A PATTERN SEARCH FILTER METHOD FOR NONLINEAR PROGRAMMING WITHOUT DERIVATIVES *

CHARLES AUDET [†] AND J.E. DENNIS JR. [‡]

Abstract. This paper formulates and analyzes a pattern search method for general constrained optimization based on filter methods for step acceptance. Roughly, a filter method accepts a step that either improves the objective function value or the value of some function that measures the constraint violation. The new algorithm does not compute or approximate any derivatives, penalty constants or Lagrange multipliers. A key feature of the new algorithm is that it preserves the useful division into global SEARCH and local POLL steps. It is shown here that the algorithm identifies limit points at which optimality conditions depend on local smoothness of the functions. Stronger optimality conditions are guaranteed for smoother functions. In the absence of general constraints, the proposed algorithm and its convergence analysis generalize the previous work on unconstrained, bound constrained and linearly constrained generalized pattern search. The algorithm is illustrated on some test examples and on an industrial wing planform engineering design application.

Key words. Pattern search algorithm, filter algorithm, surrogate-based optimization, derivative-free convergence analysis, constrained optimization, nonlinear programming.

AMS subject classifications. 90C30, 90C56, 65K05

1. Introduction. The optimization problem considered in this paper is

$$(1.1) \qquad \begin{aligned} \min_{x \in X} \quad & f(x) \\ \text{s.t.} \quad & C(x) \leq 0 \, , \end{aligned}$$

where $f : X \to \mathbb{R} \cup \{\infty\}$ and $C : X \to (\mathbb{R} \cup \{\infty\})^m$ are functions with $C = (c_1, \ldots, c_m)^T$, and X is a full dimensional polyhedron in $\mathbb{R}^n$ defined by finitely many nondegenerate explicit bound and linear constraints. It is possible, for instance when the functions are provided as "black box" subroutine calls, that some constraints might be linear without the knowledge of the user. In that case, these linear constraints are incorporated in $C(x) \leq 0$. The feasible region defined by the constraints $C(x) \leq 0$ is denoted by Ω. The proposed approach combines aspects of filter algorithms to handle Ω, and pattern search algorithms to handle X.

Filter algorithms were introduced by Fletcher and Leyffer [14] as a way to globalize sequential linear and quadratic programming (SQP and SLP) without using any merit function that would require a troublesome parameter to be provided by the user for weighting the relative merits of improving feasibility and optimality. A filter algorithm introduces a function that aggregates constraint violations and then treats the resulting biobjective problem. A step is accepted if it either reduces the value of the objective function or of the constraint violation. Although this clearly is less parameter dependent than a penalty function, still we acknowledge that specifying a constraint violation function implies assigning relative weights to reducing each constraint.

Fletcher *et al.* [15, 16] show convergence of the filter method that uses SQP or SLP to suggest steps. In both cases, convergence is to a point satisfying Fritz John optimality

†GERAD and Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada (www.gerad.ca/Charles.Audet, Charles.Audet@gerad.ca)

‡Computational and Applied Mathematics Department, Rice University - MS 134, 6100 South Main Street, Houston, Texas, 77005-1892 (www.caam.rice.edu/~dennis, dennis@caam.rice.edu)

			Form Approved OMB No. 0704-0188

Report Documentation Page

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **A Pattern Search Filter Method for Nonlinear Programming Without Derivatives**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rice University,Computational and Applied Mathematics Department,6100 South Main Street,Houston,TX,77005-1892**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **28**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

conditions. Thus, previous filter algorithms require explicit use of the derivatives of both the objective and the constraints. They also require more than simple decrease of the objective and constraint violation functions to accept a step. Furthermore, Fritz John optimality is weaker than the more common Karush-Kuhn-Tucker conditions, but, numerical results for the filter methods are very promising. It is well to remember that the theoretical justification for quasi-Newton methods trailed their efficacy by several years.

The generalized pattern search algorithm class (GPS) designed by Torczon [26] unifies a wide class of useful derivative-free algorithms for unconstrained optimization. Lewis and Torczon extended the GPS framework to bound constrained optimization [20], and more generally [22], for problems with a finite number of linear constraints. Audet and Dennis [2] allow extended valued functions, which arise often in practice (*e.g.*, [4, 5]), and provide an analysis that, among other things, identifies a specific set of limit points allowing the application of Clarke's [8] generalized derivatives under local Lipschitz continuity to unify, strengthen and simplify the unconstrained and simply constrained Lewis and Torczon theory.

Under the assumption that f is continuously differentiable, Torczon [26] showed that GPS methods produce some limit point for which the gradient of the objective function is zero, and Lewis and Torczon showed that their adaptations produce a Karush-Kuhn-Tucker point for bound constrained [20] and linearly constrained [22] problems.

Subsequences of trial points that are optimizers in some discrete sense are considered in [2]. Even without any assumptions on the smoothness of f, limit points of these subsequences are shown to exist under standard assumptions. It is also shown that the following intermediate results hold : If it turns out that f is Lipschitz near a strictly feasible limit point then Clarke's derivatives are nonnegative on a set of positive spanning directions. A positive spanning set is a set of directions in $\mathbb{R}^n$ whose non-negative linear combinations span the whole space $\mathbb{R}^n$. Moreover, if f is strictly differentiable (defined in Section 5.2) at that strictly feasible point, then the gradient is guaranteed to be zero. Similar results are shown when the limit point is on the boundary of the linearly constrained domain.

Assuming that the functions f and C are twice continuously differentiable and that the constraint Jacobian has full rank, Lewis and Torczon [23] propose and analyze a derivative-free procedure to handle general constraints. In their procedure, GPS for bound constraints is used to carry out the specified inexact minimizations of the sequence of augmented Lagrangian subproblems formulated by Conn, Gould, and Toint [9]. Our algorithm is a GPS alternative to their method, for the cases when one would prefer not to assume continuous second derivatives, or when one wishes to avoid estimating penalty parameters and Lagrange multipliers, but is willing to settle for weaker optimality conditions. Thus, we do not claim that our method is to be preferred for every problem.

One of our objectives is to construct an algorithm that can be used on applications where the objective and constraints are not given analytically, but as "black boxes". For such applications, a value $x \in \mathbb{R}^n$ will be used as input to a nontrivial simulation to evaluate $f(x)$ and $C(x)$. The subroutine call may fail to return a value a significant percentage of times it is invoked [4, 5, 6, 7, 17], and even when it succeeds several factors (*e.g.*, noise, numerical instability, modelling inaccuracy,...), may mean that one cannot construct accurate approximate derivatives. Under these circumstances, *i.e.*, the structure of the functions is unknown to the optimizer, we will settle for weakened local optimality convergence results. The pattern search filter algorithm presented here has the following features:

- It is completely derivative-free. It does not use or approximate any derivative information, nor does it attempt to linearize any constraint.
- It transparently generalizes GPS for unconstrained problems and for bound or linear constraints when they are treated, as in [2, 20, 22], by rejecting points infeasible for

those constraints, and by selecting polling directions that conform to the boundary of the domain (see Definition 4.1).

- It uses a step acceptance rule based on filter methods, so there is no need for any penalty constants or Lagrange multiplier estimates.
- It makes assumptions on the problem functions f and C that conform to the practical instances that interest us most [4, 3, 5]. They may be discontinuous and even take on infinite values. Therefore, no global smoothness assumption is justified; however, the strength of the optimality conditions guaranteed by the algorithm depends on local smoothness of the functions at the limit point.
- It preserves the desirable GPS property of requiring only simple decrease, which is expressed in the present context with respect to both the objective and and constraint violation functions.
- It does not require any constraint qualifications.

The Boeing Design Explorer software uses the GPS filter algorithm given here as a meta algorithm in the surrogate management framework for general nonlinear programming [3]. Thus, a key issue for us is that we preserve the division into SEARCH and POLL steps. The SEARCH steps we prefer make a global exploration of the variable space, and they might use inexpensive surrogates objective and constraints. The POLL step is a local exploration near an incumbent point and allows the theory to guarantee convergence. Both the SEARCH and POLL steps are detailed in Section 2. Another use of our algorithm, with properly chosen SEARCH steps is when one would rather not find only the nearby local optimizer usually found by derivative-based methods, but instead is willing to use some function evaluations to explore the domain more thoroughly. It is important to keep in mind that global optimization of black box functions is impossible, even to the point that if one had the global optimum, one would not be certain of it [25].

The paper is organized as follows. Sections 2 and 3 give a brief description of pattern search and filter algorithms. In Section 4, we present and begin the analysis of a new algorithm that combines their features. Specifically, without any smoothness assumptions on the problem, we show the existence of some promising limit points. Our optimality results rely on Clarke's calculus with respect to both the constraint violation and objective functions, and on the notion of contingent cones, and so we provide the necessary background in Section 5. Section 6 shows that if the constraint violation or objective function is locally smooth at such a limit point, then some first order optimality conditions are satisfied. In the absence of general constraints, the convergence results reduce to those presented in [2]. Finally, in Section 7, we make important points through three examples. First we show the value of a filter method as opposed to a "barrier" method, which rejects trial points that violate the linear constraints, [22, 2]. Second we show the advantages for our algorithm of a squared ℓ_2 over an ℓ_1 measure of constraint violations. Third we show that our main convergence result concerning the objective function cannot be improved without removing some of the flexibility of the algorithm or adding more assumptions about the problem. We conclude this section by applying our method to a real engineering problem. These examples show the strength and limitations of our approach. The paper concludes by a discussion of the significance of the convergence results.

M. Abramson's MatLab 6 implementation NOMADm of a suite of algorithms including the filter algorithm given here is at http://www.caam.rice.edu/~abramson/NOMADm.html.

2. Pattern search algorithms for unconstrained optimization. The reader is referred to [22, 2] for a thorough description of linearly constrained pattern search algorithms. In the present document, the same notation as in [2] is used.

2.1. Search and poll steps. Pattern search algorithms for unconstrained minimization generate a sequence of iterates $\{x_k\}$ in $\mathbb{R}^n$ with non-increasing objective function values. At each iteration, the objective function is evaluated at a finite number of points on a mesh (a discrete subset of $\mathbb{R}^n$ defined below) to try to find one that yields a lower objective function value than the incumbent. An incumbent solution is a trial point in $\mathbb{R}^n$ where the algorithm evaluated the objective function f and has the lowest value found so far. Any strategy may be used to select mesh points that are to be candidates for the next iteration, provided that only a finite number of points (possibly none) are selected. This is called the SEARCH step. We favor SEARCH procedures that choose candidate points independent of the incumbent and whose global reach is independent of the mesh size. We feel that such SEARCH procedures are more likely to discover different basins for the function than the one in which the initial point lies.

When the SEARCH fails in finding an improved mesh point, the POLL step must be invoked. In that "fall back" step the function value is evaluated at neighboring mesh points around x_k. If the POLL step also fails in finding an improved mesh point, then x_k is said to be a mesh local optimizer. The mesh is then refined and x_{k+1} is set to x_k. The situation for our constrained version is going to be a bit more complex, but is consistent in spirit.

If either the SEARCH or POLL step succeeds in finding an improved mesh point $x_{k+1} \neq x_k$ with a strictly lower objective function value, then the mesh size parameter is kept the same or increased, and the process is reiterated. Indeed, as long as improved mesh points are found, one would likely choose trial points on coarser meshes. With surrogate-based SEARCH steps [4], a great deal of progress can often be made with few function values, and $O(n)$ function values are needed only when the POLL step detects a mesh local optimizer, which indicates that the mesh needs to be refined. We warn the reader that there is only a cursory discussion of SEARCH strategies in the present paper. The reason is that since the SEARCH is free of any rule, except finiteness and being on the mesh, it cannot be used to enhance the convergence theory. Indeed, some examples in [1] exploit perverse SEARCH strategies to show negative results. However, we are willing to pay this "theoretical" price for the practical reasons given above.

The formal definition of the mesh requires the following. Let D be a finite matrix whose columns in $\mathbb{R}^n$ form a positive spanning set. We use the notation $d \in D$ to indicate that d is a column of the matrix D. It is also required that each column $d \in D$ is the product of a non-singular generating matrix $G \in \mathbb{R}^{n \times n}$ by some integer vector in $\mathbb{Z}^n$. The same generating matrix G is used for all directions d. See [1, 2] for further insight on this set D, or see [26] for the original equivalent formulation. The set valued function $M(\cdot, \cdot)$ defines the current mesh through the lattices spanned by the columns of D, centered around the current iterate x_k:

$$(2.1) \qquad\qquad M(x_k, \Delta_k) = \{x_k + \Delta_k D z : z \in \mathbb{N}^{n_D}\},$$

where $\Delta_k \in \mathbb{R}_+$ is the mesh size parameter, and n_D is the number of columns of the matrix D. Note that in Section 4.2 on the filter GPS algorithm, we will use a more general definition of the mesh.

When the SEARCH fails in providing an improved mesh point, the objective function must be evaluated at the mesh points that neighbor the current iterate x_k, the current incumbent solution. In the unconstrained case this *poll center* is x_k, the current iterate. This defines the poll set $P_k = \{x_k\} \cup \{x_k + \Delta_k d : d \in D_k\}$ for some positive spanning matrix $D_k \subseteq D$. This

notation means that the columns of D_k are chosen from those of D. We will refer to evaluating $f(x_k + \Delta_k d)$ as polling in the direction d.

In the filter algorithm presented in Section 4, we may not poll about the current iterate, but instead we will poll about some possibly different poll center with a different definition for the current mesh.

