A3 hold. Suppose also that, for all  $k > k_0$ ,  $x_k$  is never included in  $F_k$ . Then,*

$$(5.2) \quad \lim_{k \rightarrow \infty} \varphi(x_k) = 0.$$

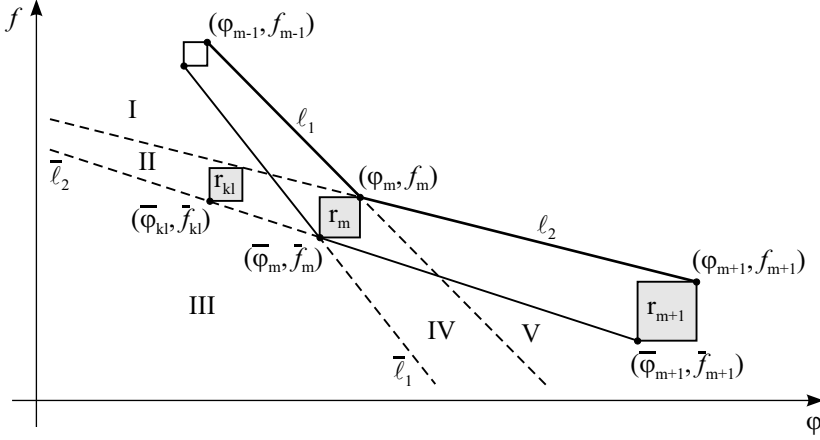


FIG. 5.1. Adding a new iterate that excludes  $(\varphi_{k_j}, f_{k_j})$  from  $F$ .

*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 3.1 are never satisfied for  $k > k_0$ . Therefore, we have

$$(5.3) \quad f(x_k) - f(x_{k+1}) \geq \gamma_r P_{red}^{opt} \geq \gamma_r \kappa \varphi(x_k) > 0,$$

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

$$(5.4) \quad \lim_{k \rightarrow \infty} f(x_k) - f(x_{k+1}) = 0.$$

Then, (5.2) follows from (5.3) and (5.4).  $\square$

**6. The algorithm finds a critical point.** Finally, we are able to prove the convergence of the algorithm to a stationary point for (1.1). In order to do that, we will need to make one additional assumption on the choice of the normal step  $s_n$ .

**A4.** The choice of  $s_n$  at step 1.3.3 of algorithm 3.1 is such that

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

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 (2.2) along  $d_t$  is bounded away from zero.

**LEMMA 6.1.** *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 (1.1). Then there exists  $k_1, c_1 > 0$  such that*

$$(6.1) \quad -\nabla Q(s_n(x, \Delta))^T d_t(H, x, \Delta) \geq c_1$$

for all  $x \in \{x_{k_i} \mid k \geq k_1\}$ . Moreover,  $\|d_t(H, x, \Delta)\|$  is bounded and bounded away from 0 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,

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

So, by A2 and A4, we have that

$$(6.2) \quad \|d_t(H, x, \Delta) - P_x(-\gamma_t \nabla f(x))\|_2 \leq \gamma_1 \|C(x)\|$$

and, by the continuity of  $\nabla f(x)$  and the fact that  $\{x_{k_i}\}$  converges, we deduce that

$$(6.3) \quad \|\nabla f(x_{k_i})^T P_x(-\gamma_t \nabla f(x_{k_i})) - \nabla f(x_{k_i})^T d_t(H_{k_i}, x_{k_i}, \Delta_{k_i})\|_2 \leq \gamma_2 \|C(x_{k_i})\|.$$

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

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

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

$$(6.4) \quad \begin{aligned} & \text{minimize } \|-\gamma_t \nabla f(x^*) - z\|_2^2 \\ & \text{subject to } A(x^*)z = 0 \\ & \quad l \leq x^* + z \leq u. \end{aligned}$$

Since  $x^*$  is regular but is not a stationary point for (1.1), it follows that  $z = 0$  is not a solution for (6.4). So,  $P_{x^*}(-\gamma_t \nabla f(x^*)) \neq 0$ . Moreover, since  $z = 0$  is feasible for (6.4), 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  $\bar{k}_2 \in \mathcal{N}$  such that, for all  $x \in \{x_{k_i} \mid k \geq \bar{k}_2\}$ , we have

$$(6.5) \quad \bar{c}_2 \leq \|P_x(-\gamma_t \nabla f(x))\| \leq \bar{c}_3 \quad \text{and} \quad \nabla f(x)^T P_x(-\gamma_t \nabla f(x)) \leq -\bar{c}_4.$$

