On the Global Convergence of an SLP–Filter Algorithm

by R. Fletcher[1], S. Leyffer[1] and Ph.L. Toint[2]

Report 98/13 September 21, 1998

[1] Department of Mathematics,
University of Dundee,
Dundee DD1 4HN, Scotland, UK.

[2] Department of Mathematics, Facultés Universitaires ND de la Paix, 61, rue de Bruxelles,
B-5000 Namur, Belgium, EU
Email: pht@math.fundp.ac.be

Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be (internet 138.48.20.102)

Also issued as Numerical Analysis Report NA/183,
Department of Mathematics, University of Dundee, UK.

On the Global Convergence of an SLP–Filter Algorithm

R. Fletcher and S. Leyffer and Ph. L. Toint

September 21, 1998

Abstract

A mechanism for proving global convergence in filter–type methods for nonlinear programming is described. Such methods are characterized by their use of the dominance concept of multiobjective optimization, instead of a penalty parameter whose adjustment can be problematic. The main point of interest is to demonstrate how convergence for NLP can be induced without forcing sufficient descent in a penalty-type merit function.

The proof technique is presented in a fairly basic context, but the ideas involved are likely to be more widely applicable. The technique allows a range of specific algorithm choices associated with updating the trust region radius and with feasibility restoration.

Keywords: nonlinear programming, global convergence, filter, SLP.

1 Introduction

In Fletcher and Leyffer [2] a new technique for globalizing methods for nonlinear programming (NLP) is presented. The idea is referred to as an NLP filter and is motivated by the aim of avoiding the need to choose penalty parameters, such as would occur with the use of l_1 penalty functions or augmented Lagrangian functions. Numerical experience with the technique in a Sequential Quadratic Programming (SQP) trust region algorithm is reported in [2] and is very promising. However, no global convergence proof is given in [2], although a number of heuristics are suggested to eliminate obvious situations in which the method might fail to converge.

This paper shows that the filter technique does provide a mechanism for forcing global convergence when used in an appropriate way. To present this result we consider a more basic situation in which a Sequential Linear Programming (SLP) algorithm is used in conjunction with a trust region and filter. A proof of global convergence is presented when the algorithm is used to solve NLP problems with just inequality constraints. It is hoped that this proof mechanism can be carried over to an SQP context, although there are a number of difficulties to be circumvented which are discussed further in Section 4. However, because our global result concerns first order conditions, it seems likely that little is lost by considering the SLP algorithm. The extension to include equality constraints should be straightforward, albeit at the expense of technical complications which would obscure the presentation.

The proposed algorithm contains an inner iteration for calculating a suitable trust region radius. In some ways this resembles the use of a backtracking line search. The motivation for this is that it provides certain conditions used in the convergence proof. To a large extent however, the approach allows conventional ideas of halving or doubling (say) the previous trust region radius to be used.

An interesting feature of the proof is that various of the heuristics used in [2] are shown to be unnecessary. These include the NW corner rule, the need to unblock the filter in some cases, and the consequent decision to reduce the strict upper bound on constraint infeasibility. In this paper we do use a feasibility restoration technique, but are not prescriptive as to how this is done. Another feature of some interest is that some points may be accepted by the algorithm, without a new entry in the filter being made. This contributes to the non-monotonic properties of the algorithm.

2 An SLP–filter Algorithm

In this paper we consider an NLP problem of the form

$$P \left\{ \begin{array}{ll} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & c_i(\mathbf{x}) \le 0 \quad i = 1, 2, \ldots, m, \end{array} \right.$$

and refer to a local solution by $\mathbf{x}^*$. We assume for the purposes of our convergence proof that the constraint set includes some linear constraints (for example simple upper and lower bounds on $\mathbf{x}$) that define a non–empty bounded region X. The LP subproblem in our algorithm depends upon the value of the current iterate $\mathbf{x}$ and trust region radius ρ, $(\rho > 0)$, and is defined by

$$LP(\mathbf{x}, \rho) \left\{ \begin{array}{ll} \underset{\mathbf{d} \in \mathbb{R}^n}{\text{minimize}} & \mathbf{g}^T\mathbf{d} \\ \text{subject to} & c_i + \mathbf{a}_i^T\mathbf{d} \le 0 \quad i = 1, 2, \ldots, m \\ & \|\mathbf{d}\|_\infty \le \rho, \end{array} \right.$$

where we denote $\mathbf{g} = \nabla f(\mathbf{x})$, $c_i = c_i(\mathbf{x})$ and $\mathbf{a}_i = \nabla c_i(\mathbf{x})$. Let $\mathbf{d}$ denote the solution (if it exists) of $LP(\mathbf{x}, \rho)$. Then we denote

$$\Delta l = -\mathbf{g}^T\mathbf{d}$$

as the *predicted reduction* in $f(\mathbf{x})$ and

$$\Delta f = f(\mathbf{x}) - f(\mathbf{x} + \mathbf{d})$$

as the *actual reduction* in $f(\mathbf{x})$. The measure of constraint infeasibility which we use in this paper is

$$h(\mathbf{c}) = \max(0, \max(c_i, \ i = 1, 2, \ldots, m))$$

and we note that this is different from the measure used in [2].

