A Brief History of Filter Methods*

Roger Fletcher†, Sven Leyffer‡, and Philippe Toint§

September 26, 2006 revised October 9, 2006

Abstract

We consider the question of global convergence of iterative methods for nonlinear programming problems. Traditionally, penalty functions have been used to enforce global convergence. In this paper we review a recent alternative, so-called filter methods. Instead of combing the objective and constraint violation into a single function, filter methods view nonlinear optimization as a biobjective optimization problem that minimizes the objective and the constraint violation. We outline the main ideas and convergence results of filter methods and indicate other areas where filter methods have been used successfully.

Keywords: Large-scale nonlinear programming, interior-point methods, sequential quadratic programming, filter methods.

AMS-MSC2000: 90C06, 90C30, 90C51, 90C55.

1 Motivation

We consider the question of global convergence for optimization algorithms that solve general nonlinear programming problems (NLPs):

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) \geq 0, \end{array} \tag{1.1}$$

where the objective function $f(x)$ and the constraint functions $c(x)$ are smooth.

Most methods for solving (1.1) are based on Newton's method and are iterative. Given an estimate x_k of the solution x_* of (1.1), a linear or quadratic approximation of (1.1) is solved for a new and, one hopes better, estimate x_{k+1}. Near a solution, this process is guaranteed to converge. Far from the solution, however, the sequence $\{x_k\}$ generated in this way may not converge. How can we ensure convergence even if we start far from a solution? We refer to this question as *global convergence* for NLP methods.

Traditionally, this question has been answered by using penalty or merit functions that are a linear combination of the objective function and a measure of the constraint violation such as

*Preprint ANL/MCS-P1372-0906

†Department of Mathematics, University of Dundee, Dundee, DD1 4HN, UK, fletcher@maths.dundee.ac.uk

‡Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA, leyffer@mcs.anl.gov

§Department of Mathematics, The University of Namur (FUNDP), rue de Bruxelles, B5000 Namur, Belgium, philippe.toint@fundp.ac.be

$h(x) := \|c(x)^-\|$, where $\|a^-\| = \|\min(a,0)\|$ for some norm. An example is the ℓ_1 exact penalty function,

$$p(x;\pi) := f(x) + \pi h(x),$$

where $\pi > 0$ is the penalty parameter. Provided π is sufficiently large, we can use this penalty function to ensure progress in our iterative scheme by enforcing sufficient decrease on each step. This trick allows us to invoke well-developed unconstrained optimization techniques.

Unfortunately, a suitable penalty parameter depends on the solution of (1.1), namely, $\pi > \|y_*\|_D$, where y_* are the optimal multipliers and $\|\cdot\|_D$ is the dual norm. This fact makes it difficult to find a suitable penalty parameter. Worse, if the penalty parameter is too large, then any monotonic method would be forced to follow the nonlinear constraint manifold very closely, resulting in much shortened Newton steps and slow convergence. Yet we have noticed that the unmodified sequential quadratic programming (SQP) method is able to quickly solve a large proportion of test problems without the need for modifications to induce global convergence.

In this paper we review a recent alternative to penalty functions, so-called filter methods. The success of the unmodified SQP method motivates us to find a way of inducing global convergence, which would allow the full Newton step to be taken much more often. Our goal therefore is the development of global optimization safeguards that *interfere as little as possible with Newton's method.* We believe filter methods achieve this goal. In the remainder of this paper, we motivate filter methods, outline the main ideas and convergence results, indicate other areas where filter methods have been used successfully, and provide references for those wishing to delve deeper into filter methods.

2 Filter Methods for NLP

Filter methods avoid the pitfalls of penalty function methods. Instead of combing the objective and constraint violation into a single function, we view (1.1) as a biobjective optimization problem that minimizes $f(x)$ and $h(x)$. However, the second objective is clearly more important because we must ensure that $h(x_*) = 0$. We borrow the concept of domination from multiobjective optimization and say that a point x_k *dominates* a point x_l if and only if $f(x_k) \le f(x_l)$ and $h(x_k) \le h(x_l)$. We define a *filter* as a list of pairs $(h(x_l), f(x_l))$ such that no pair dominates another pair. A typical filter is illustrated in Figure 1, where the shaded area shows the region dominated by the filter entries. The contours of the ℓ_1 exact penalty function would be straight lines with slope $-\pi$ in this plot, indicating that at least for a single entry, the filter is less restrictive than penalty methods.

2.1 Sequential Quadratic Programming Filter Methods

Filter methods were first introduced in the context of trust-region SQP methods, which solve a quadratic approximation of (1.1) for a trial step s that lies inside a trust region:

$$\begin{array}{ll} \min\limits_{s} & q_k(s) := f_k + \nabla f_k^T s + \frac{1}{2} s^T H_k s \\ \text{s.t.} & c_k + \nabla c_k^T s \ge 0 \\ & \|s\|_\infty \le \rho_k, \end{array} \tag{2.1}$$

where $f_k = f(x_k)$ and so on, and $H_k \simeq \nabla^2 \mathcal{L}_k$ approximates the Hessian of the Lagrangian.