2.2. Parameter update. At any iteration, there are two possible outcomes, which lead to two sets of rules to update the parameters.

If the iteration fails to produce an improved mesh point, then the POLL step guarantees that x_k is a mesh local optimizer. The mesh is then refined. More precisely,

$$(2.2) \qquad \Delta_{k+1} = \tau^{w_k} \Delta_k$$

with $0 < \tau^{w_k} < 1$, where $\tau > 1$ is a rational number that remains constant over all iterations, and $w_k \leq -1$ is an integer bounded below by the constant $w^- \leq -1$.

If the iteration produces an improved mesh point, then the mesh size parameter is kept the same or is increased, and the process is reiterated. The coarsening of the mesh follows the rule

$$(2.3) \qquad \Delta_{k+1} = \tau^{w_k} \Delta_k$$

where $\tau > 1$ is defined above and $w_k \geq 0$ is an integer bounded above by $w^+ \geq 0$. By modifying the mesh size parameters this way, it follows that for any $k \geq 0$, there exists an integer $r_k \in \mathbb{Z}$ such that $\Delta_k = \tau^{r_k} \Delta_0$.

Typical values for the mesh parameter update are $\tau = 2$ and $w_k = -1$ when the poll center is shown to be a local mesh optimizer, and $w_k = 1$ when an improved mesh point is found. This leads to setting $\Delta_{k+1} = \frac{1}{2}\Delta_k$ when the mesh needs to be refined, and $\Delta_{k+1} = 2\Delta_k$ when the mesh is coarsened. An example of the direction matrix might be $D = [I_n \ -I_n]$, where I_n is the $n \times n$ identity matrix. The mesh would then be $M(x_k, \Delta_k) = \{x_k + \Delta_k z : z \in \mathbb{Z}^n\}$ and the POLL set would be $P_k = \{x_k\} \cup \{x_k \pm \Delta_k e_i : i = 1, 2, \ldots, n\}$, where e_i is the ith column of the identity matrix. In the case where D is constructed from all the columns of the set $\{-1, 0, 1\}^n$, the mesh is the same as the previous one, but the POLL set may differ. In $\mathbb{R}^2$ for instance it could be $P_k = \{x_k, x_k + \Delta_k(1,0)^T, x_k + \Delta_k(0,1)^T, x_k + \Delta_k(-1,-1)^T\}$.

We borrow Coope and Price's [10] terminology for the following final remark on GPS. These methods are said to be *opportunistic* in the sense that as soon as an improved mesh point is found, the current iteration may stop without completing the function evaluations in the SEARCH and POLL steps.

3. Filter algorithms for constrained optimization. Filter algorithms treat the optimization problem as biobjective: one wishes to minimize both the objective function f and a nonnegative aggregate constraint violation function h. Filter algorithms attempt to minimize both functions, but clearly priority must be given to h, at least until a feasible iterate is found. This priority appears also in our algorithm in the definition of the poll centers and the poll set. Fletcher *et al.* [14, 15, 16] do this via *restoration* steps. Another difference is that in keeping with pattern search algorithms for less general problems, we require only improvement in either f or h, while Fletcher *et al.* have a sufficient decrease condition in the form of an *envelope* over the filter that constitutes a "sufficiently unfiltered" condition.

The terminology used in this paper differs slightly from that used by Fletcher *et al.* Our notation is more compact for our class of algorithms, and so it simplifies the presentation of our results. In addition, since our plan is to provide a truly multiobjective GPS algorithm in later work, and since it is likely to involve a version of the filter, it is best to conform to standard terminology in multiobjective optimization. [13]

Fletcher *et al.*'s definition of *dominance* makes it a reflexive relation, which simplifies the definition of a filter, but we will forgo that convenience to adhere to standard multiobjective terminology. The point is that the reader familiar with the filter literature should read this section carefully. We will end up with almost the standard notion of a filter, but we will define it slightly differently using the standard multiobjective notion of dominance: For a pair of vectors w, w', with finite components, w *dominates* w', written $w \prec w'$, if and only if $\forall i, w_i \leq w'_i$, and $w \neq w'$. We will use $w \preceq w'$ to indicate that either $w \prec w'$, or that $w = w'$, which is the notion of dominance used in earlier filter papers.

The constraint violation function is defined to satisfy the following properties: $h(x) \geq 0$, $h(x) = 0$ if and only if x is feasible, thus $h(x) > 0$ if and only if x is infeasible, and $h(x) = +\infty$ whenever any component of $C(x)$ is infinite. For example, we could set $h(x) = \|C(x)_+\|$ where $\|\cdot\|$ is a vector norm and where $(C(x)_+)_i$ is set to zero if $c_i(x) \leq 0$ and to $c_i(x)$ otherwise, $i = 1, 2, \ldots, n$. We show in Section 6.1 that the more locally smooth h is, the better the algorithm is able to exploit the positive spanning sets used. Our analysis and the examples in Sections 5.2 and 7.2 indicate that $h(x) = \|C(x)_+\|_2^2$ is a sound choice.

Recall that the feasible region of the optimization problem (1.1) is defined to be the intersection of a polyhedron X and Ω. Since it is simple to remain feasible with respect to X, we define a second constraint violation function $h_X = h + \psi_X$, where ψ_X is the indicator function for X. It is zero on X and $+\infty$ elsewhere. We will see in the next section that by applying our pattern search filter algorithm to h_X and f, the convergence results with respect to feasibility will depend on local smoothness of h, and not of h_X, which is obviously discontinuous on the boundary of X.

There should be no confusion in defining a special meaning of dominance for the vector arguments of our problem functions h_X, f. This will simplify our terminology rather than to use some other symbol such as $\prec_{(h_X, f)}$. Thus, a point $x_k \in \mathbb{R}^n$ is said to *dominate* $x \in \mathbb{R}^n$, $x_k \prec x$ if and only if $(h_X(x_k), f(x_k))^T \prec (h_X(x), f(x))^T$. Two points are equivalent if they generate an identical pair of h_X and f values. As above, $x \preceq x'$ indicates that either $x \prec x'$, or x and x' are equivalent.

A filter $\mathcal{F}$ is a finite set of infeasible points in $\mathbb{R}^n$ such that no pair x, x' in the filter are in the relation $x \prec x'$. A point x' is said to be filtered if either $x' \succeq x$ for some $x \in \mathcal{F}$, or if $h_X(x') \geq h_{max}$ for some positive finite upper bound h_{max} on allowable aggregate infeasibility, or if x is feasible and $f(x) \geq f^F$ (*i.e.*, the least function value found so far at a feasible point). The point x is unfiltered otherwise. The set of filtered points $\overline{\mathcal{F}}$ is denoted in standard notation as:

$$(3.1) \qquad \overline{\mathcal{F}} = \bigcup_{x \in \mathcal{F}} \left\{ x' : x' \succeq x \right\} \cup \left\{ x' : h_X(x') \geq h_{max} \right\} \cup \left\{ x' : h_X(x') = 0, f(x') \geq f^F \right\}.$$

Observe that our notation implies that trial points tying an incumbent's f and h values are filtered, and thus are not improved mesh points However, they will be poll center candidates.

Unfiltered points are added to $\mathcal{F}$ as they are generated, and filtered ones are rejected. Whether a point is filtered can depend on when it is generated. This temporal property causes "blocking entries" [14]. In order to avoid the problem of blocking entries, the filter contains only infeasible points. The incumbent best feasible point is treated separately as a sort of single point filter that can only filter other feasible points. The reason for this separation is to encourage moving over an infeasible function ridge and approaching a different part of the feasible region.

4. A pattern search filter algorithm for constrained optimization. In the previous sections we presented the filter framework for general constraints, and the GPS algorithm for unconstrained optimization. We now present a GPS filter method for the optimization problem (1.1). When some of the constraints are known to be linear, *i.e.*, when X is not trivial, it is frequently advantageous to treat them separately from the others and to ask that every iterate belongs to X. This is especially true of linear equality or bound constraints. For any derivative-free algorithm, one should surely use linear equality constraints to eliminate variables, and this is desirable for nonlinear equality constraints if it is practical.

4.1. Bound and linear constraints. By applying the algorithm to h_X instead of h, any trial point outside of X is rejected since its constraint violation function value is larger than h_{max}. We called this the "barrier" approach for X. In the absence of general constraints [2], the indicator function is added to f instead of h. In the present work, we cannot add it to f since the trial points $x \in X$ for which $-\infty < f(x) \le \infty$ and $0 < h(x) < h(x')$ for all $x' \in \mathcal{F}$ are unfiltered.

In addition, the fact that some linear constraints are explicitly known must be used to select mesh directions that take into account the geometry of the region X, just as suggested in [20, 22, 2]: When the poll center is within a given tolerance $\varepsilon > 0$ of the boundary of X, then the positive spanning directions D_k that define the poll set are chosen to contain the ones that span the tangent cone $T_X(y)$ to X at all boundary points y within the tolerance ε. The formal definition is as follows:

DEFINITION 4.1. *A rule for selecting the positive spanning sets $D_k = D(k, x_k) \subseteq D$ conforms to X for some $\varepsilon > 0$, if at each iteration k and for each y in the boundary of X for which $\|y - x_k\| < \varepsilon$, $T_X(y)$ is generated by a nonnegative linear combinations of the columns of a subset D_k^y of D_k.*

These tangent cone directions should be added to D_k before getting too close to the boundary, *i.e.*, it is best in our experience not to take the tolerance ε too small.

4.2. Meshes and poll centers. In our proposed pattern search algorithm, the test for accepting an improved mesh point is not based solely on the decrease of the objective function value when there are constraints. Therefore, the terminology *improved mesh point* (used in the unconstrained case) is not suited in a biobjective context. Instead, we will use the terminology *unfiltered mesh point* when either the SEARCH or POLL step finds a mesh point that is not filtered. If both steps fail in finding an unfiltered mesh point, then we cannot say that the poll center is a *mesh local optimizer* (as in the unconstrained case), instead we will say that the poll center, which will be chosen to be one of two special points in the filter or else a point that ties one of them, is a *mesh isolated filter point* since its mesh neighbors (the points in the poll set) are all filtered.

As in the pattern search algorithms presented in Section 2, the SEARCH and POLL steps are opportunistic, and may be terminated without any more function evaluations when an unfiltered mesh point is found. The mesh size parameter is then either increased or kept constant according to rule (2.3). When no such point is found, the poll center is a mesh isolated filter point and the filter remains unmodified. The mesh size parameter is decreased according to rule (2.2). Unlike Fletcher *et al.*'s filter algorithms, there is no "envelope" added to the filter to guarantee a form of sufficient decrease.

We define two types of incumbents: the feasible ones, and the infeasible ones with minimal constraint violation. Let f_k^F represent the feasible incumbent value, *i.e.*, the smallest objective function value (for feasible points) found by the algorithm up to iteration k If no feasible point has been found, f_k^F is set at ∞. Let $h_k^I > 0$ be the least positive constraint violation function value found up to iteration k, and let f_k^I denote the smallest objective function value of the points found whose constraint violation function values are equal to h_k^I. If no

such point exists, or if $h_k^I > h_{max}$ then h_k^I is fixed at h_{max} and f_k^I at $-\infty$. The superscript F stands for *feasible* and I for *infeasible*. Figure 4.1 shows an example of a filter. Any $x \in \Omega$, found by the algorithm, is a feasible incumbent point if $f(x) = f_k^F$, and it is an infeasible incumbent point if $h(x) = h_k^I$ and $f(x) = f_k^I$.
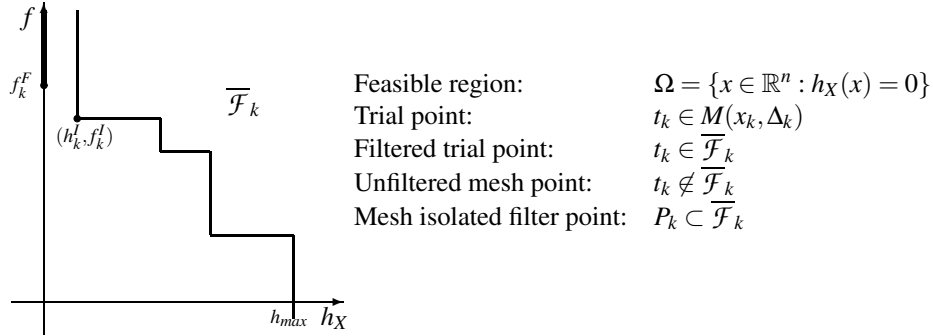

Feasible region: $\Omega = \{x \in \mathbb{R}^n : h_X(x) = 0\}$
Trial point: $t_k \in M(x_k, \Delta_k)$
Filtered trial point: $t_k \in \overline{\mathcal{F}}_k$
Unfiltered mesh point: $t_k \notin \overline{\mathcal{F}}_k$
Mesh isolated filter point: $P_k \subset \overline{\mathcal{F}}_k$

FIG. 4.1. *Feasible incumbent set:* $\{x \in \Omega \cap S_k : f(x) = f_k^F\}$. *Infeasible incumbent set* $\{x \in X \cap S_k : h(x) = h_k^I, f(x) = f_k^I\}$.

Our filter algorithm does not have a sufficient decrease condition. Any unfiltered mesh point (*i.e.*, for which $x \notin \overline{\mathcal{F}}$ – see equation (3.1)) is a candidate for being a new iterate. Our convergence results are for the iterates that are incumbents. Still, iterate is a useful term in an implementation because a new iterate means a new unfiltered mesh point has been found and the user can continue to search on the current mesh as in [3] without resorting to a POLL step.

