Convergence Examples of a Filter-Based Evolutionary Algorithm

Lauren M. Clevenger[1] and William E. Hart[2]

[1] University of New Mexico, lmcleve@aol.com
[2] Sandia National Laboratories, Discrete Mathematics and Algorithms Dept., P. O. Box 5800, MS 1110, Albuquerque, NM 87185-1110; Phone: 505-844-2217 Fax: 505-845-7442; wehart@cs.sandia.gov; http://www.cs.sandia.gov/~wehart/

Abstract. We describe and critique the convergence properties of filter-based evolutionary pattern search algorithms (F-EPSAs). F-EPSAs implicitly use a filter to perform a multi-objective search for constrained problems such that convergence can be guaranteed. We provide two examples that illustrate how F-EPSAs may generate limit points other than constrained stationary points. F-EPSAs are evolutionary pattern search methods that employ a finite set of search directions, and our examples illustrate how the choice of search directions impacts an F-EPSA's search dynamics.

1 Introduction

Although evolutionary algorithms (EAs) have been successfully applied to many unconstrained optimization applications, the investigation of constrained EAs has received far less attention [6]. Despite this, handling constraints in EAs is necessary for their application to many problem domains. Thus the development of provably robust EAs is crucial to ensure that these methods can be effectively applied to a wide range of problems, including linear, non-linear, equality and inequality constraints [4].

Of the many different constraint-handling techniques used with EAs, the most common are penalty functions. Although penalty functions can have good convergence properties for specific problems, some penalty functions require an initial feasible solution that must be provided by the user and penalty approaches may also require extra parameters that can be hard to choose, especially when they are problem-dependent. Alternatives to penalty functions tend to be developed for very specific problems and problems in which estimating good penalty functions and generating even a single feasible solution are difficult. Some of the techniques surveyed by Coello [6] include approaches that use problem-specific representations and operators, algorithms that repair infeasible points to make them feasible, and approaches that separate objectives and constraints (e.g. multi-objective optimization techniques). Unfortunately, these methods sometimes have difficulty preserving diversity and avoiding stagnation. Additionally, some of these approaches require the generation of an initial feasible point (or population), which is often NP-hard [13].

Considering all of these challenges for handling constraints in EAs, an approach that minimizes these difficulties and still maintains good convergence results is very desirable. We propose filter-based evolutionary algorithms (FEAs), which use a constraint-handling technique that is similar to multi-objective EAs. The optimization problem that we consider is

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & f(x) \\ \text{s.t.} \quad & C(x) \leq 0 \\ & l \leq x \leq u \end{aligned}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R} \cup \{\infty\}$ and $C : \mathbf{R}^n \rightarrow (\mathbf{R} \cup \{\infty\})^m$ are the constraint functions with $C = (C_1, \ldots, C_m)^T$; $u, l \in \mathbf{Q}^n$ define upper and lower bounds on each dimension.

Clevenger, Ferguson and Hart [8,4] have recently developed a filter-based evolutionary pattern search algorithm (F-EPSA) for constrained optimization. This F-EPSA is closely related to pattern search methods that do not attempt to estimate a derivative in its search process. The main qualitative difference between pattern search methods and common real-coded EAs is that pattern search methods restrict the search in each iteration to a finite pattern of trial points, while most real-coded EAs employ continuous random variables to generate mutation steps. However, this restriction provides mathematical leverage for demonstrating convergence of pattern search methods, which has been effectively translated to EAs [8,11].

We evaluate the convergence behavior of the F-EPSA defined by Clevenger, Ferguson, and Hart [4,8]. This F-EPSA uses a pattern of search steps so the search behavior is dependent on this pattern. We provide two examples that illustrate this dependence. In both cases the F-EPSA converges to a limit point, but the limit points are not constrained stationary points. The following section describes F-EPSAs. Section 3 discusses these two examples.

2 Algorithmic Formulation

A filter-based optimizer uses a nonnegative continuous function to aggregate the constraint violations and then treats the resulting bi-objective problem (e.g., see Fletcher et al. [9,10]). In other words, a filter-based optimizer tries to minimize both the objective function and the aggregate constraint violation function simultaneously. Since a feasible solution is desired, priority is usually given to the aggregate function until a feasible solution is found. We give two definitions that are very similar to those stated in Audet and Dennis [1], which will be used throughout this paper.

Definition 1 *Given objective functions $f_i(), i = 0, \ldots, k$, if $f_i(x_1) \leq f_i(x_2)$ for every $i \in 0, \ldots, k$ and there is at least one j such that $f_j(x_1) < f_j(x_2)$, then x_1 is said to* dominate *x_2. This is denoted by $x_1 \prec x_2$. Also, $x_1 \preceq x_2$ denotes that either $x_1 \prec x_2$ or $x_1 = x_2$.*

Definition 2 *Given a set of points S, a point $s \in S$ is said to be a* non-dominated point *if there does not exist $x \in S$ such that $s \prec x$.*

A *filter* F is a (finite) set of points in $\mathbf{R}^n$ such that no pair x, x' in the filter are in the relation $x \prec x'$. That is, no point in F dominates or is dominated by any other point in F. Filter-based optimizers employ a filter that is used to eliminate trial points from consideration if they are dominated by points in the filter either by having a worse function value or worse aggregate constraint violation.