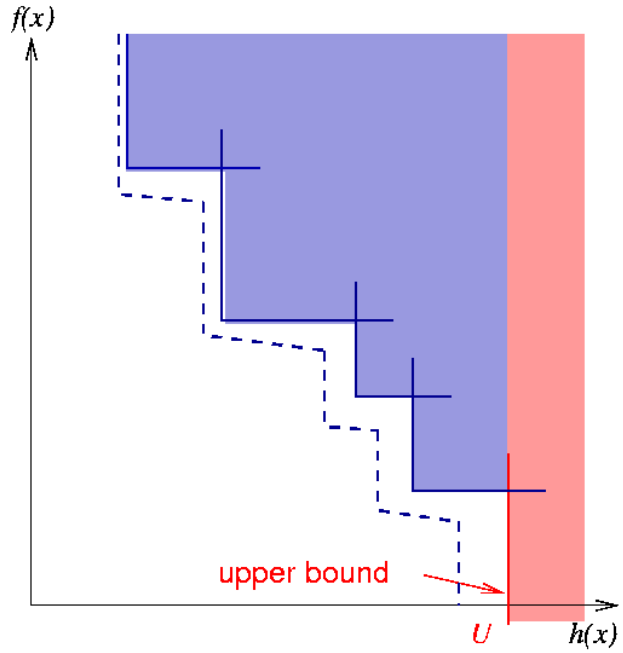

Figure 1: A typical filter. All pairs $(f(x), h(x))$ that are below and to the left of the envelope (dashed line) are acceptable to the filter

A rough outline of a filter trust-region SQP is as follows. At iteration $k = 0$, we initialize the filter $\mathcal{F}_k = \{(U, -\infty)\}$, where U is an upper bound on the acceptable constraint violation. We proceed by accepting only steps that are not dominated by the current filter. If a point is acceptable, then we set $x_{k+1} = x_k + s$, and possibly increase the trust-region radius and update the filter (adding $(_k, f_k)$ from the previous iterate and removing any dominated entries). If, on the other hand, the step is dominated by the current filter, then we reject it, set $x_{k+1} = x_k$, reduce the trust-region radius, and resolve (2.1).

This simple description of a filter method requires a number of refinements to ensure convergence:

1. *Filter Envelope.* To avoid convergence to infeasible limit points where $h_* > 0$, we add an envelope around the current filter. A new iterate is acceptable if, for all $\forall (h_l, f_l) \in \mathcal{F}_k$,

$$h_{k+1} \leq \beta h_l, \quad \text{or} \quad f_{k+1} \leq f_l - \gamma h_{k+1}, \tag{2.2}$$

   where $0 < \beta, \gamma < 1$ are constants. This *sloping envelope* is due to Chin [4, 3] and makes the management of redundant entries slightly more convenient. In [4, Lemma 1] it is shown that if an infinite number of points are added to the filter, and $f(x)$ is bounded below, then the limit point must be feasible. We note, that this result does not require the presence of an upper bound U.

2. *Sufficient Reduction.* The filter alone cannot ensure convergence to stationary points. For example, if the sequence satisfies $h_{k+1} \leq \beta h_k$, then the iterates could converge to an arbitrary feasible point. Therefore, if the constraint violation becomes small, we enforce a sufficient

reduction condition similar to unconstrained optimization. We denote the predicted reduction by $\Delta q_k := -\nabla f_k^T s - \frac{1}{2} s^T H_k s$ and introduce the following *switching condition*:

$$\begin{array}{l} \text{if } \ (\ \Delta q_k > 0\ ) \ \text{ then} \\ \qquad \text{check } \ f_k - f_{k+1} \geq \sigma \Delta q_k, \end{array} \tag{2.3}$$

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

3. *Feasibility Restoration.* By reducing the trust-region radius, the QP (2.1) may become inconsistent (halving the trust-region radius in the right plot of Figure 2 illustrates this point). We take the inconsistency of (2.1) as an indication that the current point is too far from the feasible set to make meaningful progress to optimality. Hence we invoke an SQP-like algorithm that minimizes the constraint violation $h(x)$ (see Section 3.1). We exit the restoration phase once a filter-acceptable point has been found and resume the regular SQP method.

With these modifications, we can define acceptance for a filter method.

Definition 2.1 *A trial point $x_k^+ := x_k + s$ is* acceptable to the filter *at iteration k if*

1. *x_k^+ is acceptable to the filter $\mathcal{F}_k$ and x_k, that is, (2.2) holds for $\mathcal{F}_k \cup \{h_k, f_k\}$, and*
2. *if the switching condition $\Delta q_k > 0$ holds, then we have sufficient reduction, that is, $f_k - f(x_k^+) \geq \sigma \Delta q_k$.*

Otherwise, we call x_k^+ not acceptable.

An outline of a filter method is given next.