We now turn to the definition of an NLP filter as introduced in [2]. The two aims in an NLP problem are to minimize $f(\mathbf{x})$, and to satisfy the constraints, that is to minimize $h(\mathbf{c}(\mathbf{x}))$. In a filter we consider pairs of values (h, f) obtained by evaluating $h(\mathbf{c}(\mathbf{x}))$ and $f(\mathbf{x})$ for some $\mathbf{x}$. A pair (h_i, f_i) is said to *dominate* another pair (h_j, f_j) if and only if both $h_i \le h_j$ and $f_i \le f_j$, indicating that the former point is at least as good as the latter in respect of both measures. The NLP filter is defined to be a list of pairs (h_i, f_i) such that no pair dominates any other. This is illustrated in Figure 1. A point $\mathbf{x}$ is said to be acceptable for inclusion in the filter if it is not dominated by any point in the filter. If a point $\mathbf{x}$ is included in the filter, its (h, f) pair is added to the list of pairs, and any points in the filter that are dominated by the new point are removed. We use the filter as an alternative to a penalty function as a means of deciding whether or not to accept a new point in an NLP algorithm.

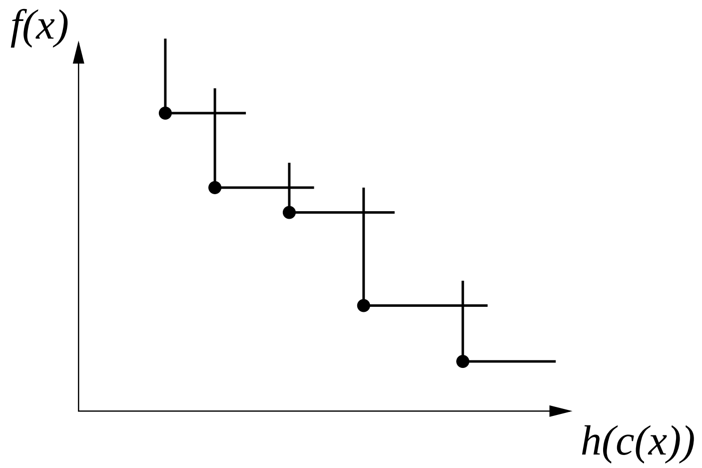

Figure 1: An NLP Filter

In fact this definition of a filter is not adequate for proving convergence as it allows points to accumulate in the neighbourhood of a filter entry that has $h_i > 0$. This is readily corrected by defining a small envelope around the current filter in which points are not accepted. Thus the condition for a point being acceptable to the filter is that its (h, f) pair satisfies

$$\text{either} \qquad h \leq \beta h_i \qquad \text{or} \qquad f \leq f_i - \gamma h_i \tag{2.1}$$

for all $i \in \mathcal{F}$, where $\mathcal{F}$ denotes the current set of filter entries, and β and γ are preset parameters such that $1 > \beta > \gamma > 0$, with β close to 1 and γ close to zero. It is also convenient to allow an upper bound

$$h(\mathbf{c}(\mathbf{x})) \leq u \tag{2.2}$$

on constraint infeasibility, and this is readily implemented by initializing the filter with the entry $(u, -\infty)$.

A common feature in a trust region algorithm for unconstrained minimization is the use of a sufficient reduction criterion

$$\Delta f \geq \sigma \Delta l, \tag{2.3}$$

where Δl is positive, and $\sigma \in (\gamma, 1)$ is a preset parameter. However, in an NLP algorithm, Δl may be negative or even zero, in which case this test is no longer appropriate. A new feature of the algorithm in this paper is that it uses (2.3) only when Δl is sufficiently positive, which we shall measure by the inequality

$$\Delta l \geq \delta . h^2, \tag{2.4}$$

where h refers to $h(\mathbf{c}(\mathbf{x}))$ evaluated at the current point, and $\delta > 0$ is a preset parameter.