2.1 A Filter-Based EA

Figure 1 presents the basic steps of Algorithm A, the F-EPSA introduced by Clevenger and Hart [8]. This EA evolves a set of points $W_t = Y_t \bigcup X_t$, where Y_t are infeasible and X_t are feasible. We say $x \prec x'$ if and only if $(f(x), h(x)) \prec (f(x'), h(x'))$, where $f(x)$ is the objective function and $h(x) = \sum_{i=1}^{m} \max[0, C_i(x)]^2$. Note that $h(x) = \infty$ if any of the constraint function values at x are infinite.

This F-EPSA implicitly uses a filter, which is the subset of W_t containing the best non-dominated infeasible solutions Y_t and the best feasible solution x_t^*. Let x_t^* be the point in X_t with the best function value, and y_t^* be the point in Y_t with the minimal constraint violation (as defined by h). If two points have the same minimal constraint violation, y_t^* is the point with the minimal function value.

Intuitively, a point is locally optimal if it cannot be improved. In the context of an F-EPSA, a point cannot be improved if the mutation steps about it are dominated by the points in the filter contained in W_t.

Suppose that all points in $\mathcal{N}(x, \Delta_t, D_t) = \{x + \Delta_t d \mid d \in D_t\}$ have been generated. If either x_t^* or y_t^* is in this set, then x is locally optimal if $(f(x), h(x))$ equals $(f(x_t^*), h(x_t^*))$ or $(f(y_t^*), h(y_t^*))$, so generating this mutation step does not give a simple improvement in either x_t^* or y_t^*. Suppose that neither x_t^* nor y_t^* is in $\mathcal{N}(x, \Delta_t, D_t)$. Then all of the points in $\mathcal{N}(x, \Delta_t, D_t)$ are dominated by either x_t^* or y_t^*. Further, in subsequent iterations these values will only improve so for all $t' > t$, the points in $\mathcal{N}(x, \Delta_t, D_t)$ are dominated by either $x_{t'}^*$ or $y_{t'}^*$

The following provides further details about the definition of Algorithm A:

- X_1 and Y_1 could be simply initialized by randomly generating P points within the bound constraints, and then applying the standard update rule. However, in practice this initialization could exploit domain knowledge of the structure of the constraints.
- D is a finite set of mutation offsets that can be applied. All subsets $D_t \subseteq D$ must be selected to ensure that D_t is a positive spanning set (i.e. non-negative linear combinations of points in D_t generate $\mathbf{R}^n$).
- The determination of whether x_{t+1}^* or y_{t+1}^* is locally optimal is not made with respect to the current population W_t. Instead, this requires the explicit cataloging of the history of mutation steps about these points.

```
Given $\Delta_0$, $\tau > 1$ ($\tau \in Q$) and mutation directions $D$
Randomly initialize $X_0$ and $Y_0$; $W_0 = X_0 \bigcup Y_0$
Select $D_0 \subseteq D$
For $t = 0, \ldots, \infty$
    For $j = 1, \ldots, P$
        Randomly select $d \in D_t$ and $w \in W_t$
        $\hat{w}_j = \Delta_t d + w$
        Evaluate $\hat{w}_j$
    End For
    Update $X_{t+1}, Y_{t+1}$; $W_{t+1} = X_{t+1} \bigcup Y_{t+1}$
    Update $x^*_{t+1}$ and $y^*_{t+1}$
    If  $(f(x^*_{t+1}) < f(x^*_t))$ or
        $(h(y^*_{t+1}) < h(y^*_t))$ or
        $((h(y^*_{t+1}) = h(y^*_t))$ and $(f(y^*_{t+1}) < f(y^*_t)))$ Then
        $\Delta_{t+1} = \Delta_t \tau^\nu$, where $0 \leq \nu \leq \nu_{max}$
        Select $D_{t+1} \subseteq D$
    Else If $x^*_{t+1}$ or $y^*_{t+1}$ is locally optimal Then
        $\Delta_{t+1} = \Delta_t \tau^\nu$, where $\nu_{\min} \leq \nu < 0$
        Select $D_{t+1} \subseteq D$
    Else
        $\Delta_{t+1} = \Delta_t$ and $D_{t+1} = D_t$
    Terminate if $\Delta_{t+1} < \Delta_{min}$
End For
```

Fig. 1. Pseudo-code for Algorithm A. For simplicity, we have not included the checks to see if either x^*_{t+1} or y^*_{t+1} are not defined because a feasible or infeasible point has not been encountered by iteration $t+1$. These checks would be used in all of the conditional statements after x^*_{t+1} and y^*_{t+1} are updated.

– Algorithm A updates the step length Δ_t by (a) possibly increasing it if some new point dominates either x^*_t or y^*_t, or (b) decreasing it if x^*_{t+1} or y^*_{t+1} are locally optimal (and thus no progress can be made about these points using D_t).
– Algorithm A terminates if the step length shrinks below some predetermined threshold, which is the termination rule commonly used with pattern search methods.