Algorithm 1: SQP Filter Method

```
x_0, k ← 0, F_0 ← {U, −∞}, optimal ← false
while not optimal do
    reset the trust-region radius: ρ_k ≥ ρ̲
    terminate ← false
    repeat
        solve the QP (2.1) for a step s
        if s = 0 then
            optimal ← true; STOP
        if QP (2.1) incompatible then
            add (h_k, f_k) to F_k
            enter restoration phase
        else
            if x_k^+ := x_k + s not acceptable then
                reduce trust-region ρ_k ← ρ_k/2
            else
                terminate ← true
    until terminate
    update the filter F_{k+1}
    set x_{k+1} ← x_k + s and k ← k + 1
```

2.1.1 Discussion of Filter Algorithm

Algorithm 1 contains an inner and an outer iteration. During the inner iteration the trust-region radius is reduced until we either find an acceptable point or enter the restoration phase. The aim of the restoration phase is to find an acceptable iterate x_{k+1} such that the corresponding QP (2.1) is compatible for some $\rho_{k+1} \geq \underline{\rho}$. The iterates and the filter are updated in the outer iteration, which also ensures that the trust-region radius is larger than a lower bound $\underline{\rho} > 0$.

We update the filter by adding entries (h_k, f_k) to $\mathcal{F}_k$ that correspond to an h-type iteration *after we move to* x_{k+1}. We can also remove any entries that are dominated by (h_k, f_k).

The switching condition (2.3) can be motivated as follows. Close to a feasible point, we expect the quadratic model to predict a decrease in the objective function, that is, $\Delta q_k > 0$. However, far from a feasible point, the predicted reduction is sometimes negative, that is $\Delta q_k < 0$, because most of the SQP step is toward feasibility. We will refer to successful steps that satisfy (2.3) as *f-type steps* and all other steps as *h-type* steps. This is illustrated in Figure 2: the left plot shows an f-type step that reduces $q_k(s)$, while the right plot shows an h-type step that reduces only infeasibility. We note that if $h_k = 0$ at a nonstationary point, then $\Delta q_k > 0$, thereby implying that we can accept only an f-type step. Thus, we never add points to the filter for which $h_k = 0$. This fact ensures that we can always generate a filter-acceptable point during the restoration phase unless the problem is (locally) infeasible.

2.1.2 Early History of Filter Methods

NLP filter methods were first proposed by Fletcher in a plenary talk at the SIAM Optimization Conference in Victoria in May 1996; the methods are described in [8]. The initial filter method contained features, such as the NW/SE corner rule and unblocking, that were shown to be redundant in the subsequent convergence analysis. The first global convergence proof of a filter method was given in [11] for a sequential linear programming (SLP) method. This proof was later generalized to SQP methods in [10].

Filter methods for NLP were developed independently of earlier similar ideas. Surry et al. [25] describe a multiobjective approach to constrained optimization in the context of genetic algorithms. The algorithm maintains a population of iterates that are evolved over time. The authors modify a vector-evaluated genetic algorithm to adaptively bias the population toward feasibility.

An idea similar to a filter was used by Lemaréchal et al. [20] to enforce convergence of a bundle method for convex nonsmooth constrained optimization problems. The authors define an exclusion region that corresponds to the *convex hull* of the filter entries (with the two out-most entries extended to infinity).

2.1.3 Convergence Proof Outline

Convergence of filter methods can be established under the following general assumptions: the iterates x_k lie in a compact set X, the functions $f(x)$ and $c(x)$ are twice continuously differentiable, and the Hessian remains bounded $\|H_k\| \leq M$. Under these assumptions, one of the following occurs (see [10, Theorem 7], [4, Theorem 1], and [27, Theorem 3]):

1. The restoration phase fails to find a filter-acceptable point for which the QP (2.1) is consistent for some $\rho \geq \underline{\rho}$.

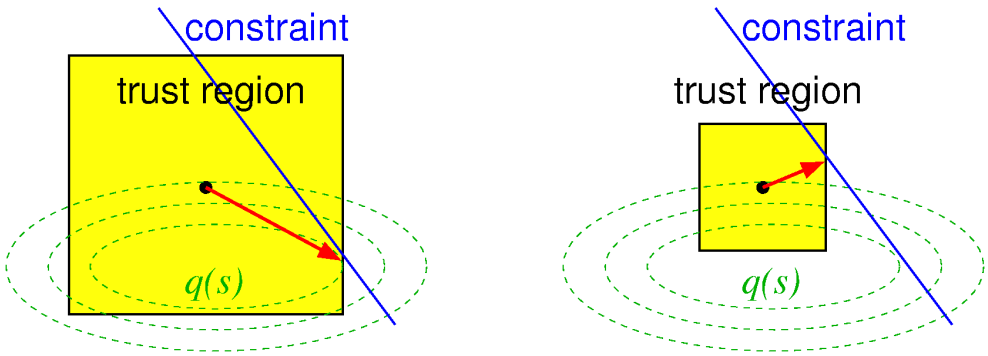

Figure 2: Illustration of f-type and h-type step.

2. The algorithm terminates at a first-order stationary point.

3. There exists a feasible accumulation point that either is stationary or the Mangasarian-Fromowitz constraint qualification fails.