In the unconstrained case, there was a single type of incumbent solution. Therefore the mesh was necessarily constructed around it. We redefine the current mesh so that it contains more points, therefore allowing more flexibility to the algorithm. Recall that the mesh is conceptual in the sense that it is never actually constructed, and its sole purpose is to allow us to capture some structure about the trial points so that we can derive some convergence results. Let S_0 be the set of initial trial points provided by the user at which the function values are computed. We assume that at least one point of S_0 has a constraint violation function value less than h_{max} and that every element of S_0 lies on $M_0(x, \Delta_0)$ (see (2.1)) for some $x \in S_0$. Define S_k to be the set of points where the functions were evaluated by the start of iteration k. The mesh is now defined as the following union:

$$(4.1) \qquad M(S_k, \Delta_k) = \bigcup_{x \in S_k} M(x, \Delta_k) ,$$

where $M(x, \Delta_k)$ is defined in equation (2.1). This new definition allows more flexibility to the user. For example, the SEARCH step is now allowed to poll around any trial point x in S_k, *i.e.*, to evaluate the functions at points from the set $\{x + \Delta_k d : d \in D\}$.

The poll center p_k, the point around which the poll set is constructed, is chosen either in the set of feasible incumbents or to be one of the infeasible incumbents with minimal constraint violation. Note that when these sets of incumbents are non-empty, they each will usually be composed of a single element. Thus, the poll center either satisfies

$$(4.2) \qquad (h_X(p_k), f(p_k)) = (0, f_k^F) \ \text{ or } \ (h_X(p_k), f(p_k)) = (h_k^I, f_k^I) .$$

The poll set is the poll center p_k together with its mesh neighbors:

$$(4.3) \qquad P_k = \{p_k\} \cup \{p_k + \Delta_k d : d \in D_k\} .$$

The positive spanning matrix D_k is composed of columns of D and conforms to the boundary of the linear constraints for an $\varepsilon > 0$ (see Definition 4.1). The current iterate x_k is typically used to seed the SEARCH step, and the poll center is always used for the POLL step.

Our class of algorithms and their analysis are completely flexible about the choice between these two sets of poll centers. The user may supply a strategy to select a poll center. In Section 7, we give what we hope are convincing arguments not to always make one choice or the other. Indeed, always choosing a stationary feasible poll center makes the filter algorithm essentially reduce to the barrier approach, which is not indicated for constraints where the polling directions do not conform to the feasible region. Still, we prefer neither to prescribe nor to proscribe any choice rule for poll centers. This flexibility may seem tedious to the reader, but the user may have a clear preference based on the results of the SEARCH, which may have involved some strategy that makes it unlikely that one or the other choice would be successful. For example, surrogate polling around several filter points in the SEARCH step seems promising in early tests, and the results would be likely to influence the choice of poll center.

Even if we already have a feasible incumbent point, we may wish to poll around one of the least infeasible points, which might have a lower objective function value, in order to try to find and explore a different part of the feasible region Ω. Also, this is what allows our filter algorithm to avoid stalling in the Lewis and Torczon [20] example when those linear constraints are treated by the filter. This is illustrated in Section 7.1.

4.3. Description of the algorithm. At any iteration, three types of unfiltered mesh points $x_{k+1} \in M(S_k, \Delta_k)$ can be generated by the algorithm: The most useful ones are the unfiltered feasible mesh points. They improve the feasible incumbent value to $f_{k+1}^F = f(x_{k+1}) < f_k^F$. Next are the infeasible ones that improve the infeasible incumbent with minimal constraint violation: $0 < h_{k+1}^I = h_X(x_{k+1}) < h_k^I$ and $f_{k+1}^I = f(x_{k+1})$. Finally there are the other infeasible ones that add some elements to the filter, but leave the incumbents unchanged. In all three cases, the mesh size parameter is updated according to rule (2.3), with possibly some different values of w_k.

To check if a trial point x is filtered or not, the following strategy is used in order to avoid wasting expensive function evaluations of f and C. First $\psi_X(x)$ is evaluated by determining if x belongs to X. If not, then x is filtered and the evaluation of $f(x)$ and $C(x)$ is avoided. Second, it is possible that partial information on $C(x)$ allows the algorithm to conclude that x is filtered. For example, if $h(x) = \|C(x)_+\|_2^2$ and if it known that $\sum_{i=1}^p |c_i(x)_+|^2 \geq h_{max}$ for some index $p < m$, then the evaluation of $f(x)$ and $c_i(x)$ for $i = p+1, p+2, \ldots, m$ is not necessary. Similar observations hold if $f(x)$ and partial information on $C(x)$ are known, though this situation is more complicated since the value of $f(x)$ alone can not allow us to conclude that x is filtered without at least partial knowledge of $h(x)$.

When all trial points are filtered, then the poll center p_k is a mesh isolated filter point. Regardless of the feasibility of x_k, the next iterate x_{k+1} is set to x_k, and the mesh size parameter is decreased according to rule (2.2). The next poll center p_{k+1} need not be fixed to p_k. These iterations usually require more function evaluations than when an unfiltered mesh point is found. A sensible strategy is to poll around both incumbents before decreasing the mesh size parameter. Logically, one can declare that the first POLL step was actually a part of the SEARCH.

A typical way in updating the mesh size parameter is to double it when a new incumbent solution is found, otherwise to keep it constant when only an unfiltered mesh point is found, and to cut it in half when the poll center is shown to be a mesh isolated filter point.

Our algorithm for constrained optimization is formally stated in figure 4.2. We allow for the fact that in some applications, a set S_0 of initial points may be available from solving

similar problems, and can be used to seed the filter. Without any loss of generality we assume that any such points, or at least the undominated ones, are on the initial mesh and that they have been "filtered" to be consistent with our initialization step in the sense that x_0 will not be filtered by the other seed points. An easy way to assure this would be to take x_0 to be the seed point with the smallest value of h_X, to break ties by taking one with the smallest objective function value, and to make sure that the necessary directions are in D in order that all the initial filter points are on the mesh. Of course, one must ensure that the directions satisfy the conditions of Section 2.

A pattern search filter algorithm

- INITIALIZATION: Let $\mathcal{F}_0$ be the filter associated with a set of initial points S_0. Let x_0 be an undominated point of $\mathcal{F}_0$. Fix $\Delta_0 > 0$ and set the iteration counter k to 0.
- DEFINITION OF INCUMBENT POINTS: Define (if possible)
  f_k^F: the least objective function value for all feasible points found so far;
  $h_k^I > 0$: the least positive constraint violation function value found so far;
  f_k^I: the least objective function value of the points found so far whose constraint violation function value is equal to h_k^I.
- SEARCH AND POLL ON CURRENT MESH $M(S_k, \Delta_k)$ (see equation (4.1)): Perform the SEARCH and possibly the POLL steps (or only part of the steps) until an unfiltered trial point x_{k+1} is found, or when it is shown that all trial points are filtered by $\mathcal{F}_k$.
  - SEARCH STEP: Evaluate h_X and f on a set of trial points on the current mesh $M(S_k, \Delta_k)$ (the strategy that gives the set of points is usually provided by the user).
  - POLL STEP: Evaluate h_X and f on the poll sets P_k (see equation (4.3)), for a poll center p_k that satisfy equation (4.2).
- PARAMETER UPDATE: Let $S_{k+1} = S_k \cup \{$ the set of all trial points visited in the SEARCH and POLL steps $\}$. If the SEARCH or the POLL step produced an unfiltered mesh point $x_{k+1} \notin \overline{\mathcal{F}}_k$, then update $\Delta_{k+1} \geq \Delta_k$ according to rule (2.3), then update the filter at the next step.
  Otherwise, set $x_{k+1} = x_k$, update $\Delta_{k+1} < \Delta_k$ according to rule (2.2), and set $\mathcal{F}_{k+1} = \mathcal{F}_k$; increase $k \leftarrow k+1$ and go back the definition of the incumbents.
- FILTER UPDATE: Let $\mathcal{F}_{k+1}$ be the union of $\mathcal{F}_k$ with all infeasible unfiltered points (with respect to $\mathcal{F}_k$) found during the SEARCH and POLL step. Remove dominated points from $\mathcal{F}_{k+1}$. Increase $k \leftarrow k+1$ and go back to the definition of the incumbents.

FIG. 4.2. *A pattern search filter algorithm*

In pattern search algorithms, one role of the POLL step is to guarantee convergence. This is why it is rigidly defined through the positive spanning sets $D_k \subset D$. In practice, the largest improvements in the incumbent points are obtained in the SEARCH step (*e.g.*, see [3, 4, 5] where a surrogate of an expensive function is constructed). The SEARCH step is usually the one that drives the iterates away from a local optimum. In a SEARCH implementation, it might be a good idea to try some points that are near points of the filter. Paul Frank made the interesting suggestion that SEARCH might include polling around the next most feasible filter point, *i.e.*, $x \in \mathcal{F}_k$ with the least value of $h_X(x) > h^I$. The objective here again is to attempt to find and then explore a different part of the feasible region. This is illustrated by the example in Section 7.4.

In the next section, we discuss the reduction of the algorithm proposed here in the ab-

sence of nonlinear constraints to those given earlier for unconstrained and linearly constrained problems.

4.4. Reduction of the GPS-filter method to linearly constrained optimization. Consider the case where $m = 0$, *i.e.*, when there are no nonlinear constraints. In [2], linear constraints defining X were handled by adding the indicator function to f and in the present paper, it is added to h. The effect is the same, since in both cases the indicator function simply eliminates from consideration the infeasible points with respect to X. In both cases, the convergence results are relative to the smoothness of h and f and not of h_X and f_X.

The main result of [2] was that for the limit $\hat{x}$ of a refining subsequence $\{p_k\}_{k \in K}$, Clarke derivatives are nonnegative in all the unsuccessful polling directions used infinitely often in the subsequence. The analysis below generalizes this by identifying a large set of directions for which Clarke derivatives are nonnegative.

4.5. Infinite refinement of the mesh. In this section, we identify a set of limit points of the algorithm that exists even without assuming any problem smoothness. Later we will show some optimality conditions hold at these limit points for which the problem is locally smooth.

The convergence analysis of our algorithm is based on the standard (see [9, 14, 15, 16]) assumption that all trial points produced by the algorithm lie in a compact set. A consequence of this is that since the mesh size parameter does not decrease when an unfiltered mesh point is found ($\Delta_{k+1} \geq \Delta_k$), then it follows that only finitely many *consecutive* unfiltered mesh points can be generated.

We will be mainly concerned with the poll centers p_k that are mesh isolated filter points (*i.e.*, the mesh neighbors of p_k are filtered) and for which the mesh size parameter is reduced ($\Delta_{k+1} < \Delta_k$). The proofs of the results in this subsections are omitted, even if the definition of the mesh is slightly different. The key element required is not the mesh, but the fact that any mesh point $x \in M(S_k, \Delta_k)$ can be written as $x + \sum_{i=0}^{k} \Delta_i D z_i$ for some $x \in S_0$ and $z_i \in \mathbb{N}^{n_D}$, for $i = 0, 1, \dots, k$.

Our first result is that there is a subsequence of iterations for which the mesh size parameter goes to zero. In order to prove it we require the following lemma from Torczon [26] or Audet and Dennis [2] whose proof can be easily modified with our definition (4.1) of the mesh.

LEMMA 4.2. *The mesh size parameters Δ_k are bounded above by a positive constant independent of the iteration number k.*

Combining this lemma with the assumption that all iterates lie in a compact set implies the following result. Its proof is omitted since it is identical to that of the same result in [2]. The original proof of this, using slightly different notation can be found in Torczon [26].

LEMMA 4.3. *The mesh size parameters satisfy* $\liminf_{k \to +\infty} \Delta_k = 0.$

Coope and Price [10] analyze mesh-based algorithms for the unconstrained and linearly constrained problems in which instead of requiring that the SEARCH be performed on the mesh, they assume that the limit inferior of the mesh size parameter goes to zero. This shifts the burden from the algorithm specification to the implementation.

Since the mesh size parameter shrinks only at mesh isolated filter points, Lemma 4.3 guarantees that there are infinitely many iterations for which the poll centers are mesh isolated filter points. Thus by compactness, the mesh isolated filter points have limit points. Moreover, all these limit points belong to the polyhedron X. At such an iteration the entire trial set, and in particular the poll set P_k, is filtered. Therefore, for each direction $d \in D_k$ either $h_X(p_k + \Delta_k d) \geq h_{max}$, or there exists some element x in the filter $\mathcal{F}_k$ such that both $f(p_k + \Delta_k d) \geq f(x)$ and $h_X(p_k + \Delta_k d) \geq h_X(x)$ or, $h_X(p_k + \Delta_k d) = 0$ and $f_X(p_k + \Delta_k d) \geq f_k^F$.

5. Background for optimality results. As in [2], Clarke's [8] generalized derivatives are the key to our convergence analysis. To use this powerful tool, we analyze the case where the function is Lipschitz in a neighborhood of the limit point in question. Of course, there are some optimization problems on which we would apply our algorithm where the functions are not Lipschitz, or optimization problems where we cannot show that the functions are Lipschitz. But this is beside the point. We show how the algorithm behaves on problems with Lischitz functions. Another ingredient needed for optimality conditions is the contingent cone, which generalizes the notion of tangent cone to more general constraints. The following material is adapted from [18, 24].