We are now in a position to state our SLP–filter algorithm, which we do by means of the flow diagram of Figure 2. The sequence of points accepted by the algorithm is referred to by $\{\mathbf{x}^{(k)}\}$, and quantities derived from $\mathbf{x}^{(k)}$ are superscripted in a similar manner, for example $h^{(k)}$ refers to $h(\mathbf{c}(\mathbf{x}^{(k)}))$ and $f^{(k)}$ to $f(\mathbf{x}^{(k)})$. Also $\mathcal{F}^{(k)}$ refers to the set of filter entries at the start of iteration k. It can be seen in Figure 2 that there is an inner loop in which the trust region radius

ρ is successively reduced until either certain tests are satisfied, or the current LP subproblem becomes incompatible (for clarity we avoid the use of the word 'infeasible' in this context). The inner loop is initialized with any value of $\rho \geq \rho^\circ$, where $\rho^\circ > 0$ is a preset parameter. We may assume without loss of generality that $\rho^\circ \leq |X|$ (the diameter of the region X defined by the set of linear constraints). The inner loop chooses a decreasing geometric sequence of values of ρ and generates corresponding values of $\mathbf{d}$, Δl and Δf (unsubscripted). When the inner iteration terminates, the current values of ρ, $\mathbf{d}$, Δl and Δf are denoted respectively by $\rho^{(k)}$, $\mathbf{d}^{(k)}$, $\Delta l^{(k)}$ and $\Delta f^{(k)}$. We observe that all points that are generated by the algorithm lie in the region X.

This algorithm differs in one important respect from that in [2] in that not all points $\mathbf{x}^{(k)}$ are included in the filter, even though they are acceptable to the filter. If an iteration satisfies both

$$\Delta f^{(k)} \geq \sigma \Delta l^{(k)} \qquad \text{and} \qquad \Delta l^{(k)} \geq \delta (h^{(k)})^2 \tag{2.5}$$

then it is deemed to be an *f–type iteration*, and otherwise an *h–type iteration*. Only points $\mathbf{x}^{(k)}$ that give rise to an h–type iteration are included in the filter. As a consequence, all the current filter entries have $h_i > 0$, $i \in \mathcal{F}^{(k)}$. This is because if $h^{(k)} = 0$ then $LP(\mathbf{x}^{(k)}, \rho)$ must be compatible and also $\Delta l^{(k)} \geq 0$ occurs so that the test $\Delta l^{(k)} \geq \delta(h^{(k)})^2$ is always satisfied. Thus the conditions for an h-type iteration (see Figure 2) are not satisfied and an entry with $h^{(k)} = 0$ cannot occur in the filter. It is convenient to denote

$$\tau^{(k)} = \min_{i \in \mathcal{F}^{(k)}} h_i > 0. \tag{2.6}$$

It can be seen that our algorithm includes the provision for a feasibility restoration phase if the current LP subproblem becomes incompatible. We do not prescribe how this should be done, but the intention is to use some sort of Newton-like scheme for minimizing $h(\mathbf{c}(\mathbf{x}))$, in order to get close to the boundary of the feasible region. The intention is that this phase should terminate with a point that is acceptable to the filter, and for which $LP(\mathbf{x}, \rho^\circ)$ is compatible. Of course, this may not always be possible and this phase might converge to a non–zero local minimum of $h(\mathbf{c}(\mathbf{x}))$, such as when the original problem P is incompatible. Otherwise the fact that $h_i > 0$, $i \in \mathcal{F}^{(k)}$ will usually be sufficient to ensure that the restoration phase terminates as intended. However this is not guaranteed: for example consider the pathological problem $\min (x_2 - 1)^2$ subject to $x_1^2 \geq 0$ and $-x_1^2 \geq 0$, starting from $\mathbf{x} = (1, 0)^T$. A Newton-like iteration is likely to converge to the feasible point $\mathbf{x} = \mathbf{0}$, which is not a solution of the NLP, without finding a point at which the LP subproblem is compatible. In this paper we content ourselves with the possibility that the restoration phase may not converge. Note however that the restoration phase makes no demands on the resulting value of $f(\mathbf{x})$, which could be significantly worse than that at the previous point. Nonetheless, and even if this happens an infinite number of times, the global convergence proof shows that this does not prevent the algorithm from accumulating at a point that satisfies first order necessary conditions.

3 A Global Convergence Proof

In this section we present a proof of global convergence of the SLP–filter algorithm of Figure 2 when applied to problem P. We assume only that the set X defined by the linear constraints is non–empty and bounded, and that the problem functions $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are twice continuously differentiable on X.

When the algorithm is applied, one of four different possible outcomes can occur, which we itemize as follows.

($\mathbb{A}$) The restoration phase converges infinitely and fails to find a point $\mathbf{x}$ which is acceptable to the filter and for which $LP(\mathbf{x}, \rho^\circ)$ is compatible.

($\mathbb{B}$) A Kuhn–Tucker point is found ($\mathbf{d} = \mathbf{0}$ solves $LP(\mathbf{x}^{(k)}, \rho)$ for some k).

($\mathbb{C}$) An infinite subsequence of consecutive f–type iterations occurs.

($\mathbb{D}$) An infinite subsequence of h–type iterations occurs on which the filter is augmented by $(h^{(k)}, f^{(k)})$.

Note in case ($\mathbb{C}$) that there are only a finite number of h–type iterations. Case ($\mathbb{C}$) typifies the situation in which convergence to an unconstrained stationary point takes place (or convergence to a KT point at which only linear constraints are active).