These results are as strong as can be expected for general NLPs. For example, the first outcome corresponds to a situation where the restoration phase has converged to a local minimum of the constraint violation. One undesirable assumption in [10] is the need for global solution to the QP subproblem (2.1). This assumption may be difficult to ensure, unless H_k is positive semi-definite.

The convergence proof makes use of the insights from Figure 2. The filter ensures that all limit points are feasible. Next, we consider two cases: (a) an infinite subsequence of h-type iterations, and (b) an infinite sequence of f-type steps. We assume that the limit point is not stationary and seek a contradiction. In case (a), we can show that for sufficiently large k we must generate an f-type iteration, which contradicts the construction of the sequence. In case (b), we obtain the usual contradiction that $f(x_k)$ is unbounded below.

2.1.4 Fast Local Convergence

The transition of filter methods to fast local convergence had been an outstanding issue from the start. Early on, we conjectured that filter methods may be able to avoided the Maratos effect. This effect causes penalty function SQP methods to reject the full SQP step arbitrarily close to a solution, leading to a loss of second-order convergence. We applied filter methods to the original example by Maratos and observed second-order convergence. However, the following example shattered the hope that filter methods can avoid the Maratos effect in general:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & 2(x_1^2 + x_2^2 - 1) - x_1 \\ \text{subject to} & x_1^2 + x_1^2 - 1 = 0. \end{array}$$

The starting point $x = (\cos(t), \sin(t))$ for $t > 0$ small and multipliers $y = 3/2$ shows that the SQP step increases *both* $f(x)$ and $\|c(x)\|$, leading to a filter-rejected step. This example motivated us to include second-order correction (SOC) steps. Since then, Ulbrich [27] and Wächter and Biegler [29] have considered the transition to fast local convergence.

Ulbrich [27] proves fast local convergence *without the use of SOC steps* by making three modifications to the filter SQP method: (1) he uses the augmented Lagrangian as a technical tool, which

motivates an alternative definition of the filter, replacing $h(x)$ and $f(x)$ by

$$\theta(x,y) := \|c(x)^-\|_2^2 + (y^T c(x)^-)^2 \;\text{ and }\; \mathcal{L}(x,y)$$

respectively; (2) the switching condition (2.3) is tightened to

$$\Delta\widehat{q_k} := \Delta q_k + y_k^T s_k > \kappa\theta_k^{\psi/2} \text{ and } \Delta\mathcal{L}_k(s) < \sigma\Delta\widehat{q_k},$$

where $\psi \in (\frac{1}{2}, 1)$ is a constant; and (3) the restoration phase is also entered if the multiplier weigh inactive constraints too strongly, which happens, if $\theta_k^{1/2} \leq \kappa_\rho \rho^{1+\xi}$, for $\kappa_\rho > 0$ and $\xi \in (0,1)$. Under a linear independence constraint qualification and second-order sufficient condition, Ulbrich is able to show q-quadratic convergence.

Wächter and Biegler [29] analyze a filter method with SOC steps. SOC steps solve a second QP that captures constraint curvature and is often cheap to solve, requiring, for example, only a shift in the QP constraints. Like [27], Wächter and Biegler also modify the switching condition and strengthen it to

$$\nabla f_k^T s_k < 0 \;\text{ and }\; \alpha_{k,l}(-\nabla f_k^T s_k)^{s_f} > \delta(h_k)^{s_h},$$

where $\alpha_{k,l}$ is the Armijo step size and $\delta > 0, s_h > 1$, and $s_f \geq 2s_h$ are constants. Thus, sufficient reduction in the objective is checked less frequently than in [10]. The analysis shows that ultimately, the SQP step or the SOC step is acceptable to the filter implying superlinear convergence for different types of SOC steps.

We currently prefer to use SOC steps to obtain fast local convergence because this approach allows us to keep the original filter definition with $f(x)$, rather than the Lagrangian $\mathcal{L}(x,y)$. This approach also avoids the need for a multiplier function.

2.1.5 Other SQP Filter Methods

Fletcher et al. [6] (see also [5, Chapter 15.5]) analyze a trust-region SQP filter method that decomposes the SQP step into a normal and tangential step. The normal step attains feasibility for the linearized constraints of (2.1), and the tangential step reduces a quadratic objective beyond the Cauchy point while maintaining feasibility. The algorithm uses the envelope

$$h_{k+1} \leq \beta h_j \;\text{ or }\; f_{k+1} \leq f_j - \gamma h_j \;,\; \forall j \in \mathcal{F}_k,$$

and removes only entries whose *envelope* is dominated by a new entry. The algorithm also uses a stronger switching condition, namely, $\Delta q_k \geq \sigma h_k^2$, resulting in fewer f-type steps. We note that the algorithm in [6] removes the need for a global solution of the QP.