DEFINITION 5.1. *Let $S \subset \mathbb{R}^n$ be nonempty. The cone generated by S is*

$$(5.1) \qquad cone(S) = \{ \lambda s \mid \lambda \geq 0 \text{ and } s \in S \} .$$

A tangent vector to S at x in the closure of S is $v \in \mathbb{R}^n$ such that there exists a sequence $\{y_k\}$ of elements of S that converges to x and a sequence of positive real numbers $\{\lambda_k\}$ for which $v = \lim_k \lambda_k(y_k - x)$. The set $T(S,x)$ of all tangent vectors to S at x is called the contingent cone (or sequential Bouligand tangent cone) to S at x. The polar cone of a cone $K \subset \mathbb{R}^n$ is $K^\circ = \{x \in \mathbb{R}^n : x^T v \leq 0 , \forall v \in K\}$.

For X, the contingent cone is the same as the tangent cone. The normal cone, used to define KKT points, is less useful here than the polar cone since the normal cone in our context may have little to do with optimality given its usual definition as the convex conic hull of the gradients of the constraints. The polar cone of the contingent cone is more useful in this context.

Optimality conditions for a differentiable function can be stated in terms of the cone generated by the convex hull of a set S, *i.e.*, the set of nonnegative linear combinations of elements of S. We will use the standard notation $co(S)$ for the convex hull of S, but rather than use the induced but somewhat unwieldy notation $cone(co(S))$, we will use the notation $cc(S)$ for the convex conic hull of S. Thus, for example, to say that a set S is a positive spanning set is to say that $cc(S) = \mathbb{R}^n$.

DEFINITION 5.2. *[8] Let $g : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz near $\bar{x} \in \mathbb{R}^n$. Clarke's generalized derivative at $\bar{x}$ in the direction $v \in \mathbb{R}^n$ is*

$$g^\circ(\bar{x};v) := \limsup_{y \to \bar{x}, \, t \downarrow 0} \frac{g(y+tv) - g(y)}{t}.$$

The generalized gradient of g at $\bar{x}$ is the set

$$\partial g(\bar{x}) := \{s \in \mathbb{R}^n : g^\circ(\bar{x};v) \geq v^T s \text{ for all } v \in \mathbb{R}^n\}.$$

The generalized derivative may be obtained from the generalized gradient as follows : $g^\circ(\bar{x};v) = \max\{v^T s : s \in \partial g(\bar{x})\}$.

The following alternate definition of directional derivative will be useful.

LEMMA 5.3. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz near $\bar{x} \in \mathbb{R}^n$. Then,*

$$g^\circ(\bar{x};v) = \limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y+tw) - g(y)}{t}.$$

Proof. Let L be a Lipschitz constant for g near $\bar{x}$. Then

$$\limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y+tw) - g(y)}{t} = \limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y+tv) - g(y)}{t} + \frac{g(y+tw) - g(y+tv)}{t}$$

$$\leq \limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y+tv) - g(y)}{t} + L\|w - v\| = g^\circ(\bar{x};v) .$$

On the other hand, setting $w = v$ gives a lower bound on the limit supremum.

$$\limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y+tw) - g(y)}{t} \geq \limsup_{y \to \bar{x}, \, t \downarrow 0} \frac{g(y+tv) - g(y)}{t} = g^{\circ}(\bar{x}; v) .$$

∎

In order to show that the Clarke generalized directional derivative is nonnegative at a point $\bar{x} \in \mathbb{R}^n$ in the direction $v \in \mathbb{R}^n$, it suffices to generate three subsequences: $\{y_k\}$ converging to $\bar{x}$, $\{w_k\}$ converging to v, and $\{t_k\}$ converging to zero from above in such a way that $g(y_k) \leq g(y_k + t_k w_k)$ for infinitely many k's.

5.1. Clarke's derivatives at limit points. In this subsection, we develop some results about the directions in which the Clarke derivatives indicate optimality. To save space, we prove our preliminary results for a function g, which can be either h or f. Also, for any $\bar{x} \in \mathbb{R}^n$, we define $\Gamma_g(\bar{x})$ to be the closure of $\{x \in \mathbb{R}^n : g(x) \geq g(\bar{x})\}$.

PROPOSITION 5.4. *Let $S \subset \mathbb{R}^n$ be nonempty, and g be defined on an open superset of S, and let g be Lipschitz near $\bar{x} \in S$. Necessary conditions for $\bar{x}$ to be a local minimizer of g on S are:*

- $g^{\circ}(\bar{x}; v) \geq 0$ *for every $v \in T(S, \bar{x})$;*
- *If g has a Fréchet derivative $\nabla g(\bar{x})$ at $\bar{x}$, then $\nabla g(\bar{x})^T v \geq 0$ for every $v \in co(T(S, \bar{x}))$, and so $-\nabla g(\bar{x}) \in co(T(S, \bar{x}))^{\circ}$. Thus, if $T(S, \bar{x})$ contains a positive spanning set, then $co(T(S, \bar{x}))^{\circ} = \{0\}$ and $\nabla g(\hat{x}) = 0$.*

Proof. Let S, g and $\bar{x}$ be as in the statement, and let v be in $T(S; \bar{x})$. Then, there exists a sequence $\{x_k\}$ of elements of S converging to the local minimizer $\bar{x}$ of g on S, and some positive sequence $\{\lambda_k\}$ such that $v = \lim_k \lambda_k (x_k - \bar{x})$. If $v = 0$, the result is trivial. If $v \neq 0$, then $\lim_k \frac{1}{\lambda_k} = 0$.

Now, take $y_k = \bar{x}$, $w_k = \lambda_k (x_k - \bar{x})$, and $t_k = \frac{1}{\lambda_k}$; we see that

$$g^{\circ}(\bar{x}; v) \geq \limsup_k \frac{g(y_k + t_k w_k) - g(y_k)}{t_k} = \limsup_k \lambda_k [g(x_k) - g(\bar{x})] .$$

But, since $x_k \in S$, $\{x_k\}$ converges to $\bar{x}$, and $\bar{x}$ is a local minimizer of g on S, we have that for sufficiently large k, $\lambda_k [g(x_k) - g(\bar{x})]$ is nonnegative and the first result follows.

Now assume that $\nabla g(\bar{x})$ is the Fréchet derivative at $\bar{x}$. Then by Theorem 4.14 of [18], $\nabla g(\bar{x})^T v \geq 0$ for every $v \in T(S, x)$. Let $v \in co(T(S, \bar{x}))$. Then, there is a nonnegative coefficient vector α such that $v = \sum_i \alpha_i s_i$ for some $s_i \in T(S, \bar{x})$. The second result follows from the linearity of the inner product and the definition of polar cone. If $T(S, \bar{x})$ contains a positive spanning set, then $co(T(S, \bar{x})) = \mathbb{R}^n$, and therefore, for every $v \neq 0$, we have that $v, -v \in T(S, \bar{x})$, and so $\nabla g(\bar{x})^T v \geq 0$ and $\nabla g(\bar{x})^T (-v) \geq 0$, which completes the proof. ∎

The approach we now give for generating directions in which Clarke derivatives are nonnegative generalizes the one presented in [2]. Indeed, the following result will be useful in enlarging the set of directions, and in addition, it relates the generalized directional derivative to the class of iterative methods that require decrease in some merit function at each iteration. We prove this more general result first.

LEMMA 5.5. *Let g be Lipschitz near the limit $\bar{x}$ of a sequence $\{y_k\}$ for which the corresponding values $g(y_k)$ are monotone nonincreasing, and for which $y_k \neq \bar{x}$ for all k. If v is any limit point of the sequence $\left\{ \frac{y_k - \bar{x}}{\|y_k - \bar{x}\|} \right\}$, then $v \in T(\Gamma_g(\bar{x}), \bar{x})$ and $g^{\circ}(\bar{x}; v) \geq 0$.*

Proof. Let $\{y_k\}$ and $\bar{x}$ be as in the above statement. There is at least one limit point v of $\left\{ \frac{y_k - \bar{x}}{\|y_k - \bar{x}\|} \right\}$ since the unit ball is compact. Setting $\lambda_k = \frac{1}{\|y_k - \bar{x}\|}$ in Definition 5.1 yields trivially

that $v \in T(\Gamma_g(\bar{x}), \bar{x})$. Moreover, Lemma 5.3 implies

$$
\begin{aligned}
g^\circ(\bar{x}; v) &= \limsup_{y \to \bar{x}, \, w \to v, \, t \downarrow 0} \frac{g(y + tw) - g(y)}{t} \\
&\geq \limsup_{k \to \infty} \frac{g\left(\bar{x} + \|y_k - \bar{x}\| \frac{y_k - \bar{x}}{\|y_k - \bar{x}\|}\right) - g(\bar{x})}{\|y_k - \bar{x}\|} = \limsup_{k \to \infty} \frac{g(y_k) - g(\bar{x})}{\|y_k - \bar{x}\|} \geq 0 \, .
\end{aligned}
$$

∎

5.2. Choice of the constraint violation norm. The constraint violation function $h_2(x) = \|C(x)_+\|_2^2$ will give our best results since it is continuously differentiable whenever C is (see Dennis, El Alem and Williamson [11] for a compact formulation of ∇h_2). The constraint violation function $h_1(x) = \|C(x)_+\|_1$ is another common choice, at least for SQP. Thus, the question arises as to the differentiability of h_1. The answer, which implies that it is rarely strictly differentiable at $\hat{x}$, is given by the following result. Recall that a function g is said to be *strictly differentiable* [19, 8] at x if $\lim_{y \to \hat{x}, t \downarrow 0} \frac{g(y + tv) - g(y)}{t} = \nabla g(\hat{x})^T v$ for all $v \in \mathbb{R}^n$, and g is said to be *regular* [8] at x if for all $v \in \mathbb{R}^n$, the one-sided directional derivative $g'(x, v)$ in the direction v exists and coincides with $g^\circ(x; v)$. In Section 7.2 we will see an example showing the cost of this lack of smoothness.

PROPOSITION 5.6. *If C is regular at every x, then so is h_1. Let $I(x) = \{i : c_i(x) > 0\}$ and $A(x) = \{i : c_i(x) = 0\}$ be the inactive and active sets at x. Then the generalized gradients are related by*

$$
\partial h_1(x) = \sum_{i \in I(x)} \partial c_i(x) + \left\{ \sum_{i \in A(x)} \gamma_i \zeta_i \, : \, \gamma_i \in [0, 1], \zeta_i \in \partial c_i(x), i \in A(x) \right\}.
$$

The generalized directional derivatives of h_1 and C in a direction v at x are related by

$$
h_1^\circ(x; v) = \sum_{i \in I(x)} c_i^\circ(x; v) + \sum_{i \in A(x)} (c_i^\circ(x; v))_+.
$$

Thus, if C is strictly differentiable at x, then

$$
h_1^\circ(x; v) = \sum_{i \in I(x)} \nabla c_i(x)^T v + \sum_{i \in A(x)} (\nabla c_i(x)^T v)_+.
$$

Proof. The proof follows from various results in [8], and some simple observations. Clarke's Propositions 2.3.12 and 2.3.6 guarantee that $c_i(x)_+$ thus $h_1(x)$ are regular at x whenever $c_i(x)$ is.

The third corollary to Proposition 2.3.3 implies that $\partial h_1(x) = \sum_i \partial c_i(x)_+$, where this means all possible sums of an element from each $\partial c_i(x)_+$. Propositions 2.3.12 implies that

$$
\partial c_i(x)_+ = \begin{cases} \partial c_i(x) & \text{if } c_i(x) > 0, \\ co\{\partial c_i(x), \partial 0(x)\} = \{\gamma_i \zeta_i : \gamma_i \in [0, 1], \zeta_i \in \partial c_i(x)\} & \text{if } c_i(x) = 0, \\ \partial 0(x) = \{0\} & \text{if } c_i(x) < 0. \end{cases}
$$

The generalized directional derivative in any direction v can be written as: $h_1^\circ(x; v) = \sum_i (c_i(x; v)_+)^\circ$. If $c_i(x) > 0$ then $(c_i(x; v)_+)^\circ = \max\{v^T \zeta : \zeta \in \partial c_i(x)\} = c_i^\circ(x; v)$. If $c_i(x) < 0$ then $(c_i(x; v)_+)^\circ = \max\{v^T \zeta : \zeta \in \partial 0(x)\} = 0$. And if $c_i(x) = 0$, then

$$
\begin{aligned}
(c_i(x;v)_+)^\circ &= \max\{v^T\eta : \eta \in \partial c_i(x)_+\} \\
&= \max\left\{v^T\eta : \eta \in \{\gamma_i\zeta_i : \gamma_i \in [0,1], \zeta_i \in \partial c_i(x)\}\right\} \\
&= \begin{cases} 0 & \text{if } \max\{v^T\zeta_i \in \partial c_i(x)\} \le 0 \\ \max\{v^T\zeta_i \in \partial c_i(x)\} & \text{otherwise} \end{cases} \\
&= (\max\{v^T\zeta_i \in \partial c_i(x)\})_+ \quad = \quad (c_i^\circ(x;v))_+.
\end{aligned}
$$

The last part of the result follows by definition of strict differentiability. ∎

Note that the above result could be slightly rewritten by using one-sided directional derivatives instead of generalized directional derivatives. Indeed, if C is regular at x, then $c_i^\circ(x;v)$ coincides with the one-sided directional derivative $c_i'(x;v)$, and $h_1^\circ(x;v)$ coincides with $h_1'(x;v)$.

5.3. Refining sequences. The purpose of the following definition is to identify a limit of trial points and as many directions as possible for which Clarke's derivatives are nonnegative at that limit. We will make the convention, which is implied by the algorithm, that for p_k to be an *active poll center* implies that polling around p_k was at least initiated at iteration k, although function values at all the corresponding poll set may not have been computed because polling is allowed to stop if some poll step yields an improved mesh point.