Our global convergence theorem concerns Fritz–John necessary conditions and we first review the theory associated with these (see for example Bazaraa and Shetty [1] for a simple exposition). Necessary conditions for $\mathbf{x}^*$ to solve P are that $\mathbf{x}^*$ is a feasible point, and that the set of directions

$$\{\mathbf{s} \mid \mathbf{s}^T\mathbf{g}^* < 0 \tag{3.1}$$
$$\mathbf{s}^T\mathbf{a}_i^* < 0 \quad i \in \mathcal{A}^*\} \tag{3.2}$$

is empty, where $\mathcal{A}^*$ denotes the set of active constraints at $\mathbf{x}^*$. These are the conditions that $\mathbf{s}$ is respectively a *descent direction* and an *interior feasible direction*. When $\mathbf{x}^*$ is a feasible point at which the set in (3.1) and (3.2) is empty, we shall refer to $\mathbf{x}^*$ as an *FJ point*. That the set in (3.1) and (3.2) is empty is equivalent to the existence of multipliers, but we do not need that result in our presentation.

What we do need for our theorem is that if $\mathbf{x}^\infty$ is a feasible point, but not an FJ point, then there exists an $\varepsilon > 0$ and a vector $\mathbf{s}$ such that $\|\mathbf{s}\|_\infty = 1$ for which

$$\mathbf{s}^T\mathbf{g}(\mathbf{x}) \leq -\varepsilon \tag{3.3}$$
$$\mathbf{s}^T\mathbf{a}_i(\mathbf{x}) \leq -\varepsilon \quad i \in \mathcal{A}^\infty \tag{3.4}$$

for all $\mathbf{x}$ in some neighbourhood $\mathcal{N}^\infty$ of $\mathbf{x}^\infty$. This result is a staightforward consequence of (3.1), (3.2) and continuity of the vectors $\mathbf{g}(\mathbf{x})$ and $\mathbf{a}_i(\mathbf{x})$. We are now in a position to prove global convergence of the SLP-filter algorithm.

The global convergence proof is divided into three parts. First, two useful results following from Taylor's theorem are stated in Lemma 1. Next, Lemma 2 shows that the inner iteration always terminates. Thus if ($\mathbb{A}$) or ($\mathbb{B}$) do not occur, then an infinite subsequence of either type ($\mathbb{C}$) or ($\mathbb{D}$) must occur. Finally, Theorem 1 establishes global convergence of the algorithm by showing that these subsequences converge to an FJ point. The standard assumptions that are made throughout are summarized as follows.

Standard Assumptions *The set X is non-empty and bounded and the functions $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are twice continuously differentiable on an open set containing X.*

Lemma 1 *Let the standard assumptions hold and let $M > 0$ be an upper bound on all the terms $\frac{1}{2}\mathbf{s}^T\nabla^2 f(\mathbf{x})\mathbf{s}$ and $\frac{1}{2}\mathbf{s}^T\nabla^2 c_i(\mathbf{x})\mathbf{s}$ for all $\mathbf{x} \in X$ and all vectors $\mathbf{s}$ such that $\|\mathbf{s}\|_\infty = 1$. Let $\mathbf{d} \neq \mathbf{0}$ solve $LP(\mathbf{x}^{(k)}, \rho)$. It then follows that*

$$c_i(\mathbf{x}^{(k)} + \mathbf{d}) \leq \rho^2 M \qquad i = 1, 2, \ldots, m, \tag{3.5}$$

and that

$$\Delta f \geq \Delta l - \rho^2 M. \tag{3.6}$$

Proof By Taylor's theorem and feasibility of $\mathbf{d}$ in $LP(\mathbf{x}^{(k)}, \rho)$,

$$c_i(\mathbf{x}^{(k)} + \mathbf{d}) = c_i^{(k)} + \mathbf{a}_i^{(k)T}\mathbf{d} + \tfrac{1}{2}\mathbf{d}^T\nabla^2 c_i(\mathbf{y}_i)\mathbf{d} \leq \rho^2 M \qquad i = 1, 2, \dots, m,$$

where $\mathbf{y}_i$ denotes some point on the line segment from $\mathbf{x}^{(k)}$ to $\mathbf{x}^{(k)} + \mathbf{d}$. Likewise

$$f(\mathbf{x}^{(k)} + \mathbf{d}) = f^{(k)} + \mathbf{g}^{(k)T}\mathbf{d} + \tfrac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{y})\mathbf{d}$$

and hence by definition of $\Delta l = -\mathbf{g}^{(k)T}\mathbf{d}$,

$$\Delta f = \Delta l - \tfrac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{y})\mathbf{d} \geq \Delta l - \rho^2 M.$$

q.e.d.

Next we examine what can happen for fixed $\mathbf{x}^{(k)}$ as ρ is reduced in the inner iteration, and we show that the inner iteration must terminate.