Recently, there has been renewed interest in trust-region methods that avoid the solution of the computationally expensive QP (2.1). One such method is SLP-EQP, which dates back to [12] in the context of ℓ_1-penalty functions. The method solves an LP inside a trust-region to obtain an estimate of the active set (i.e., setting $H_k = 0$ in (2.1)). This active set is then explored further by solving an equality-constrained QP corresponding to the active constraints (with the trust-region bounds removed). Chin and Fletcher [4] (see also [3]) analyze and implement a filter SLP-EQP method. Their convergence proof adapts the proof in [11] to allow for a finite set of possible steps, namely, a Cauchy step (along the LP solution to the first minimum of the quadratic), an EQP step, and an SOC step.

Gonzaga et al. [14] propose a general framework for filter methods where the step computation is decomposed into a normal and tangential step. Unlike [6], however, where the QP solution is decomposed, Gonzaga et al. enforce filter conditions on both the normal and tangential step. The normal step must generate an intermediate point $x_{k+1/2}$ such that the constraint violation is acceptable to the current point: $h(x_{k+1/2}) < \beta h_k$. The tangential step must generate a new iterate that reduces the objective function by an amount that is proportional to the *filter slack*:

$$H_k := \min\,(1, \min_{j \in \mathcal{F}_k : f_j \le f(x_k)} h_j).$$

The step s satisfies $\nabla c_{k+1/2}^T s + c_{k+1/2} \ge 0$, and the new point, $x_{k+1} = x_{k+1/2} + s$, satisfies the following decrease condition:

$$f(x_{k+1}) \le f(x_{k+1/2}) - M\sqrt{H_k}.$$

The authors show that such a step can be computed by minimizing a quadratic model beyond the Cauchy point within a trust-region framework. In addition, a sufficient decrease condition is also enforced. This framework is very general, but the step acceptance seems slightly more restrictive.

Ribeiro et al. [22] extend the analysis in [14] by developing a general global convergence analysis of filter methods that does not depend on the particular way in which the step is computed. Instead, the authors prove convergence under fairly general assumptions that are shown to hold, for example, for SQP methods.

Finally, a *nonmonotone filter method* based on [6] is analyzed in [16]. The authors measure the area that a new entry contributes to the dominated region. This area is positive for monotone filters. The key idea is to request that this area be positive on average only over the last K reference iterations. This strategy allows the filter to accept points that would otherwise be rejected.

2.2 Filter Interior Methods

Interior-point methods (IPMs) are an attractive alternative to SQP methods for solving NLPs. Instead of computing a step by solving a QP, which can be computationally demanding, IPMs compute a step by solving a linear system. Thus, it is not surprising that researchers have extended filter methods to IPMs: Ulbrich et al. [26] and Wächter and Biegler [30] develop convergence theory for IPM filter methods. A related filter criterion has also been used by Benson et al. [2].

Interior-point methods first reformulate the NLP (1.1) so that the inequalities are simple bound constraints:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) = 0, \quad x \ge 0. \end{array} \tag{2.4}$$

IPMs can be viewed as applying Newton's method to the perturbed optimality conditions of (2.4):

$$F_\mu(x,y,z) \;:=\; \begin{pmatrix} \nabla_x \mathcal{L}(x,y,z) \\ c(x) \\ Xz - \mu e \end{pmatrix} = 0, \tag{2.5}$$

for a decreasing sequence of barrier parameters $\mu \searrow 0$, where X is the diagonal matrix with x along its diagonal, $\mathcal{L}(x,y,z) = f(x) - y^T c(x) - z^T x$ is the Lagrangian of (2.4), and $e = (1,\ldots,1)^T$. The IPM filter methods differ significantly in how the filter is employed to achieve global convergence.

2.2.1 The Interior Filter of Ulbrich et al.

Ulbrich et al. [26] employ the filter to enforce convergence of the IPM as $\mu \searrow 0$. They decompose the perturbed optimality conditions into a normal (r^n) and tangential (r^t) component,

$$\begin{aligned} F_{\sigma\mu'}(x,y,z) &= r^n + r^t \\ &= \begin{pmatrix} 0 \\ c(x) \\ Xz - \mu' e \end{pmatrix} + \begin{pmatrix} \nabla_x \mathcal{L}(x,y,z) \\ 0 \\ (1-\sigma)\mu' e \end{pmatrix}, \end{aligned}$$

where $\mu = \mu'\sigma$, and $\mu' = x^T z/n$, and $\sigma \in (0,1)$ is a centering parameter. This decomposition motivates the two filter components and a consistent step decomposition. Denoting $w = (x,y,z)$, the filter is defined a a collection of pairs of a *measure of quasicentrality*

$$\theta(w) := \|c(x)\| + \|Xz - x^T z/n \cdot e\|$$

and a *measure of optimality*

$$\theta_g(w) := \|\nabla_x \mathcal{L}(w)\| + x^T z/n.$$

Each step $s = (s_x, s_y, s_z)$ is computed from a normal and tangential step,

$$F'(w)s^n = -r^n, \; F'(w)s^t = -r^t,$$

where $F'(w)$ is the Jacobian of $F_{\sigma\mu'}(w)$. The authors exploit the flexibility of choosing different step sizes for each component. Once a step has been computed, the algorithm performs a backtracking line search until a filter-acceptable point has been found. Similar to SQP filter methods, the algorithm also enforces a sufficient decrease condition on a quadratic model of the residual of $\theta_g(w)$. If no acceptable point can be found, then a restoration phase is entered to restore quasicentrality.