Now, from (6.2), (6.3) and (6.5), the continuity of  $C(x)$  and the feasibility of  $x^*$ , there exists  $\bar{k}_3 \geq \bar{k}_2$  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, (6.1) 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 6.1, we state in the next lemma that, in the neighborhood of a feasible, regular and non-stationary point, the decrease of the quadratic model (2.2) is proportional to the trust region radius  $\Delta$ .

LEMMA 6.2. *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 (1.1). Then there exists  $c_2, k_2 > 0$  and  $\Delta_1 \in (0, \Delta_{min})$  such that*

$$(6.6) \quad Q(x, s_n(x, \Delta)) - Q(x, s_c) \geq c_2 \min\{\Delta, \Delta_1\}$$

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 6.3. *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 (1.1). Then there exists  $\epsilon, c_3, k_3 > 0$  and  $\Delta_1 \in (0, \Delta_{min})$  such that, for  $k_i > k_3$ , if*

$$(6.7) \quad \varphi(x_{k_i}) \leq \epsilon \Delta^2,$$

we have that

$$(6.8) \quad P_{red}^{opt}(x_{k_i}, s_c) = Q(x_{k_i}, 0) - Q(x_{k_i}, s_c) \geq c_3 \min\{\Delta, \Delta_1\}.$$

*Proof.* By Lemma 6.2, 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  $\Delta_1$  are defined as in Lemma 6.2 and  $\gamma_4 > 0$ . Therefore, (6.8) 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  $\Delta$  is bounded away from zero and an infinite subsequence of points is added to  $F$ .

LEMMA 6.4. *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 restoration always terminates successfully and that  $\Delta_{k_j} \geq \Delta_2$ , where  $\Delta_2$  is a positive scalar. Then there exists a limit point of this sequence that is a stationary point for (1.1).*

*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 (1.1).

From Lemma 5.1 we know that there exists  $\bar{k}_5 \in \mathbb{N}$  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 (1.1). So, from Lemma 5.1 and (6.8), 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 5.1 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 6.3), we deduce from (6.8) 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 (1.1).  $\square$

Supposing again that  $\Delta$  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 6.5. *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  $\Delta_3$ . 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 stationary point of (1.1).*

*Proof.* Assumption A3 implies that there exists a convergent subsequence  $\{x_{k_i}\}$ . If the limit point of this subsequence is not stationary for (1.1), then from Lemma 6.3, we have

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

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 that

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

Therefore,  $f(x_{k_i}) - f(x_{k_i} + s_c) \geq \gamma_r c_3 \min\{\Delta_1, \Delta_3\}$  for all  $k_i$  sufficiently large, which contradicts the compactness assumption A3.  $\square$

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  $\Delta$  is small.

LEMMA 6.6. *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 (1.1). Suppose also that  $\varphi_k$  satisfies (6.7) and that*

$$(6.9) \quad \Delta < \min\{c_3/(\kappa\epsilon), \Delta_1\} = \Delta_5$$

for  $k_i > k_6$ , where  $c_3$ ,  $\epsilon$  and  $\Delta_1$  are defined as in Lemma 6.3. 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 (6.8), 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 (6.9). 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  $\Delta$  is not reduced further at step 1.7.1 of algorithm 3.1.

The first lemma shows the relation between the predicted reduction of the infeasibility and  $\Delta$ .