Lemma 2: *Let the standard assumptions hold, then the inner iteration terminates finitely.*

Proof Clearly, if the inner iteration does not terminate finitely then the rule for decreasing ρ ensures that $\rho \to 0$. Two cases need to be considered, depending on whether $h^{(k)} > 0$ or $h^{(k)} = 0$.

If $h^{(k)} > 0$ and i is an index for which $c_i^{(k)} = h^{(k)}$ then for all $\mathbf{d}$ such that $\|\mathbf{d}\|_\infty \leq \rho$ it readily follows that

$$c_i^{(k)} + \mathbf{a}_i^{(k)T}\mathbf{d} \geq c_i^{(k)} - \rho\|\mathbf{a}_i^{(k)}\|_1 > 0$$

if $\rho < c_i^{(k)}/\|\mathbf{a}_i^{(k)}\|_1$. Thus for sufficiently small ρ, constraint i cannot be satisfied and $LP(\mathbf{x}^{(k)}, \rho)$ is incompatible. Note that if $\|\mathbf{a}_i^{(k)}\|_1 = 0$, then the constraint i cannot be satisfied for any ρ. Thus the inner iteration terminates in this case.

If $h^{(k)} = 0$, then by a similar argument, inactive constraints at $\mathbf{x}^{(k)}$ are inactive at the solution of $LP(\mathbf{x}^{(k)}, \rho)$ for sufficiently small ρ. In this case $LP(\mathbf{x}^{(k)}, \rho)$ is equivalent to the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{g}^{(k)T}\mathbf{d} \\ \text{subject to} & \mathbf{a}_i^{(k)T}\mathbf{d} \leq 0 \qquad i \in \mathcal{A}^{(k)} \\ & \|\mathbf{d}\|_\infty \leq \rho, \end{array}$$

where $\mathcal{A}^{(k)}$ denotes the set of active constraints at $\mathbf{x}^{(k)}$. On dividing each expression by ρ we see that $\mathbf{d}$ solves this problem if and only if $\mathbf{s} = \mathbf{d}/\rho$ solves the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{g}^{(k)T}\mathbf{s} \\ \text{subject to} & \mathbf{a}_i^{(k)T}\mathbf{s} \leq 0 \qquad i \in \mathcal{A}^{(k)} \\ & \|\mathbf{s}\|_\infty \leq 1. \end{array}$$

Thus, for sufficiently small ρ, $\Delta l = \eta\rho$ where $\eta = -\mathbf{g}^{(k)T}\mathbf{s} > 0$ is independent of ρ. If $\rho \leq (1-\sigma)\eta/M$, it then follows from (3.6) that $\Delta f \geq \sigma\Delta l$. Also if $\rho^2 \leq \beta\tau^{(k)}/M$, it follows from (3.5) that $h(\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{d})) \leq \beta\tau^{(k)}$. Thus, for sufficiently small ρ, $\mathbf{x}^{(k)} + \mathbf{d}$ is acceptable to the filter, and the tests in (2.5) are both satisfied. Also, because $f(\mathbf{x}^{(k)} + \mathbf{d}) < f^{(k)}$, it follows that

$\mathbf{x}^{(k)} + \mathbf{d}$ is acceptable to $(h^{(k)}, f^{(k)})$. Thus the inner iteration terminates in this case also, with an f-type step. *q.e.d.*

A consequence of Lemma 2 is that, if the SLP-filter algorithm does not terminate with ($\mathbb{A}$) or ($\mathbb{B}$), then it generates an infinite subsequence either of type ($\mathbb{C}$) or ($\mathbb{D}$). The following theorem completes the global convergence proof by showing that this subsequence converges to an FJ point.

Theorem 1 *If the standard assumptions hold, then for the SLP–filter algorithm of Figure 2, either ($\mathbb{A}$) or ($\mathbb{B}$) occurs, or there exists an infinite subsequence, either of type ($\mathbb{C}$) or ($\mathbb{D}$). Any accumulation point of this subsequence satisfies Fritz-John conditions.*

Proof We need only consider the case in which neither ($\mathbb{A}$) nor ($\mathbb{B}$) occurs. The existence of an infinite subsequence of type ($\mathbb{C}$) or ($\mathbb{D}$) follows from Lemma 2. Let this subsequence have an accumulation point $\mathbf{x}^\infty$, and let $\{\mathbf{x}^{(k)}\}$, $k \in \mathcal{S}$ be any thinner subsequence converging to $\mathbf{x}^\infty$. We shall show that $\mathbf{x}^\infty$ is an FJ point. First we show that $\mathbf{x}^\infty$ is a feasible point.