Under the strong assumption that the inverse of the Jacobian is bounded, namely, $\|[F'(w)]^{-1}\| \leq C$, the authors show finite termination of the restoration phase and the existence of a subsequence converging to a stationary point. The step decomposition has a similar flavor to [6], but the two components have a slightly different interpretation, with quasicentrality replacing feasibility and the optimality measure $\theta_g(w)$ replacing the objective. The latter condition means that the algorithm may be more likely to converge to stationary points that are not local minimizers. The IPM of Wächter and Biegler avoids this problem by taking a different approach.

2.2.2 The Filter of Wächter and Biegler

Wächter and Biegler [30] have successfully incorporated a filter mechanism in the NLP solver IPOPT [31]. They develop a line-search filter method that avoids the pitfall of many IPMs that may converge to spurious stationary points illustrated by the example in [28]. Wächter and Biegler exploit the relationship between (2.5) and the barrier problem

$$\begin{aligned} \min_x \quad & \varphi_{\mu_k}(x) := f(x) - \mu_k \sum \ln(x_i) \\ \text{s.t.} \quad & c(x) = 0. \end{aligned} \tag{2.6}$$

IPOPT performs a number of line-search SQP iterations to minimize (2.6) to within a tolerance $\epsilon_k \searrow 0$, whilst keeping $x_i > 0$. In contrast to [26], where the filter safeguards the convergence

of IPM as $\mu \searrow 0$, IPOPT employs the filter only for *fixed* μ_k to ensure convergence of the SQP algorithm. This approach is justified because it can be shown that for a suitable choice of the sequences $\mu_k \searrow 0$, $\epsilon_k \searrow 0$ one SQP iteration and an extrapolation step are sufficient to generate an acceptable point near a solution.

A consequence of employing the filter for a fixed barrier parameter is that we can now use $(h(x), \varphi(x))$ again in the filter. Hence, the method is less likely to converge to stationary points that are not minimizers.

Another important difference from [26] is the absence of a full-rank assumption, which provides robustness for degenerate and infeasible NLPs. As a consequence, however, we must modify the Armijo line-search because a poor step may never be acceptable no matter how small a step size is chosen. Therefore, [30] derives a lower bound that indicates when the algorithm should switch to a restoration phase. The restoration algorithm in [30] differs from the SQP restoration algorithms in the sense that it must also produce a new point $x_{k+1} \geq \epsilon e$ that is strictly feasible with respect to the bounds.

The analysis in [30] is general and includes as special cases SQP methods, IPMs, and *augmented Lagrangian methods*. The augmented Lagrangian is another popular penalty function:

$$\mathcal{L}_\pi(x, y) := f(x) - y^T c(x) + \frac{\pi}{2} c(x)^T c(x).$$

We can split this function into two "objectives" similar to the way we split the exact penalty function. This motivates a filter method where the Lagrangian $\mathcal{L}(x, y)$ replaces $f(x)$. The analysis is readily extended by including a line search on the multipliers y and by modifying the switching condition in an obvious way. We note, that this is similar to the filter in [27]

Benson et al. [2] have also included a filter-like mechanism in LOQO. The filter used in LOQO consists of a *single entry.* We are not sure that this device alone can guarantee convergence. The practical performance of LOQO has been encouraging, however, underlining the computational advantage of filter methods.

3 Filters beyond NLP

Filter methods have been extended to other areas of optimization such as nonlinear equations and inequalities [9, 15, 17], nonsmooth optimization [7, 19, 21], unconstrained optimization [18], derivative-free optimization [1], and augmented Lagrangian methods [13].

3.1 Nonlinear Equations

We have developed a filter SQP method for solving a nonlinear system of inequalities $c(x) \geq 0$ in [9], similar to the restoration phase suggested in [8]. Formulating $c(x) \geq 0$ as a norm minimization problem,

$$\underset{x}{\text{minimize}} \;\; h(x) := \|c(x)^-\|, \tag{3.1}$$

allows us to define two objectives and apply the filter concept. We divide the constraints into two sets indexed by J and its complement $J^\perp$: the set $J^\perp$ collects the constraints that are close to being satisfied, and the set J collects the constraints that are difficult to satisfy. This partition

gives rise to the following feasibility problem,

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \sum_{i \in J} c_i(x)^- \\ \text{subject to} & c_i(x) \geq 0 \; \forall i \in J^\perp, \end{array} \tag{3.2}$$

which can be interpreted as a weighted ℓ_1 constraint residual minimization. The sets J and $J^\perp$ can be chosen adaptively as long as we ensure that

$$J \subset \{i | c_i(x) < 0\}.$$

Motivated by the feasibility problem (3.2), we define the two filter entries as

$$f_J(x) := \sum_{i \in J} c_i(x)^- \quad \text{and} \quad h_J(x) := \sum_{i \in J^\perp} c_i(x)^-,$$

respectively. We apply an SQP method to the minimization of (3.2) that enables us to achieve fast local convergence even if no feasible solution exists.