DEFINITION 5.7. *A convergent subsequence of active poll centers $\{p_k\}_{k\in K}$ (for some subset of indices K) is said to be a* refining subsequence *if $\lim_{k\in K}\Delta_k = 0$. The set of* refining directions *for g associated with a refining subsequence $\{p_k\}_{k\in K}$ is*

$$
\begin{aligned}
R_g(K) = \{v \in \mathbb{R}^n : v = \zeta - \xi \ne 0 \text{ and } -\infty < g(p_k + \Delta_k\xi) \le g(p_k + \Delta_k\zeta) < \infty \\
\text{and } p_k + \Delta_k\zeta, p_k + \Delta_k\xi \in V_k \text{ for infinitely many } k \in K\},
\end{aligned}
$$

where $V_k \subset P_k$ are the members of the poll set visited by GPS. The set of limit directions *for g associated with the limit $\hat{x}$ of a refining subsequence $\{p_k\}_{k\in K}$ is*

$$
L_g(K) = \{v \in \mathbb{R}^n : \exists \{y_k\}_{k\in K} \subset V_k \setminus \{\hat{x}\} \text{ such that } \lim_{k\in K} y_k = \hat{x}, \text{ and } g(y_k) \ge g(y_{k'})
$$

$$
\forall k' > k \in K, \text{ and } v \text{ is a limit point of } \left\{\frac{y_k - \hat{x}}{\|y_k - \hat{x}\|}\right\}_{k\in K}\}.
$$

Figure 5.1 illustrates an example of refining directions for a refining subsequence p_{k_i}: The subsequence $\{p_k\}_{k\in K}$ converges to $\hat{x}$ and has six associated directions, represented by vectors, and four limit directions, represented by dotted lines. Note that in the above definition, if $\xi = 0$ then the refining direction $v = \zeta - \xi$ belongs to D_k. Thus, all the directions in infinitely many D_k where polling was unsuccessful in finding a better point are refining directions. Also, if the function is constant in the refining direction $v \in R_g(K)$, then $-v$ also will be a refining direction.

We now show the existence of refining subsequences and directions, but because the limit directions depend on whether g is f or h, we postpone their existence results for the next section.

LEMMA 5.8. *There exists at least one refining subsequence composed of mesh isolated filter points. Let $\{p_k\}_{k\in K}$ be a refining subsequence. If there exists some $t_k \in V_k$ with $-\infty < g(t_k) < \infty$ for infinitely many k in K, then the set of refining directions $R_g(K)$ is nonempty.*

Proof. Lemma 4.3 guarantees that there exists a subsequence of iterations whose mesh size parameter goes to zero. The mesh size parameter Δ_k only decreases when the POLL step
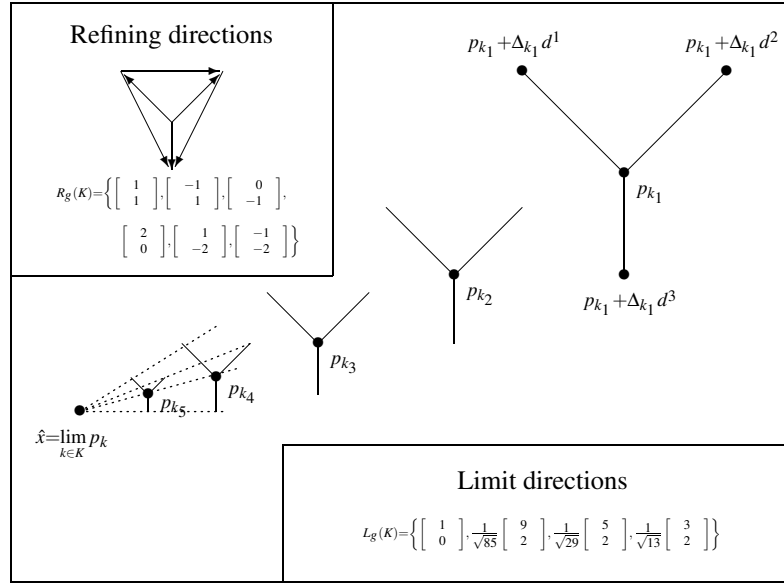
FIG. 5.1. *An example of refining and limit directions. The six refining directions (represented by vectors) use the fact that $g(p_{k_i} + \Delta_{k_i} d^1) > g(p_{k_i} + \Delta_{k_i} d^2) > g(p_{k_i} + \Delta_{k_i} d^3) > g(p_{k_i})$. The four limit directions are represented by dotted lines.*

shows that all trial points in P_k are filtered. Moreover, the assumption that all trial points (thus all active poll centers) are in a compact set implies that one such subsequence has a limit point. Thus, there exists a refining subsequence consisting exclusively of mesh isolated poll centers.

Let p_k and t_k be as in the above statement. The second result follows from the fact that the directions $\frac{t_k - p_k}{\Delta_k}$ belong to the finite set D, and therefore there is a direction $d \in D$ used infinitely often. By definition, either d or $-d$ (or both) belongs to $R_g(K)$. ∎

We now show that the generalized directional derivative is nonnegative at limit points of refining subsequences for all associated refining and limit directions for g.

THEOREM 5.9. *Let g be Lipschitz near the limit point $\hat{x}$ of a refining subsequence $\{p_k\}_{k \in K}$. Then g satisfies optimality conditions at $\hat{x}$ on $cone(R_g(K) \cup L_g(K))$ in the sense that if $v \in cone(R_g(K) \cup L_g(K))$ then $g^{\circ}(\hat{x}; v) \geq 0$. Moreover, $L_g(K) \subset T(\Gamma_g(\hat{x}), \hat{x})$.*

Proof. Let g be Lipschitz near the limit point $\hat{x}$ of a refining subsequence $\{p_k\}_{k \in K}$. If $v = \zeta - \xi \neq 0$ belongs to $R_g(K)$ for some $\zeta, \xi \in \mathbb{R}^n$, then by definition of the generalized directional derivative we have that:

$$g^{\circ}(\hat{x}; v) \geq \limsup_{k \in K} \frac{g((p_k + \Delta_k \xi) + \Delta_k d) - g(p_k + \Delta_k \xi)}{\Delta_k} \geq 0.$$

The same result on $cone(R_g(K) \cup L_g(K))$ follows from the positive homogeneity of the Clarke generalized directional derivative. If v belongs to $L_g(K)$ for some subsequence $\{y_k\}_{k \in K} \subset V_k \setminus \{\hat{x}\}$ converging to $\hat{x}$, then Lemma 5.5 completes the proof. ∎

The previous theorem implies that one of the advantages of using a large number of positive spanning directions in the algorithm is that the set of directions for which Clarke's generalized derivatives are shown to be nonnegative will be larger.

The following corollary strengthens Theorem 5.9 when g is strictly differentiable at the

limit point $\hat{x}$ Assuming that g is strictly differentiable at $\hat{x}$ as defined in Section 5.2 is equivalent in finite dimensions to assuming that g is Lipschitz near $\hat{x}$, Fréchet differentiable and regular at $\hat{x}$ [8].

COROLLARY 5.10. *If g is strictly differentiable at $\hat{x}$, then $\nabla g(\hat{x})^T v \geq 0$ for every $v \in cc(R_g(K) \cup L_g(K))$ and thus, if $R_g(K) \cup L_g(K)$ contains a positive spanning set, then $\nabla g(\hat{x}) = 0$.*

Proof. Assume that $\nabla g(\hat{x})$ is the Fréchet derivative at $\hat{x}$. Then Theorem 5.9 ensures that $\nabla g(\hat{x})^T v \geq 0$ for every $v \in R_g(K) \cup L_g(K)$. Let $v \in cc(R_g(K) \cup L_g(K))$. Then, there is a nonnegative coefficient vector α such that $v = \sum_i \alpha_i s_i$ for some $s_i \in R_g(K) \cup L_g(K)$. The first result follows from the linearity of the inner product.

If $R_g(K) \cup L_g(K)$ contains a positive spanning set, then $cc(R_g(K) \cup L_g(K)) = \mathbb{R}^n$, and therefore, for every $v \neq 0$, we have that $v, -v \in R_g(K) \cup L_g(K)$, and so $\nabla g(\bar{x})^T v \geq 0$ and $\nabla g(\bar{x})^T (-v) \geq 0$, which completes the proof. ∎

6. Optimality conditions for the Filter-GPS method. We will continue with results that only consider the behavior of h, and then complete our results by analyzing the effect of the filter on the objective function f.

6.1. Results for the constraint violation function. The algorithm definition gives priority to feasibility, as expressed by the constraint violation function, over minimizing the objective function. A consequence of this is that the optimality conditions guaranteed by the algorithm are stronger for h, i.e., achieving feasibility, than for they are for achieving constrained optimality. Indeed, in the absence of the assumption of linearly independent constraint gradients, our feasibility results are what one would prove for standard SQP methods – that we obtain a stationary point of the ℓ_2 norm of the constraint violations.

A first obvious comment is that $h(p_k) = h_X(p_k)$ for every poll center p_k since trial points violating some linear constraints are rejected. Therefore, any limit of poll centers belongs to X. So the analysis can be done in terms of h instead of h_X. Another observation is that, by definition, if $h(\hat{x}) = 0$ then $\hat{x}$ is a global minimizer for h. Furthermore, any limit point of a sequence of feasible points would be feasible if h were lower semicontinuous there, or if the feasible region is closed. However, it is possible for a sequence of least infeasible poll centers to converge to an infeasible point. We will therefore concentrate on limit points of infeasible mesh isolated poll centers.

Before presenting results that assume local Lipschitz continuity, we prove the following result, which shows in particular that if any limit point of least infeasible poll centers is feasible and if h is continuous there, then all limit points at which h is lower semicontinuous are also feasible. It also provides a way to identify some limit directions in $L_h(K)$.

THEOREM 6.1. *Let $\{p_k^I\}_{k \in K}$ be a convergent subsequence of least infeasible poll centers. Then $\lim_k h(p_k^I)$ exists, and if h is lower semicontinuous at any limit point $\bar{x}$ of $\{p_k^I\}$, then $\lim_k h(p_k^I) \geq h(\bar{x}) \geq 0$. Every limit point of least infeasible poll centers at which h is continuous has the same constraint violation function value. Furthermore, if h is Lipschitz near any $\bar{x}$, then $h^\circ(\bar{x}; v) \geq 0$ for any limit direction v of $\left\{ \frac{p_k^I - \bar{x}}{\|p_k^I - \bar{x}\|} \right\}$. In addition, each limit direction satisfies $v \in T(\Gamma_h(\bar{x}), \bar{x})$.*

Proof. The sequence $\{h(p_k^I)\}$ is convergent because it is a nonincreasing sequence of positive numbers. Of course, for any subsequence of $\{p_k^I\}$, the corresponding h values have the same limit. Thus, if h is lower semicontinuous at $\bar{x}$, we know that for any subsequence $\{p_k\}_{k \in K}$ of the iteration sequence that converges to $\bar{x}$,

$$\lim_k h(p_k^I) = \lim_{k \in K} h(p_k^I) = \liminf_{k \in K} h(p_k^I) \geq h(\bar{x}) \geq 0 \, .$$

If h is continuous at some limit points of infeasible poll centers, then the same argument shows that all such limit points have the same value of the constraint violation function. Thus, if any such limit point is feasible, they all are feasible.

The rest of the proof follows by noticing that $v \in L_h(K)$, and by applying Theorem 5.9. ∎

The next result guarantees some nonsmooth first order optimality conditions. It shows that Clarke derivatives for h are nonnegative in a subset of refining directions whose convex conic hull is the tangent cone to X.

PROPOSITION 6.2. *Let $\hat{x}$ be the limit of a refining subsequence composed of mesh isolated filter points $\{p_k\}_{k \in K}$, and assume that the rule for selecting D_k conforms to X for an $\varepsilon > 0$. If h is Lipschitz near $\hat{x}$, then $h^\circ(\hat{x}; v) \geq 0$ for any direction v in a set of directions $D' \subset R_h(K)$ satisfying $cc(D') = T(X, \hat{x})$.*

Proof. Let $\{p_k\}_{k \in K}$, ε and $\hat{x}$ be as in the statement of the result. When $h(\hat{x}) = 0$, $h^\circ(\hat{x}; v) \geq 0$ for any $v \in \mathbb{R}^n$, so assume that $h(\hat{x}) > 0$. Since the rule for selecting D_k conforms to X for an $\varepsilon > 0$, then there exists a subset of directions D' of D such that $cc(D') = T(X, \hat{x})$, and for any $v \in D'$ and sufficiently large $k \in K$, $p_k + \Delta_k v \in P_k \cup X$ and $h(p_k + \Delta_k v) > 0$. However, since the poll centers are mesh isolated filter points, it follows that $h(p_k + \Delta_k v) \geq h(p_k)$, and therefore v belongs to $R_h(K)$. Theorem 5.9 completes the proof. ∎

A consequence of this result is that if h is strictly differentiable at the limit point of a refining subsequence composed of mesh isolated filter points, then standard first order optimality conditions for h are satisfied.