In case ($\mathbb{C}$) we note that $f^{(k)}$ is monotonically decreasing and is bounded below from $\mathbf{x} \in X$, so we can deduce that $\sum_{k\in\mathcal{S}} \Delta f^{(k)}$ is bounded above. Also in case ($\mathbb{C}$), all iterations are of f-type, and it follows from (2.5) that

$$\Delta f^{(k)} \geq \sigma\delta(h^{(k)})^2.$$

Since $\sum_{k\in\mathcal{S}} \Delta f^{(k)}$ is bounded, it follows that $h^{(k)} \to 0$.

In case ($\mathbb{D}$), let k' and k'' be any two adjacent indices in $\mathcal{S}$. Since the coordinates $(h',\ f')$ are entered into the filter, any subsequent iterate must satisfy either $h \leq \beta h'$ or $f \leq f' - \gamma h'$. If $h^\infty > 0$, it follows for k' sufficiently large that $h'' > \beta h'$ and hence $f'' \leq f' - \gamma h'$. Taken over all $k \in \mathcal{S}$, this contradicts the fact that $f^{(k)}$ is bounded below. Thus, in either case ($\mathbb{C}$) or case ($\mathbb{D}$), it follows that $\mathbf{x}^\infty$ is a feasible point.

Now we consider the possibility that $\mathbf{x}^\infty$ is not an FJ point and show that this leads to a contradiction. Let $\mathbf{s}$ be the vector defined by (3.3) and (3.4) that exists for all $\mathbf{x} \in \mathcal{N}^\infty$. For sufficiently large k and any $\rho \leq |X|$, we consider the effect of a step $\rho\mathbf{s}$ in $LP(\mathbf{x}^{(k)}, \rho)$.

For active constraints at $\mathbf{x}^\infty$ we have from (3.4) that

$$c_i^{(k)} + \rho \mathbf{a}_i^{(k)T}\mathbf{s} \leq h^{(k)} - \rho\varepsilon \qquad i \in \mathcal{A}^\infty. \tag{3.7}$$

For inactive constraints $i \notin \mathcal{A}^\infty$, if $k \geq K$ is sufficiently large, then there exist positive constants $\bar{c}$ and $\bar{a}$, independent of k, such that

$$c_i^{(k)} \leq -\bar{c} \qquad \text{and} \qquad \mathbf{a}_i^{(k)T}\mathbf{s} \leq \bar{a},$$

by continuity of c_i and boundedness of $\mathbf{a}_i$ on X. It follows that

$$c_i^{(k)} + \rho \mathbf{a}_i^{(k)T}\mathbf{s} \leq -\bar{c} + \rho\bar{a} \qquad i \notin \mathcal{A}^\infty. \tag{3.8}$$

Denoting $\kappa = \bar{c}/\bar{a}$, it follows for $k \geq K$ that if

$$h^{(k)}/\varepsilon \leq \rho \leq \kappa, \tag{3.9}$$

then from (3.7) and (3.8),

$$c_i^{(k)} + \rho \mathbf{a}_i^{(k)T}\mathbf{s} \leq 0 \qquad i = 1, 2, \ldots, m.$$

Thus if (3.9) holds, we are assured that $\rho\mathbf{s}$ is a feasible step, and hence that $LP(\mathbf{x}^{(k)},\rho)$ is a compatible subproblem. It also follows by optimality of $\mathbf{d}$ that

$$\Delta l \geq -\rho\mathbf{g}^{(k)T}\mathbf{s} \geq \rho\varepsilon \tag{3.10}$$

from (3.3).

If $\rho^2 \leq \beta\tau^{(k)}/M$, it follows from (3.5) that $h(\mathbf{c}(\mathbf{x}^{(k)}+\mathbf{d})) \leq \beta\tau^{(k)}$. Also we deduce from (3.6) and (3.10) that

$$\frac{\Delta f}{\Delta l} \geq 1 - \frac{\rho^2 M}{\Delta l} \geq 1 - \frac{\rho M}{\varepsilon},$$

so if $\rho \leq (1-\sigma)\varepsilon/M$ it follows that $\Delta f \geq \sigma\Delta l$. Putting these results with (3.9), we see for sufficiently large k that if ρ satisfies

$$\frac{h^{(k)}}{\varepsilon} \leq \rho \leq \min\left\{\frac{(1-\sigma)\varepsilon}{M},\ \sqrt{\frac{\beta\tau^{(k)}}{M}},\ \kappa,\ |X|\right\} \tag{3.11}$$

then $LP(\mathbf{x}^{(k)},\rho)$ is compatible, $h(\mathbf{c}(\mathbf{x}^{(k)}+\mathbf{d})) \leq \beta\tau^{(k)}$ and $\Delta f \geq \sigma\Delta l$. Also from (3.10) and (3.11), $\Delta l \geq h^{(k)}$ and hence $\Delta l \geq \delta(h^{(k)})^2$. Moreover, it follows from $\Delta f \geq \sigma\Delta l$, $\Delta l \geq h^{(k)}$ and $\sigma \geq \gamma$ that $f(\mathbf{x}^{(k)}+\mathbf{d}) \leq f^{(k)} - \gamma h^{(k)}$. Thus for sufficiently large $k \in \mathcal{S}$ there is a range of values (3.11) that guarantee that $\mathbf{x}^{(k)}+\mathbf{d}$ is acceptable to the filter and to $(h^{(k)}, f^{(k)})$, and the conditions for an f-type step are satisfied.