So far, the filter methods have been concerned with two competing aims. However, filter algorithms can also be developed for more objectives. In [15], we develop a multi-dimensional filter for the solution of $c(x) = 0$. The idea is to split the constraint residuals into p components

$$h_j(x) := \|c_{I_j}(x)\|, \; j = 1, \ldots, p,$$

where $\{1, \ldots, m\} = I_1 \cup \ldots \cup I_p$. We adapt the filter-acceptability by saying that a trial point x_k^+ is acceptable if and only if, $\forall l \in \mathcal{F}_k$,

$$\exists j \in \{1, \ldots, p\} : h_j(x_k^+) < h_j(x_l) - \beta \|h(x_l)\|,$$

where $\beta \in (0, 1/\sqrt{p})$ ensures that the right-hand side of this condition always has at least one positive entry, which ensures that we can always generate a filter-acceptable point. The algorithm minimizes a Gauss-Newton or, alternatively, a Newton-model of the least-squares formulation of $c(x) = 0$:

$$\underset{x}{\text{minimize}} \; f(x) := \frac{1}{2} \|h(x)\|_2^2.$$

The trust region is enforced only if a trial point is *not* filter-acceptable. The resulting algorithm is very nonmonotone and works best if we choose $p = m$. We extend this work in [17] to general feasibility problems such as (3.1) by defining $h_j(x) := \|c_{I_j}(x)^-\|, \; j = 1, \ldots, p$.

This multidimensional filter is also extended to unconstrained minimization in [18] by casting the minimization of $f(x)$ as the solution of the system $\nabla f(x) = 0$. The algorithm contains provisions for negative curvature and is shown to be convergent to second-order critical points. We generalize this algorithm to bound-constrained optimization in [24]. In related work, Sainvitu [23] studies the effect of using approximate derivatives within a filter method.

3.2 Nonsmooth Optimization

Filter methods for nonsmooth optimization provide a convenient extension of bundle methods to include nonsmooth constraints. We can assume without loss of generality, that the nonsmooth

NLP has only a single constraint $c(x) \in \mathbb{R}$, because we can reformulate multiple constraints as a single constraint using the max-function. In [7], we present a straightforward extension of of filter methods to bundle trust-region methods. We use two bundles (one for the objective, and one for the constraints) and solve an LP inside a trust region for a step. The convergence analysis is an extension of the SLP convergence proof in [11].

In contrast, the filter method of Karas et al. [19] combines ideas from proximal point methods and filter methods. The authors create a cutting plane model of the *improvement function*

$$g_x(y) := \max\{f(y) - f(x), -c(y)\}.$$

This function allows standard unconstrained proximal point methods to be used and requires only a single bundle to be maintained. The authors establish convergence to stationary points and present encouraging numerical results.

A recent variable-metric filter method is presented in [21].

3.3 Derivative-Free Optimization

Audet and Dennis [1] incorporate filter into a pattern-search method for derivative-free constrained optimization. Pattern-search methods target "black-box" applications, where the problem functions $f(x)$ and $c(x)$ are available only as oracles, and derivative information is prohibitive to obtain. The filter in [1] differs in three important aspects from the filters described above: (1) it requires only simple decrease similar to unconstrained pattern-search algorithms, (2) the incumbent (POLL center) is either feasible or the least infeasible iterate, and (3) the filter includes an entry $(0, f_F)$ corresponding to a feasible iterate. A new point x_k^+ is acceptable if either of the following two conditions hold:

$$h(x_k^+) = 0 \;\text{ and }\; f(x_k^+) < f_F$$

or

$$h(x_k^+) < h_l \;\text{ or }\; f(x_k^+) < f_l,\ \forall l \in \mathcal{F}_k.$$

The authors extend the usual patter-search convergence results to filter methods.

3.4 Augmented Lagrangian

An augmented Lagrangian filter method for QPs is developed in [13]. The algorithm efficiently accommodates matrix-free implementation and is based on two main phases. First, gradient projection iterations approximately minimize the augmented Lagrangian function and provide an estimate of the optimal active set. Second, an equality-constrained QP is approximately minimized on this subspace in order to generate a second-order search direction.

The iterations of augmented Lagrangian methods typically are controlled by two fundamental forcing sequences that ensure convergence to a solution. A decreasing sequence $\omega_k \searrow 0$ determines the required optimality of each subproblem solution and controls the convergence of the dual infeasibility. The second decreasing sequence, $\eta_k \searrow 0$, tracks the primal infeasibility $\|Ax - b\|$ and determines whether the penalty parameter ρ_k is increased or left unchanged.

In the definition of our filter we use quantities that are analogous to ω_k and η_k. Define

$$\begin{aligned} h(x,y) &= \|\min(x, \nabla_x \mathcal{L}(x,y))\|, \\ f(x) &= \|Ax - b\|, \end{aligned}$$

which are based on the optimality and feasibility of a current pair (x, y). The axis in this filter appear to be the reverse of the usual definition ($f(x)$ measures feasibility). This choice reflects the dual view of the augmented Lagrangian: it can be shown that $Ax_k - b$ is a steepest descent direction for the augmented Lagrangian. We use the filter, rather than the usual forcing sequences, to terminate the inner iteration (minimization of the augmented Lagrangian).