COROLLARY 6.3. *Let $\hat{x}$ be the limit of a refining subsequence composed of mesh isolated filter points $\{p_k\}_{k \in K}$, and assume that the rule for selecting D_k conforms to X for an $\varepsilon > 0$. If h is strictly differentiable at $\hat{x}$, then $\nabla h(\hat{x})^T v \geq 0$ for every v in $T(X, \hat{x})$.*

Proof. The result is a direct consequence of Corollary 5.10 and Proposition 6.2. ∎

6.2. Results for the objective function. We have shown above that the limit point for a refining subsequence generated by the algorithm satisfies local optimality conditions for the constraint violation function. We now derive some results for the objective function. The first result proposes a way to identify some limit directions in $L_f(K)$.

PROPOSITION 6.4. *Let $\{p_k^F\}_{k \in K}$ be a convergent subsequence of feasible poll centers. If f is lower semicontinuous at $\bar{x}$, then $\lim_k f(p_k^k)$ exists and is greater than or equal to $f(\bar{x})$. The set of such limit points at which f is continuous all have the same objective function value. Furthermore, if f is Lipschitz near any $\bar{x}$, then any limit direction v of $\left\{ \frac{p_k^F - \bar{x}}{\|p_k^F - \bar{x}\|} \right\}$ is such that $f^\circ(\bar{x}; v) \geq 0$ and $v \in T(\Omega, \bar{x}) \cap T(\Gamma_f(\bar{x}), \bar{x})$.*

Proof. Let $\{p_k^F\}_{k \in K}$ and $\bar{x}$ be as in the above statement. The subsequence $f(p_k^F)_{k \in K}$ is monotone nonincreasing and bounded below by the finite value $f(\bar{x})$, and therefore it converges.

Since $p_k^F \in \Omega$ for every $k \in K$, it follows by definition of the contingent cone that $v \in T(\Omega, \bar{x})$. The rest of the proof follows by noticing that $v \in L_f(K)$, and by applying Theorem 5.9. ∎

The next pair of results guarantees some nonsmooth first order optimality conditions related to the cone tangent to X. They are similar to the first order optimality results for h, Proposition 6.2 and Corollary 6.3, except that they require the limit point to be strictly feasible with respect to Ω. Basically, these results show that when the nonlinear constraints are not binding, the use of the filter does not interfere with the linearly constrained results.

PROPOSITION 6.5. *Let $\hat{x}$ be the limit of a refining subsequence composed of mesh isolated filter points $\{p_k\}_{k \in K}$, and assume that the rule for selecting D_k conforms to X for an $\varepsilon > 0$. If f is Lipschitz near $\hat{x}$, and if $\hat{x}$ is strictly feasible with respect to Ω, then $f^\circ(\hat{x}; v) \geq 0$ for any direction v in a set of directions $D' \subset R_h(K)$ satisfying $cc(D') = T(X, \hat{x})$.*

Proof. Let $\{p_k\}_{k \in K}, \varepsilon$ and $\hat{x}$ be as in the statement of the result. Since the rule for selecting D_k conforms to X for an $\varepsilon > 0$, then there exists a subset of directions D' of D such that $cc(D') = T(X, \hat{x})$, and for any $v \in D'$ and sufficiently large $k \in K$, $p_k + \Delta_k v \in P_k \cup X \cup \Omega$. However, since the poll centers are mesh isolated filter points, it follows that $f(p_k + \Delta_k v) \geq f(p_k)$, and therefore v belongs to $R_f(K)$. Theorem 5.9 completes the proof. ∎

The following corollary to this result shows standard first order optimality conditions for f on X under the additional assumption of strict differentiability.

COROLLARY 6.6. *Let $\hat{x}$ be the limit of a refining subsequence composed of mesh isolated filter points $\{p_k\}_{k \in K}$, and assume that the rule for selecting D_k conforms to X for an $\varepsilon > 0$. If f has a strict derivative $\nabla f(\hat{x})$ at $\hat{x}$, and if $\hat{x}$ is strictly feasible with respect to Ω, then $\nabla f(\hat{x})^T v \geq 0$ for every v in $T(X, \hat{x})$.*

Proof. The result is a direct consequence of Corollary 5.10 and Proposition 6.5. ∎

Note that since it is assumed in Corollary 6.6 that $\hat{x}$ is feasible with respect to Ω, and since the algorithm reduces to the one in [22, 2] in the absence of general constraints, the proof of the corollary also follows from a result in [2].

Our next result does not assume strict feasibility of the limit point. It is a corollary of Corollary 5.10. It gives conditions for the limit point of a refining sequence to satisfy optimality conditions on problem (1.1). It is that the convex conic hull of the union of the refining and the limit directions contains the contingent cone for the feasible region at $\hat{x}$. An interesting aside is that this condition can be met without any feasible descent directions in any poll set. In the simple linearly constrained case, this is implied by ensuring that the polling directions conform to the boundary of X, but here, the corresponding assumption that the polling directions for a refining sequence generate the contingent cone for the feasible region at $\hat{x}$ is not as constructive. This result is illustrated on the three examples of Section 7.

PROPOSITION 6.7. *Let $\hat{x}$ be the limit point of a refining subsequence $\{p_k\}_{k \in K}$, If f has a strict derivative $\nabla f(\hat{x})$ at $\hat{x}$, then $-\nabla f(\hat{x})$ belongs to the polar C_f° of $C_f = cc(R_f(K) \cup L_f(K))$, and so $\hat{x}$ satisfies the optimality conditions of Proposition 5.10 for f on C_f. Moreover, if $T(\Omega \cap X, \hat{x}) \subset C_f$, then $\hat{x}$ satisfies the optimality conditions of Corollary 5.10 for f on Problem (1.1).*

Proof. Let $\hat{x}, f$ and C_f be as in the above statement. Corollary 5.10 guarantees that $\nabla f(\hat{x})^T v \geq 0$ for any vector $v \in C_f$. The results follows from the definition of polarity: In general $-\nabla f(\hat{x}) \in C_f^\circ = \{u \in \mathbb{R}^n : u^T v \leq 0 \ \forall v \in C_f\}$. If $C_f \supseteq T(\Omega \cap X, \hat{x})$, then $C_f^\circ \subset T^\circ(\Omega \cap X, \hat{x})$, and the proof is complete. ∎

REMARK: Notice that under the assumption that the contingent cone generators of the nonlinear constraints binding at $\hat{x}$ belong to the set of refining or limit directions (as will be the case for linear constraints and conforming directions, see Definition 4.1), then the preceding result reduces to the corresponding result from [2, 22]. This is because in that case the contingent cone is the tangent cone, and the polar of the contingent cone is the normal cone so $\hat{x}$ is a KKT point.

By using a filter based step acceptance criterion, we have overcome a difficulty in applying pattern search algorithms to constrained optimization. Specifically, that the objective function descent directions in the positive spanning set D may be infeasible. Lewis and Torczon [20] give an example where a nonfilter version of the pattern search algorithm stalls (*i.e.*, all subsequent iterates are the same mesh isolated filter point), at a point containing a strictly feasible descent direction. The following result shows that, under assumptions on the smoothness of the functions but regardless of the choice of positive spanning set, our algorithm will eventually find an unfiltered mesh point, except when $\nabla f(p_k) = 0$. This is an essential ingredient of any method with ambitions to find more than a single local constrained

minimizer.

PROPOSITION 6.8. *If h and f are both strictly differentiable at the poll center p_k, and if $\nabla f(p_k) \neq 0$, then there cannot be infinitely many consecutive iterations where p_k is a mesh isolated filter point.*

Proof. Let h and f be strictly differentiable at p_k where $\nabla f(p_k) \neq 0$. Assume that there are infinitely many consecutive iterations where p_k is a mesh isolated filter point. Let d be a direction used infinitely often in the (constant) subsequence of poll centers such that $\nabla f(p_k)^T d < 0$.

Since the function h is strictly differentiable at p_k, there exists an $\varepsilon > 0$ such that one of the two conditions is satisfied: either $h_X(p_k + \Delta d) \leq h_X(p_k) < h_{max}$ or $h_X(p_k + \Delta d) > h_X(p_k)$ for all $0 < \Delta < \varepsilon$.

If the first condition is satisfied, then for $\Delta_k < \varepsilon$ the POLL step will find an unfiltered mesh point. This is a contradiction. If the second condition is satisfied, then let $\tilde{h}$ be the smallest value of $\{h_X(x) : h_X(x) > h_X(p_k), x \in \mathcal{F}_k\} \cup \{h_{max}\}$, and let $\tilde{f}$ be the corresponding objective function value, *i.e.*, either $\tilde{f} = f(\tilde{x})$ for some vector $\tilde{x} \in \mathcal{F}_k$ that satisfies $h_X(\tilde{x}) = \tilde{h}$, or $\tilde{f} = -\infty$ in the case that $\tilde{h} = h_{max}$. It follows that $\tilde{h} > h_X(p_k)$ and $\tilde{f} < f(p_k)$. Therefore, whenever $\Delta_k < \varepsilon$ is small enough, the following inequalities hold: $h_X(p_k) < h_X(p_k + \Delta_k d) < \tilde{h}$ and $\tilde{f} < f(p_k + \Delta_k d) < f(p_k)$, thus the trial mesh point is unfiltered. This is a contradiction. ∎

7. Illustration of our results. We now illustrate the behavior of our algorithm on three test examples and on a real engineering problem. The first test example is due to Lewis and Torczon [20]. Unlike the barrier approach in [20], the filter approach can converge even with a badly chosen positive spanning set.

The second example justifies our choice of the squared ℓ_2 norm over the ℓ_1 norm in the definition of the constraint violation function. The non-smoothness of the latter may not provide descent on h_1 in some of the poll directions for which h_2 does descend. The example shows that since h_1 does not allow movement, using it can result in stalling at an infeasible point.

The third example shows the limitations of our results; there is more left to do. This example uses the algorithm's flexibility as a loophole to avoid a desirable outcome. Even with the squared ℓ_2 norm, it is still possible to choose the positive spanning sets, and to be unlucky, in a way that there is a polling direction which is a feasible descent direction for the objective function f from the limit point $\hat{x}$. This does not contradict our results, but it does show their limitations without a suitable SEARCH scheme.

The last example is a wing planform design problem from Boeing for a different airplane than the two airplanes used to generate the results reported in [3].

7.1. Example of Lewis and Torczon. Consider the linear program [20]

$$\min_{x=(a,b)^T} \quad -a - 2b$$
$$\text{s.t.} \quad 0 \leq a \leq 1$$
$$b \leq 0.$$

The optimal solution is $\hat{x} = (1,0)^T$. Let us apply our algorithm with initial point $x_0 = (0,0)^T$ and initial mesh size parameter $\Delta_0 = 1$ and with a single positive spanning matrix $D_k = D$ constructed with the four directions $\pm(1,1)^T$ and $\pm(1,-1)^T$. We will not use any SEARCH step for this example. It is pointed out in [20] that all iterations of a "barrier" pattern search algorithm that assigns an objective function value of $+\infty$ to infeasible points but does not take into consideration the geometry of the feasible region, remain at the origin since the polling directions that yield decrease in the objective function are infeasible.

Suppose that the constraints are given as black boxes, and that the algorithm is not aware that they are linear. Therefore $X = \mathbb{R}^2$ and for any $x \in \mathbb{R}^2$ the value $C(x)$, and thus $h(x) = h_X(x)$ can be computed.

One might consider using the unconstrained GPS on an ℓ_1 exact penalty function for this problem. It turns out that for any penalty constant greater than or equal to 3, the algorithm with the same starting data never moves from the origin. The penalty constant must be greater than 2 for the problem to have the same solution, so the penalty function approach is not useful here.

Our filter algorithm, using the above mentioned spanning set converges to the optimal solution. Mesh directions that conform to the boundary of the feasible region cannot be identified. Figure 7.1 displays the first few iterations. The shaded area is the feasible region. The poll centers are underlined, and the functions values are displayed between brackets: $[h(x), f(x)]$. The points in the poll set are joined to the poll center by dotted lines.
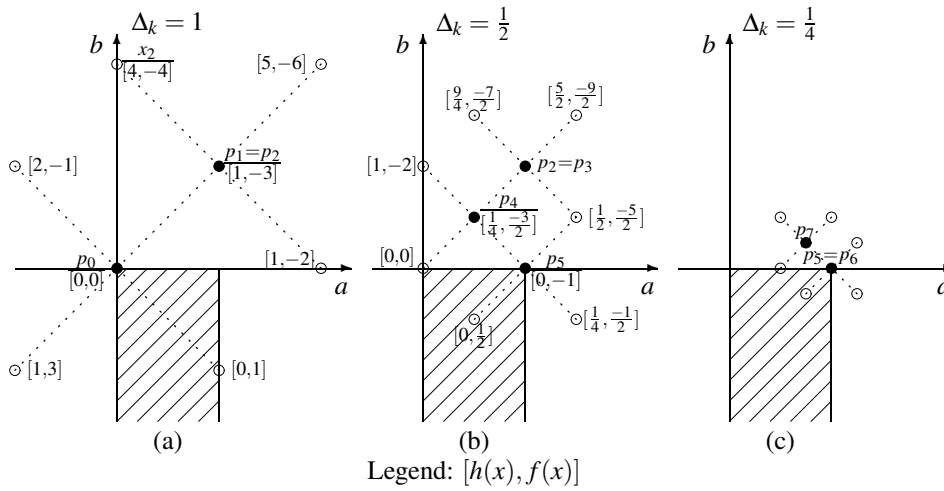

FIG. 7.1. *First iterations on example from Lewis and Torczon.*

Figure 7.1(a) illustrates the iterations for which $\Delta_k = 1$. Starting at $x_0 = p_0 = (0,0)^T$ the algorithm evaluates both functions at $\pm(1,1)^T$ and $\pm(1,-1)^T$. Only the trial point $(1,-1)^T$ is feasible; it is however dominated by x_0. The point $(1,1)^T$ dominates the two other trial points and is unfiltered.