If the subsequence $\mathcal{S}$ arises from case $(\mathbb{C})$, then $\tau^{(k)}$ is fixed and the right hand side of (3.11) is just a number, $\bar{\rho}$ say ($\bar{\rho} > 0$), whilst the left hand side converges to zero. Thus for sufficiently large k we can guarantee that a value $\rho^{(k)} > \frac{1}{2}\min(\bar{\rho},\,\rho^{\circ})$ will be chosen in the inner iteration. We then deduce from (3.10) that $\Delta f^{(k)} > \frac{1}{2}\sigma\varepsilon\min(\bar{\rho},\,\rho^{\circ})$ which contradicts the fact that $\sum_{k\in\mathcal{S}}\Delta f^{(k)}$ is bounded.

Finally we look at case $(\mathbb{D})$. Because $h^{(k)} \to 0$ there is an infinite subsequence of $\mathcal{S}$ for which $h^{(k)} = \tau^{(k)}$. On this subsequence, for sufficiently large k, the range (3.11) becomes

$$\frac{\tau^{(k)}}{\varepsilon} \leq \rho \leq \sqrt{\frac{\beta\tau^{(k)}}{M}}. \tag{3.12}$$

If $h^{(k)}$ and hence $\tau^{(k)}$ is sufficiently close to zero, the upper bound in this inequality is more than twice the lower bound, and both bounds converge to zero. Hence the technique of initializing $\rho \geq \rho^{\circ}$ and successively reducing ρ by a factor of two in the inner iteration will eventually locate a value in the range (3.12) which provides the conditions for an f–type step to occur. It is not possible for a larger value of ρ to produce an h–type step since Δl decreases monotonically as ρ decreases. Thus there exists a k sufficiently large for which an f–type step will be taken. This contradicts the fact that case $(\mathbb{D})$ is formed by a subsequence of h–type steps.

Thus a contradiction has been obtained in both case $(\mathbb{C})$ and case $(\mathbb{D})$ and the theorem is proved. *q.e.d.*

Remark In cases $(\mathbb{C})$ or $(\mathbb{D})$, a consequence of the theorem, by virtue of X being bounded, is that an accumulation point exists and convergence of a subsequence to a Fritz–John point takes place.

4 Discussion

Of course, the algorithm of Figure 2 is only a guide to what might be successfully implemented in practice, and is incomplete in various ways. For example, the rule for adjusting ρ in the inner iteration could be more intricate, based partly on interpolation. To some extent the algorithm of Figure 2 does allow the more usual trust region procedure in which one may double or halve (say) the value of ρ from the previous iteration. However this choice must be overriden by initializing $\rho = \rho^\circ$ if otherwise a lesser value would be chosen. The potential danger of just taking ρ from the previous iteration is that the existence of a successful f-type step may not be recognised in the limit of the ($\mathbb{D}$) subsequence. The algorithm of Figure 2 ensures that the upper bound in (3.12) exceeds the lower bound by a factor of 2 or more if FJ conditions are not satisfied in the limit. By starting with $\rho \geq \rho^\circ$, we thus ensure that an f-type step will be taken. Adjusting the trust region in this sort of way has featured in other recent work, see for example [3], [4] and references contained therein.

Other observations include the need to be more specific about how the restoration phase is implemented. Also the aim of the restoration phase can be relaxed by requiring that $LP(\mathbf{x}^{(k)}, \rho)$ is compatible, where $\rho \geq \rho^\circ$ is the value to be used subsequently to initialize the inner loop. It also may be appropriate to modify (2.4) in some way, for example replacing h by $\min(1, h)$ or something similar. Most important however is the need to include second order information, such as by using SQP steps rather than SLP steps. However, from the point of view of a convergence proof, this would introduce difficulties due to the possible existence of local minima in the QP subproblem, and the possibility of multiplier estimates becoming unbounded, which at present we cannot resolve, other than by assuming that these situations do not occur. Also it would be important to pay attention to the asymptotic behaviour of the algorithm to ensure that the second order convergence property of the SQP iteration is not compromised. It is not yet clear how best to do this. We hope to address these issues in our future work.

Nonetheless we feel that the result presented in this paper does show a mechanism by which filter methods can induce global convergence. The argument has two strands: first is that the filter plus envelope will ultimately force the iterates to become arbitrarily close to the boundary of the feasible region. Then, if the iterates are not close to an FJ point, f–type steps giving a significant reduction in $f(\mathbf{x})$ can be found. In the limit, these steps will either converge to an FJ point, or the points at which f-type steps can no longer be taken will themselves converge to an FJ point.