4 Conclusions

We have presented filter methods that promote convergence for constrained optimization algorithms without the need of artificial penalty parameters. Filter methods are an alternative to penalty function methods and build on the concept of domination from multiobjective optimization. Filter methods were initially designed for nonlinear programming problems but have quickly become popular in other areas such as nonlinear equations, nonsmooth optimization, and derivative-free methods. We believe that filter methods will continue to grow and find application in more diverse areas.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

References

[1] C. Audet and J.E. Dennis Jr. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 14(4):980–1010, 2004.

[2] H.Y. Benson, D.F. Shanno, and R.J. Vanderbei. Interior-point methods for nonconvex nonlinear programming: Filter-methods and merit functions. *Computational Optimization and Applications*, 23(2):257–272, 2002.

[3] C.M. Chin. A global convergence theory of a filter line search method for nonlinear programming. Technical report, Department of Electrical Engineering, University of Malaya, Kuala Lumpur, Malaysia, August 2002.

[4] C.M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177, 2003.

[5] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

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

[7] R. Fletcher and S. Leyffer. A bundle filter method for nonsmooth nonlinear optimization. Numerical Analysis Report NA/195, Dundee University, April 1999.

[8] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270, 2002.

[9] R. Fletcher and S. Leyffer. Filter-type algorithms for solving systems of algebraic equations and inequalities. In G. di Pillo and A. Murli, editors, *High Performance Algorithms and Software for Nonlinear Optimization*, pages 259–278. Kluwer, 2003.

[10] R. Fletcher, S. Leyffer, and Ph. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optimization*, 13(1):44–59, 2002.

[11] R. Fletcher, S. Leyffer, and Ph.L. Toint. On the global convergence of an SLP-filter algorithm. Numerical Analysis Report NA/183, University of Dundee, UK, August 1998.

[12] R. Fletcher and E. Sainz de la Maza. Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, 43:235–256, 1989.

[13] M.P. Friedlander and S. Leyffer. Gradient projection for general quadratic programs. Preprint ANL/MCS-P1370-0906, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL, 2006.

[14] C.C. Gonzaga, E. Karas, and M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM Journal on Optimization*, 14(3):646–669, 2003.

[15] N. I. M. Gould, S. Leyffer, and Ph. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least squares. *SIAM Journal on Optimization*, 15(1):17–38, 2004.

[16] N. I. M. Gould and Ph.L. Toint. Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming. Numerical Analysis Report RAL-TR-2003-003, Rutherford Appleton Laboratory, UK, 2003.

[17] Nicholas I. M. Gould and Philippe L. Toint. FILTRANE, a Fortran 95 filter-trust-region package for solving nonlinear least-squares and nonlinear feasibility problems. *ACM Transactions on Mathematical Software*, 2007.

[18] N.I.M. Gould and Ph. L. Toint C. Sainvitu. A filter-trust-region method for unconstrained optimization. *SIAM Journal on Optimization*, 16(2):341–357, 2006.

[19] E. Karas, A. Ribeiro, C. Sagastizabal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. Technical report, Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, 2005.

[20] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69(1):111–147, 1995.

[21] Yehui Peng and Zhenhai Liu. A filter-variable-metric method for nonsmooth convex constrained optimization. Technical report, School of Mathematical Sciences and Computing Technology, Central South University, Changsha, Hunan, P. R. China, 2006.

[22] A.A. Ribeiro, E.W. Karas, and C.C. Gonzaga. Global convergence of filter methods for nonlinear programming. Technical report, Department of Mathematics, Federal University of Paraná, Brazil, 2006.

[23] C. Sainvitu. A numerical investigation of the influence of approximate derivatives on the filter-trust-region method for unconstrained optimization. Technical Report TR06/03, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium, 2006.

[24] C. Sainvitu and Ph. L. Toint. A filter-trust-region method for simple-bound constrained optimization. Technical Report TR05/06, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium, 2006.

[25] P. D. Surry, N. J. Radcliffe, and I. D. Boyd. A multi-objective approach to constrained optimization of gas supply networks: The COMOGA method. In T. C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, number 993 in Lecture Notes in Computer Science, pages 166–180, Berlin, Springer Verlag, 1995.

[26] M. Ulbrich, S. Ulbrich, and L.N. Vicente. A globally convergent primal-dual interior-point filter method for nonconvex nonlinear programming. *Mathematical Programming*, 100:379–410, 2003.

[27] S. Ulbrich. On the superlinear local convergence of a filter-SQP method. *Mathematical Programming, Series B*, 100:217–245, 2004.

[28] A. Wächter and L.T. Biegler. Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming*, 88(3):565–574, 2000.

[29] A. Wächter and L.T. Biegler. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 16(1):32–48, 2005.

[30] A. Wächter and L.T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2005.

[31] A. Wächter and L.T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):22–57, 2006.