Let $x_1 = p_1 = (1,1)^T$. The functions are evaluated at the four points around p_1 and two unfiltered points are found: $x_2 = (0,2)^T$ and $(2,2)^T$. Even if an unfiltered mesh point was found, the poll center p_2 remains at p_1. Polling around p_2 yields filtered points, thus p_2 is a mesh isolated filter point. Figure 7.2 displays the filters corresponding to the iterates in Figure 7.1. Figure 7.1(b) starts at iteration 3 with $p_3 = (1,1)^T$ and $\Delta_3 = \frac{1}{2}$. Two consecutive iterations where an unfiltered mesh point is found lead to $p_4 = (\frac{3}{2}, \frac{1}{2})^T$ then $p_5 = (1,0)^T$, which is the optimal solution.

However, since the gradient is nonzero at this point, Proposition 6.8 ensures that polling around this point will eventually produce an unfiltered mesh point. Indeed, as shown in Figure 7.1(c), iteration 5 produces a mesh isolated filter point, but iteration 6 generates an unfiltered mesh point, which is a new infeasible incumbent with minimal constraint violation. For this example, the limit point is feasible, and so it is a global optimizer for the constraint
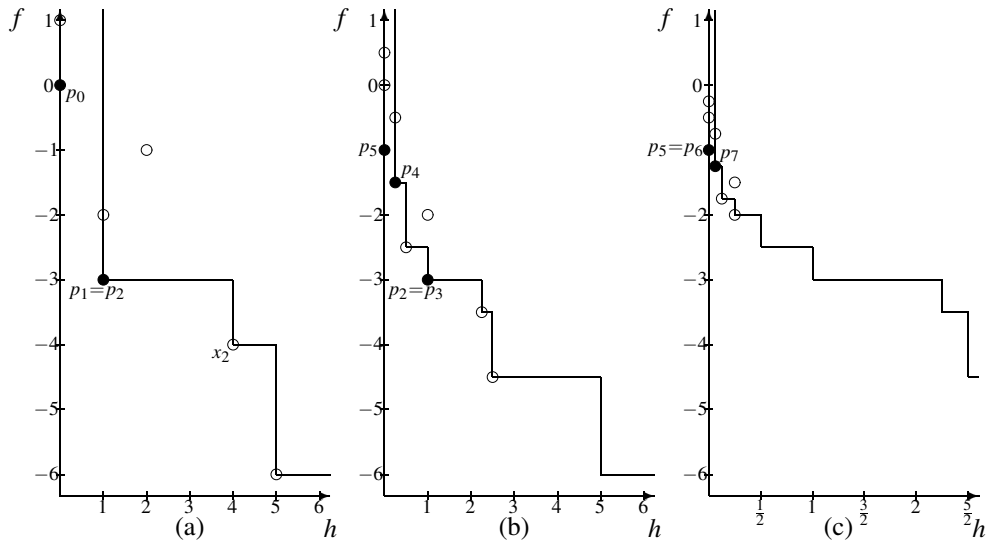
FIG. 7.2. *Filter for the example from Lewis and Torczon.*

violation function. The set of refining and limit directions for f are

$$R_f(K) = \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ -2 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix} \right\}$$

$$L_f(K) = \left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}.$$

The polar of the cone spanned by the refining and limit directions of f is spanned by $\{(1,1)^T, (0,1)^T\}$, and indeed contains $-\nabla f(1,0) = (1,2)^T$ (as stated by Proposition 6.7). Moreover, non-negative combinations of the last two directions of $R_f(K)$ span the contingent cone to the constraints, and therefore the solution satisfies the KKT conditions.

REMARK: Observe that the choice of the poll centers is also important to the quality of the limit points the algorithm finds. Indeed, in this example, if one were to always take the current poll center to be the best feasible incumbent, then the refining sequence will have every term equal to x_0. C_f is again spanned by the same set of directions $R_f(K)$.

Of course, continuing to poll around an unchanging feasible incumbent is a bad idea since it ignores the flexibility of the filter method by reducing it to the barrier method. A better poll center selection strategy could be to alternate between the two incumbents every time the POLL step detects a mesh isolated filter point. Polling around the infeasible one with minimal constraint violation is especially interesting when f^I is less than f^F since it might move the iterates away from a local optimum, or toward a more interesting part of the feasible region. That way, there will be infinitely many poll steps around both types of incumbents. It is also worthwhile to change the positive spanning set to enrich the set of refining directions for both h and f. The flexibility of our theory ensures that such heuristics can be part of a rigorously convergent algorithm.

7.2. Choice of the constraint violation norm. The choice of the norm in the definition of the constraint violation function $h(x) = \|C(x)_+\|$ affects the convergence behavior. The example presented here complements the theoretical results of section 5.2. We prefer the squared ℓ_2 norm over the ℓ_1 norm since it is differentiable whenever the constraint function C is (see [11] for an explicit formulation of the gradient). This means that if there is a descent

direction, then a positive spanning set will detect it with the ℓ_2 norm, but ℓ_1 might miss it (see Corollary 6.3). This is illustrated in the following simple linear program

$$\min_{x=(a,b)^T} b$$
$$\text{s.t.} \quad -b \le 3a \le b$$
$$b \ge 1$$

with ℓ_1 constraint violation function $h_1(a,b) = \max(3a-b,0) + \max(-3a-b,0) + \max(1-b,0)$.

Let the algorithm start at the infeasible point $x_0 = p_0 = (0,0)^T$, and let the positive spanning set be $D = \{(1,1)^T, (1,-1)^T, (0,-1)^T\}$. The iterates and the filter are depicted in Figure 7.3.
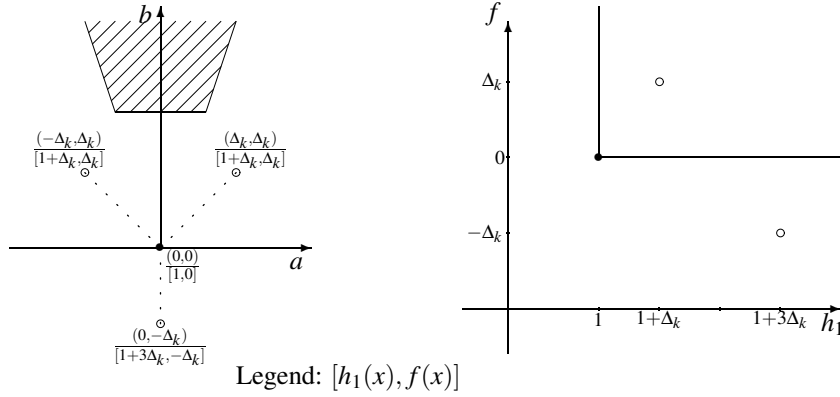

Legend: $[h_1(x), f(x)]$

FIG. 7.3. *The algorithm stalls with the ℓ_1 norm.*

Every even iteration k produces an unfiltered mesh point that does not improve any of the incumbents: The trial point $x_{k+1} = p_k + \Delta_k(1,1)^T$ is unfiltered by $\mathcal{F}_k$. Every odd iteration confirms that the poll center is a mesh isolated filter point: The three trial points are filtered since $p_{k+1} = p_k = (0,0)^T$. Therefore, the mesh size parameter is reduced at each odd iteration.

The sequence of poll centers stalls at the infeasible point $\hat{x} = (0,0)^T$. This means that the non-differentiability of h_1 hides the descent directions for the constraint violation function. It also means that this is another example where the unconstrained GPS algorithm applied to the ℓ_1 exact penalty function fails as a solution approach. However, as guaranteed by Theorem 5.9, $h^\circ(\hat{x}, v)$ is nonnegative for the positive spanning directions in D as well as for other refining and limit directions for h:

$$R_h(K) = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right\}$$
$$L_h(K) = \left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}.$$

The sets of refining and limit directions for f are

$$R_f(K) = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right\}$$
$$L_f(K) = \left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right\}.$$

The polar of the cone spanned by the refining and limit directions for f reduces to the negative gradient of f: $\{(0,-1)^T\}$.

REMARK: If the squared ℓ_2 norm is used for h instead of ℓ_1, then the poll center moves away from $(0,0)$ to $(\Delta_k, \Delta_k)^T$ as soon as the mesh size parameter drops below $\frac{2}{3}$ since $0 < h(\Delta_k, \Delta_k) < 1$ whenever $0 < \Delta_k < \frac{2}{3}$. The set of refining and limit directions for f are the same as above but the algorithm converges to a global optimal solution.

7.3. Illustration of the limitation of the results. Consider the problem

$$\min_{x=(a,b)^T} \quad b$$
$$\text{s.t.} \quad a(1-a) - b \leq 0.$$

The algorithmic strategies described below are such that the algorithm goes through infinitely many consecutive cycles of three iterations, and the sequence of iterates converge to a feasible limit point containing a feasible descent direction used infinitely often by the iterates. The first iteration of each cycle improves the feasible incumbent, the second one improves the least infeasible incumbent, and the last one produces a mesh isolated filter point. We admit that the flexibility in the choice of polling directions is exploited to lead to a weak result, but our point is that it can happen.

The trial points generated during cycle ℓ are summarized in Table 7.1. The algorithm does not perform any SEARCH, and complete polling is always performed. The table also displays the positive spanning directions used at each POLL step. The initial points in cycle $\ell = 1$ are $p_0 = x^I = (\frac{1}{4}, 0)^T$ and $x^F = (0,1)^T$, and the initial mesh size parameter is $\Delta_0 = \frac{1}{8}$.

TABLE 7.1

Description of the three iterates of cycle ℓ with initial incumbents $x^F = \left(0, \frac{1}{2^{\ell+1}}\right)^T$ and $x^I = \left(\frac{1}{2^{\ell+1}}, 0\right)^T$ and function values $[h(x^F), f(x^F)] = \left[0, \frac{1}{2^{\ell+1}}\right]$ and $[h(x^I), f(x^I)] = \left[\frac{2^{\ell+1}-1}{4^{\ell+1}}, 0\right]$.

Mesh Size (Δ_k)	Poll Center (p_k)	$[h,f]$	Poll Dirs (D_k)	Trial Points	$[h,f]$	Comments
			$(-2,1)$	$\left(0, \frac{1}{2^{\ell+2}}\right)$	$0, \frac{1}{2^{\ell+2}}$	x^F is improved
$\frac{1}{2^{\ell+2}}$	$\left(\frac{1}{2^{\ell+1}}, 0\right)$	$\left[\frac{2^{\ell+1}-1}{4^{\ell+1}}, 0\right]$	$(2,0)$	$\left(\frac{1}{2^\ell}, 0\right)$	$\frac{2^\ell-1}{4^\ell}, 0$	filtered by poll center
			$(0,-1)$	$\left(\frac{1}{2^{\ell+1}}, \frac{-1}{2^{\ell+2}}\right)$	$\frac{3\times 2^\ell-1}{4^{\ell+1}}, \frac{-1}{2^{\ell+2}}$	unfiltered
			$(0,-2)$	$\left(0, \frac{-1}{2^{\ell+2}}\right)$	$\frac{1}{2^{\ell+2}}, \frac{-1}{2^{\ell+2}}$	unfiltered
$\frac{1}{2^{\ell+2}}$	$\left(0, \frac{1}{2^{\ell+2}}\right)$	$\left[0, \frac{1}{2^{\ell+2}}\right]$	$(1,-1)$	$\left(\frac{1}{2^{\ell+2}}, 0\right)$	$\frac{2^{\ell+2}-1}{4^{\ell+2}}, 0$	x^I is improved
			$(-1,2)$	$\left(\frac{-1}{2^{\ell+2}}, \frac{3}{2^{\ell+2}}\right)$	$0, \frac{3}{2^{\ell+2}}$	filtered by poll center
			$(1,0)$	$\left(\frac{1}{2^{\ell+1}}, 0\right)$	$\frac{2^{\ell+1}-1}{4^{\ell+1}}, 0$	already in filter
$\frac{1}{2^{\ell+2}}$	$\left(\frac{1}{2^{\ell+2}}, 0\right)$	$\left[\frac{2^{\ell+2}-1}{4^{\ell+2}}, 0\right]$	$(-1,1)$	$\left(0, \frac{1}{2^{\ell+2}}\right)$	$0, \frac{1}{2^{\ell+2}}$	already in filter
			$(-1,-1)$	$\left(0, \frac{-1}{2^{\ell+2}}\right)$	$\frac{1}{2^{\ell+2}}, \frac{-1}{2^{\ell+2}}$	already in filter