5 References

[1] Bazaraa M.S. and Shetty C.M. (1979), *NONLINEAR PROGRAMMING Theory and Algorithms*, John Wiley and Sons, New York.

[2] Fletcher R. and Leyffer S. (1997), Nonlinear Programming Without a Penalty Function, Dundee University, Dept. of Mathematics, Report NA/171.

[3] Jiang H., Fukushima M., Qi L. and Sun D. (1998), A Trust Region Method for Solving Generalized Complementarity Problems, *SIAM J. Optim.*, **8**, pp. 140–158.

[4] Kanzow Ch. and Zupke M. (1997), Inexact Trust-Region Methods for Nonlinear Complementarity Problems, University of Hamburg, Institute of Applied Mathematics, Report Number 127.

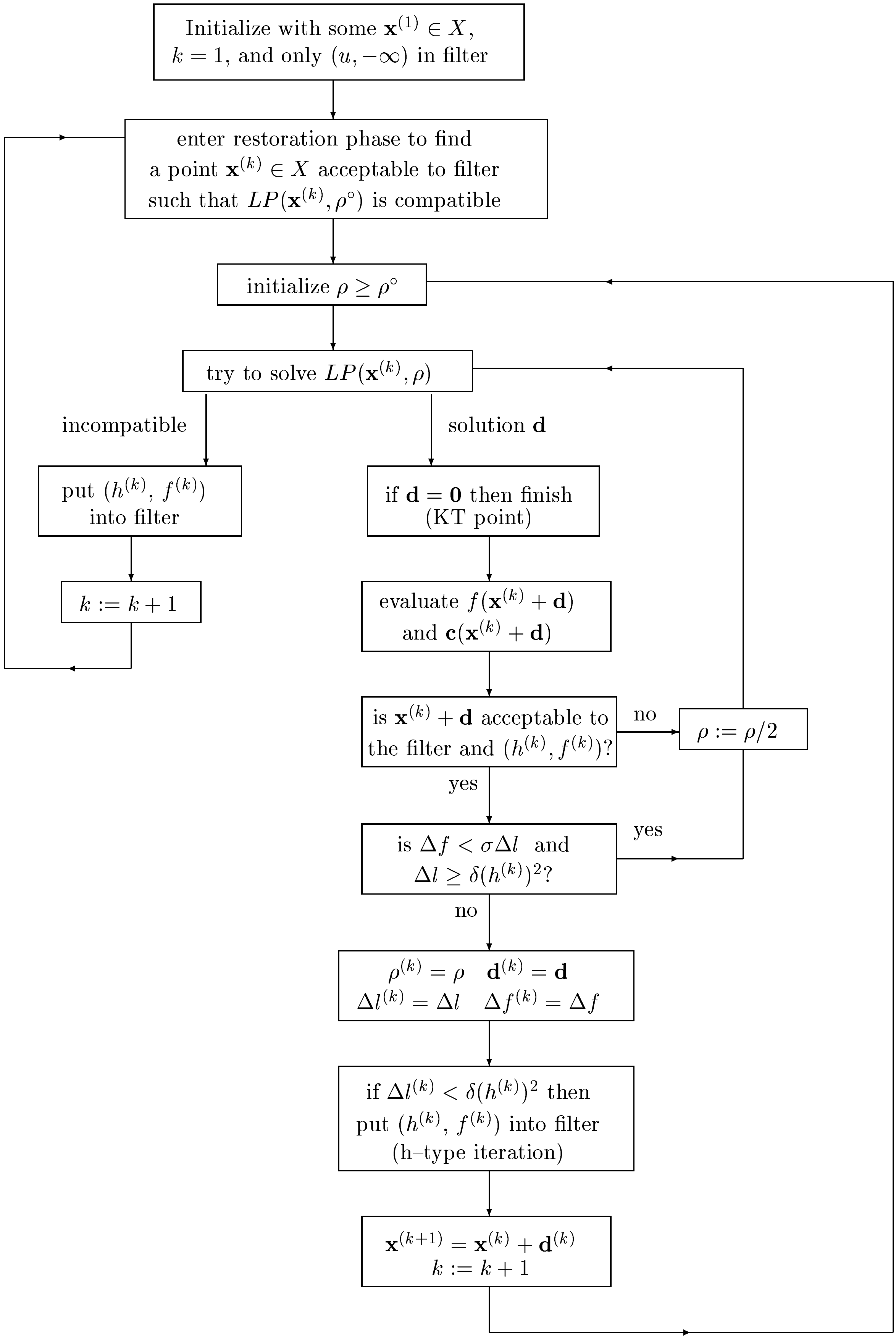

Figure 2: An SLP Filter Algorithm