LEMMA 6.7. *Suppose that assumption A1 holds and that  $x_k$  is not  $\varphi$ -stationary. Then there exists  $\Delta_6, c_4 > 0$  such that*

$$(6.10) \quad P_{red}^{fsb}(x_k, s_c) \geq c_4\Delta_k,$$

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

*Proof.* Since  $x_k$  is not  $\varphi$ -stationary, we have that  $d_n \neq 0$ . Thus, we can define

$$t^{max}(\Delta) = \max\{t > 0 \mid [x_k, x_k + td_n] \in \Omega \text{ and } \|td_n\| \leq 0.8\Delta\}.$$

Clearly, if  $\Delta \leq 1.25\|d_n\|$ , then  $\|t^{max}(\Delta)d_n\| = 0.8\Delta$  (since  $x_k + d_n \in \Omega$ ). Now, define

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

By some elementary properties of one-dimensional quadratics, there exists  $\Delta_6 \in (0, \|d_n\|]$  such that

$$M(0) - M(t^{max}(\Delta)) \geq -\frac{1}{2} d_n^T \nabla \varphi(x_k) t^{max}(\Delta) = c \|d_n\| t^{max}(\Delta) = 0.8c\Delta.$$

for all  $\Delta \in (0, \Delta_6)$ . Therefore, for the normal step  $s_n$  computed at step 1.3.3 of algorithm 3.1, we have

$$M(0) - M(s_n) \geq 0.72c\Delta_k.$$

But, since  $A(x_k)s_c = A(x_k)s_n$ , we deduce from (1.4) that

$$P_{red}^{fsb}(x_k, s_c) \geq 0.72c\Delta_k$$

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  $\Delta$ ,  $\bar{\varphi}_i$ , defined in (2.4), will depend on the predicted reduction of the infeasibility.

LEMMA 6.8. *Suppose that A1 holds and that  $x_k$  is not  $\varphi$ -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 6.7 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  $\Delta_6$  is given in Lemma 6.7, 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 6.7 again, we can also show that  $\bar{f}$ , defined in (2.5), will depend on  $P_{red}^{opt}$  if  $\Delta$  is sufficiently small.

LEMMA 6.9. *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 (1.1), and that  $\bar{\varphi}_k$  is given by (2.4). 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  $\epsilon$  is defined as in Lemma 6.3.

*Proof.* From Lemma 6.3 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  $\Delta$  is small, not matter how the penalty parameter  $\theta$  is chosen.

LEMMA 6.10. *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 (1.1). 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 6.3) such that, for  $k_i > k_7$ , the condition  $\varphi(x_{k_i}) \leq \epsilon \Delta_{k_i}^2$  is satisfied, so (6.8) holds. Besides,

since that  $x_{k_i}$  is infeasible, (6.10) also holds if we take  $\Delta_9 \leq \Delta_6$  (defined in Lemma 6.7). 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  $\Delta_1$ ,  $c_3$  and  $c_4$  are defined as in Lemmas 6.3 and Lemma 6.7.

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  $\Delta_{k_i} \leq \Delta_1$ ,

$$(6.11) \quad \frac{|A_{red}(\Delta_{k_i}) - P_{red}(\Delta_{k_i})|}{P_{red}(\Delta_{k_i})} = \left| \frac{A_{red}(\Delta_{k_i})}{P_{red}(\Delta_{k_i})} - 1 \right| \leq \frac{c_5}{\min\{c_3, c_4\}} \Delta_{k_i}.$$

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 (1.1).

LEMMA 6.11. *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 (1.1).*

*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 (1.1).

Now, we need to consider separately two mutually exclusive situations. First, let us suppose that  $x_{k_i}$  is feasible. In this case, Lemma 4.2 assures that, for  $\Delta < \bar{\Delta}_4$  (defined in (4.5)), 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  $\varphi$ -stationary, Lemmas 6.8 and 6.9 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 (2.6) 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 (2.6).

Now, using Lemmas 6.6 and 6.10, 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 (1.1).  $\square$

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

THEOREM 6.12. *Suppose that A1, A2, A3 and A4 hold and that  $\{x_k\}$  is an infinite sequence generated by algorithm 3.1. Then either the restoration converges to a  $\varphi$ -stationary but infeasible point of (1.1), 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 stationary point for (1.1). In particular, if all of the  $\varphi$ -stationary points are feasible and regular, then there exists a subsequence of  $\{x_k\}$  that converges to a feasible, regular and stationary point of (1.1).