Figure 7.4 displays the first cycle (polling around the poll centers p_0, p_1 and p_2) and the corresponding filter. Cycle 1 terminates with a mesh isolated filter point, and cycle 2 is initiated at $p_3 = (\frac{1}{8}, 0)^T$ with $\Delta_3 = \frac{1}{16}$. More generally, cycle ℓ terminates with a mesh isolated filter point. The mesh size parameter is divided by 2, and cycle $\ell + 1$ starts.
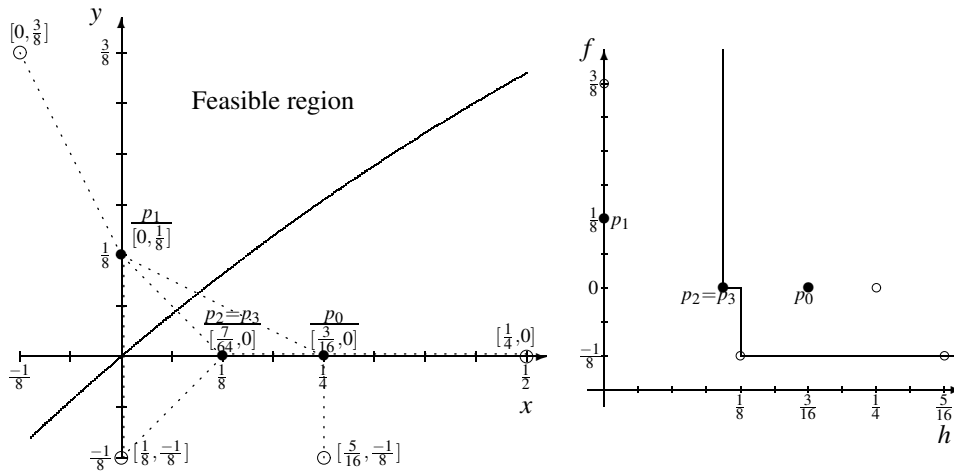
FIG. 7.4. *The first cycle.*

All trial points including the sequence of mesh isolated filtered points, *i.e.*, the iterates corresponding to the third step of the cycles, converge to the feasible point $\hat{x} = (0,0)^T$. There are no other limit points. The results of Section 6.1 concerning the constraint violation function are clearly satisfied since the limit point is at the global minimum of h. However, there is a feasible direction from the limit point used infinitely often by the subsequence, which is also a descent direction for the objective function. The sets of refining and limit directions for f are

$$R_f(K) = \left\{ \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

$$L_f(K) = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \frac{1}{\sqrt{10}} \begin{bmatrix} -1 \\ 3 \end{bmatrix} \right\}.$$

The polar of the cone spanned by these directions is the cone spanned by the single direction $(0,-1)^T$, which is the gradient at the origin. Thus, Proposition 6.7 is again sharp. The contingent cone at the origin is the half space $a - b \leq 0$, and the intersection of the contingent cone at the origin with C_f, the cone generated by convex conic hull of $R_f(K) \cup L_f(K)$, is the convex conic hull of $(-2,0)^T$ and $(1,1)^T$.

7.4. Filter results on a Boeing planform design application. The GPS filter algorithm has been applied often to Boeing wing planform design problems. The wing planform is the two-dimensional, downward, vertical projection of the wing. The design variables are the line segment end point for the wing leading edges, trailing edges, and spars. Also there are variables related to wing thickness and aerodynamic loading [3]. A typical design problem is to minimize direct operating cost subject to several constraints. The constraints include required range, maximum approach velocity, maximum required runway length, and several others. The analysis code is a sophisticated combination of preliminary design tools from many disciplines. The disciplines include structures, aerodynamics weights, costing, and configuration management.

This problem has 17 variables, 13 nonlinear constraints, and no linear constraints. The best point in the initial surrogate (a kriging model that interpolates data from 200 points

obtained from an orthogonal-array-based Latin hypercube) is the least infeasible point, which has a constraint violation of 0.426 and an objective of 9.845.
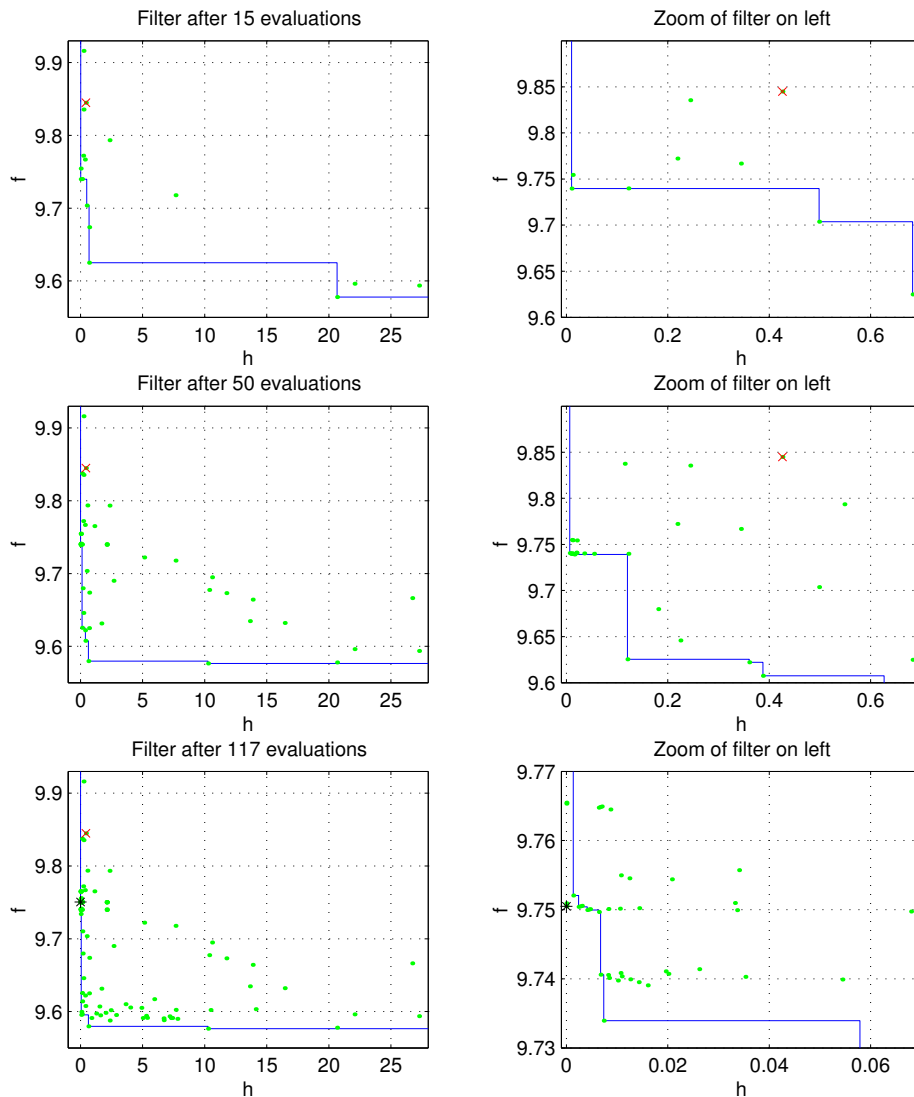

FIG. 7.5. *Filter progression on a Boeing planform design application*

Figure 7.5 illustrates the progression of the filter for this application. In all plots, the symbol $\times$ represents the initial point, except in the bottom right plot due to the scale change. The top two plots correspond to the first 15 function evaluations, the middle ones the first 50 and the bottom ones are the whole filter after completing the 117 evaluations after the initial 200. The initial point gets filtered at the 3rd function evaluation. The first feasible point is found at the 58th evaluation. The best feasible point is denoted on the two bottom plots by a star at $(0, 9.75)^T$.

The bottom left plot contains several trial points with an objective function value near 9.6. This suggests that the SEARCH strategy tried, but was unsuccessful in finding a feasible

design with such a low f value.

7.5. Discussion. Though the algorithm behaved very well on the industrial design problem above (as well as on those in [3] and others), at first glance, one might be unimpressed by the behavior of the algorithm on the academic examples of the last section. Of course, they were designed to illustrate the tightness of our convergence results, and the crucial directional dependence of GPS methods. Our interest is for optimization problems, like the planform problem, where derivative based methods are impractical. Our algorithm can only rely on function values, and sometimes even these values are not reliable. A design example is presented in [4, 5] where two times out of three the evaluation of the objective function failed to produce a value.

A consequence of this absence of structure is that the convergence results that are guaranteed depend in part on the set of directions used in the POLL step. Indeed, the richer the set of directions, the stronger the convergence result, since adding directions can increase the number of refining and limit directions and widen the cone C_f of Proposition 6.7, and hence narrow its polar cone where the negative gradient of the objective is shown to reside. Intuitively, if a POLL step identifies a poll center that is a mesh isolated filter point, then, the next time a POLL is performed there (with a reduced mesh size parameter) it would be natural to use a different positive spanning set to increase the likelihood of detecting an eventual descent direction. However, essential to the convergence proof, is a finite total number of polling directions. It follows that one cannot attempt to obtain a dense set of polling directions.

In practice, one would never use a pattern search algorithm following the rules upon which these examples are based. First, we would use a SEARCH step such as a space-filling latin hypercube sampling, or a surrogate based exploration or a more local SEARCH such as the type suggested in [12]. Second, the set of polling directions would be enlarged in order to avoid large gaps in the directions explored. Finally, the polling centers would sometimes be the feasible incumbent, and sometimes the infeasible one with least constraint violation value, but when promising filter points are generated (such as one with low f and h values), nothing stops the SEARCH step from including an unofficial POLL around these candidates as a part of the search. These simple algorithmic enhancements fit in the general description of the algorithm presented in Section 4.3. Even with these improvements one could devise twisted examples with the behavior of the above examples. It is however unlikely that such behavior would be encountered in practice.

REFERENCES

[1] AUDET C. (2002), "Convergence results for pattern search algorithms are tight," *Les Cahiers du GERAD* G-2002-56, Montréal.

[2] AUDET C. and DENNIS J.E.JR.(2003), "Analysis of generalized pattern searches," *SIAM Journal on Optimization* 13, 889-903.

[3] AUDET C., BOOKER A.J., DENNIS J.E.JR., FRANK P.D., and MOORE D.(2000), "A Surrogate-Model-Based Method For Constrained Optimization", AIAA no.4891, *Proceedings of the Symposium on Multidisciplinary Analysis and Optimization.*

[4] BOOKER A.J., DENNIS J.E.JR, FRANK P.D., SERAFINI D.B., TORCZON V. and TROSSET M.W.(1999), "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization* Vol.17 No.1, 1-13.

[5] BOOKER A.J., DENNIS J.E.JR, FRANK P.D., MOORE D.W. and SERAFINI D.B.(1999), "Managing Surrogate Objectives to Optimize a Helicopter Rotor Design - Further Experiments," AIAA Paper 98-4717, St. Louis, September 1998.

[6] CHOI T.D., ESLINGER O.J., KELLEY C.T., DAVID J.W., ETHERIDGE M.(1998), "Optimization of automotive valve train components with implicit filtering," to appear in *Optimization and Engineering*.

[7] CHOI T.D., KELLEY C.T.(1999), "Superlinear convergence and implicit filtering," *SIAM Journal on Optimization* Vol.10 No.4, 1149–1162.

[8] CLARKE, F.H.(1990) "Optimization and Nonsmooth Analysis" SIAM Classics in Applied Mathematics Vol.5, Philadelphia.

[9] CONN A.R., GOULD N.I.M. and TOINT PH.L.(1991) "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis* 28, 545–572.

[10] COOPE I.D., PRICE C.J. (2001), "On the convergence of grid-based methods for unconstrained optimization," *SIAM Journal on Optimization* 11, 859–869.

[11] DENNIS J.E.JR, EL-ALEM M. and WILLIAMSON K.(1999), "A trust-region approach to nonlinear systems of qualities and inequalities," *SIAM Journal on Optimization* Vol.9 No.2, 291–315.

[12] DENNIS J.E., TORCZON V.(1991), "Direct Search Methods on Parallel Machines," *SIAM Journal on Optimization* Vol 1, pp. 448–474.

[13] ESCHENAUER, H., KOSKI, J., OSYCZKA, A. (EDS)(1990), "Multicriterion Design Optimization," Springer, Berlin.

[14] FLETCHER R. and LEYFFER S.(2002), "Nonlinear programming without a penalty function," *Mathematical Programming* 91, 239–269.

[15] FLETCHER R, LEYFFER S., TOINT PH.L.(2002), "On the global convergence of an SQP-Filter algorithm," *SIAM Journal on Optimization*, Vol.13, No.1, 44-59.

[16] FLETCHER R, GOULD N.I.M., LEYFFER S., TOINT PH.L. and WÄCHTER(2002), "On the global convergence of trust-region SQP-Filter algorithms for general nonlinear programming," *SIAM Journal on Optimization*, Vol.13, No.3, 635–659.

[17] GILMORE P, KELLEY, C.T. (1995), "An implicit filtering algorithm for optimization of functions with many local minima," *SIAM Journal on Optimization*, Vol. 5, 269–285.

[18] JAHN J.(1994), "Introduction to the Theory of Nonlinear Optimization," Springer, Berlin.

[19] LEACH E.B.(1961) "A note on inverse function theorem," Proc. AMS Vol.12, 694-697.

[20] LEWIS R.M. and TORCZON V.(1999), "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, Vol.9 No.4, 1082-1099.

[21] LEWIS R.M. and TORCZON V.(1996), "Rank ordering and positive basis in pattern search algorithms," *ICASE NASA Langley Research Center* TR 96-71.

[22] LEWIS R.M. and TORCZON V.(2000), "Pattern search methods for linearly constrained minimization," *SIAM Journal on Optimization*, Vol.10 No.3, 917–941.

[23] LEWIS R.M., TORCZON V.(2002), "A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Optimization*, Vol.12 No.4, 1075-1089.

[24] ROCKAFELLAR R.T.(1968), *Convex Analysis,* Princeton University Press Mathematical Series No.28, Princeton

[25] STEPHENS C.P. and BARITOMPA W.(1998), "Global optimization requires global information," *Journal of Optimization Theory and Applications* Vol.96 No.3, 575–588.

[26] TORCZON V.(1997), "On the convergence of pattern search algorithms," *SIAM Journal on Optimization* Vol.7 No.1, 1–25.