*Proof.* This result is a direct consequence of Lemmas 5.1, 5.2, 6.4, 6.5 and 6.11.  $\square$

**7. Numerical experience.** The description of some steps of Algorithm 3.1 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 3.1 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 (3.2), 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

$$M(x_k, 0) - M(x_k, s_n) \geq 0.9[M(x_k, 0) - M(x_k, s_n^{dec})]$$

and

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

are satisfied, where  $g_p$  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 (3.3), stopping the algorithm when both

$$(7.1) \quad \|g_p(s_c)\|_\infty \leq 10^{-6} \left\{ \frac{\|\lambda\|_1}{\sqrt{m}}, 1 \right\},$$

and condition (6.6) are satisfied, where, again,  $g_p$  is the projected gradient and  $\lambda$  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

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

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 (6.6) and (7.1). In this case, after determining  $d_n$  and  $s_t^{dec}$ ,  $s_c$  was 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 (1.1).

Algorithm 3.1 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  $\lambda_{typ}$  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  $\varphi$ -stationary but infeasible point was found.

The remaining parameters used are  $\Delta_0 = 10^5$ ,  $\Delta_{min} = 10^{-5}$ ,  $\gamma_n = 1.0$ ,  $\gamma_t = 10^{-8}$ ,  $\gamma_r = 10^{-2}$ ,  $\gamma_g = 5.10^{-2}$ ,  $\kappa = 10^{-4}$ ,  $\eta = 0.9$ ,  $\alpha_A = 2.5$ ,  $\alpha_R = 0.25$ ,  $\beta = 16.0$ ,  $\gamma_s = 10^{-4}$ ,  $\gamma_f = 10^{-6}$ ,  $\gamma_c = 10^{-3}$ ,  $\gamma_m = 10^{-4}$ ,  $\Delta_{rest} = 10^{-1}$  and  $\epsilon_h = 10^2$ .

**7.2. Numerical results.** 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 7.1, 7.2 and 7.3. 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 (1.1). The number of constraints is given by  $m$ .

TABLE 7.1  
*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 7.2  
*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				

TABLE 7.3  
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	2004		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				

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*.

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 7.4 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 7.5 shows the number of times GMM99 and Lancelot obtained the best



TABLE 7.4  
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

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 horizontal step in GMM99. In general, GMM99 had a slightly better performance.

TABLE 7.5  
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

The new algorithm was also compared with GMM, the standard trust-region SQP method presented in [10]. The results are shown in Tables 7.6 and 7.7.

TABLE 7.6  
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 CATE-NARY, 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 7.6 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 7.6, 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 7.7 shows how the good performance of GMM and GMM99 is distributed among different types of problems.

TABLE 7.7  
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

Tables 7.6 and 7.7 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.

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  $\theta$  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  $(\varphi, f)$ -plane, but also as a convex piecewise filter.
- The effective use of the points in  $F$  to define the forbidden region in the  $(\varphi, f)$ -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 reject points if they belong to the forbidden region depicted in Figure 2.1 or if condition (2.8) 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  $(\varphi, f)$ -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*, Tech. Report TR00-09, Dep. Computational and Applied Mathematics, Rice University, Houston, TX, 2000.
- [2] R. H. BIELSCHOWSKY, A. FRIEDLANDER, F. A. M. GOMES, J. M. MARTÍNEZ AND M. RAYDAN, *An adaptive 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. Optim.*, 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, PA, 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. Optim.*, 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. Optim.*, 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. Optim.*, 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*, Tech. Report 03/15, Dep. Mathematics, University of Namur, Namur, Belgium, 2003.
- [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*, Tech. Report 04/03, Dep. Mathematics, University of Namur, Namur, Belgium, 2004.
- [16] N. I. M. GOULD AND P. L. TOINT, *Global convergence of a hybrid trust-region SQP-filter algorithm for general nonlinear programming*, Tech. report 01/07, Dep. Mathematics, University of Namur, Namur, Belgium, 2001.
- [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, Springer, Heidelberg, 1981.
- [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, NY, 1999.
- [21] T. D. PLANTENGA, *A trust-region method for nonlinear programming based on primal interior point techniques*, SIAM J. Sci. Comput., 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, *Global and local convergence of line search filter methods for nonlinear programming*, Tech. Report CAPD B-01-09, Dep. Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 2001.
- [25] A. WÄCHTER AND L. T. BIEGLER, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, IBM Research Report RC23149, IBM T. J. Watson Research Center, Yorktown Heights, NY, 2004.