2.2 Related Constrained EAs

The concept of a filter is directly analogous to the notion of an *archive* of pareto optimal solutions, which has been used in a wide range of evolutionary algorithms (e.g., see Knowles and Corne [12]). The method of constraint handling proposed here shares some commonalities with a few of the techniques surveyed by Coello [6]. Since Algorithm A separates constraints from objectives, it is most similar to approaches that also use this separation. Consider the Similarity of Feasible Points technique proposed by Deb [7]. Deb gives three rules for comparing points:

1. A feasible point is always preferred over an infeasible one.
2. Between two feasible points, the one having a better objective function value is preferred.
3. Between two infeasible points, the one having a smaller constraint violation is preferred.

Deb's method also includes a selection procedure that only performs pairwise comparisons so that no penalty factor is required [6]. Similarly, Algorithm A performs pairwise comparison for selection and follows rules 2 and 3 of Deb's method. It does not necessarily follow the first rule because we want to keep infeasible points to ensure a robust search.

Algorithm A is also similar to some of the multi-objective optimization techniques surveyed by Coello [5]. The most closely related technique is the one proposed by Camponogara and Talukdar [2]. Their procedure restates a single optimization problem to consider two objectives: the optimization of the original objective function and the optimization of

$$\Phi(x) = \sum_{i=1}^{n} \max[0, C_i(x)].$$

Thus Φ is the analogue of h using the L_1 norm instead of the squared L_2 norm. Camponogara and Talukdar use pareto sets (implicitly using a filter) to impose dominance-based selection, which is used to estimate new search directions. The technique we propose implicitly uses a filter to impose dominance-based selection, but it is not used to generate new search directions. Instead, the filter is used to determine when step lengths are expanded and contracted (by imposing conditions for local optimality).

3 Convergence Analysis

Although Algorithm A is quite similar to several existing EAs, the structure of this F-EPSA ensures that with probability one, some subsequence of the points $\{x_t^*, y_t^*\}$ generated by Algorithm A provably converges. Let X_t and Y_t be the stochastic processes, defined on some probability space (Ω, F, P), that describe the behavior of Algorithm A for some problem and for some set of algorithmic parameters. We make the standard assumption that the processes X_t and Y_t generate points that lie in a compact set.

Ferguson and Hart [8] summarize a convergence theory for which Algorithm A generates a convergence subsequence with probability one. They assume that f is strictly differentiable at the limit point, which implies that $\nabla f(x)$ exists and

$$\nabla f(x)^T \omega = lim_{y \to x, t \downarrow 0} \frac{f(y + t\omega) - f(y)}{t}$$

for all $\omega \in \mathbf{R}^n$ [3]. If the limit point is strictly feasible, then the limit point is a first-order stationary point. Otherwise, the algorithm may converge to a constrained local optimimum for a problem that is implicitly defined by the set of search directions in D. Let $\hat{x}$ be a limit point of a convergent subsequence generated by Algorithm A. A convergent subsequence (for some set of indices K) is said to be refining if $lim_{k \in K} \Delta_t = 0$. Ferguson and Hart [8] describe the following result, which illustrates how an L-EPSA converges near a constraint boundary.

Theorem 1. *Let $\hat{x}$ be a limit point of a refining subsequence generated by Algorithm A that is not strictly feasible. Let $D' \subseteq D$ be the set of all the associated directions of all the refining subsequences that converge to the limit point $\hat{x}$ in such a manner that the constraint violation is constant. If f is strictly differentiable at $\hat{x}$, then $\nabla f(\hat{x})$ belongs to the polar of the cone generated by D'. If h is strictly differentiable at $\hat{x}$ then $\nabla h(\hat{x}) = 0$.*

Theorem 1 is not quite as strong as would be desired, since it does not guarantee convergence to a first-order constrained stationary point. In particular, this result depends on the set of search directions D that are defined, since this ultimately limits the cone that contains $\nabla f(\hat{x})$. Thus Algorithm A will perform a more robust search for constrained local minima as the number of search directions in D is increased.

None of these results ensures convergence to a globally optimal feasible point. It is not clear that such a convergence theory exists for methods like Algorithm A that dynamically adapt their search step lengths without imposing fundamental limitations on their adaptive dynamics (e.g. lower bounds on the step lengths). Our analysis provides insight into mechanisms that facilitate robust local convergence without concern of the global search dynamics. However, the efficacy of the global search is clearly influenced by the algorithmic choice, and we expect that methods like Algorithm A will perform a more global search than the pattern search methods discussed by Audet and Dennis [1].

The following examples illustrate the implications of Theorem 1 on two test problems. Specifically, these examples illustrate how the choice of search directions can limit the ability of Algorithm A to converge to constrained stationary points. In all of our examples we consider the case where Algorithm A is used with a single search pattern throughout the search, which is consistent with the manner in which most pattern search methods are employed. We discuss this point further in the next section.

3.1 Example I

Consider the problem

$$\begin{aligned} &\min -ab \\ &\text{s.t. } \; a^2 + b^2 \leq 16 \\ &\qquad -4 \leq a, b \leq 4 \end{aligned}$$

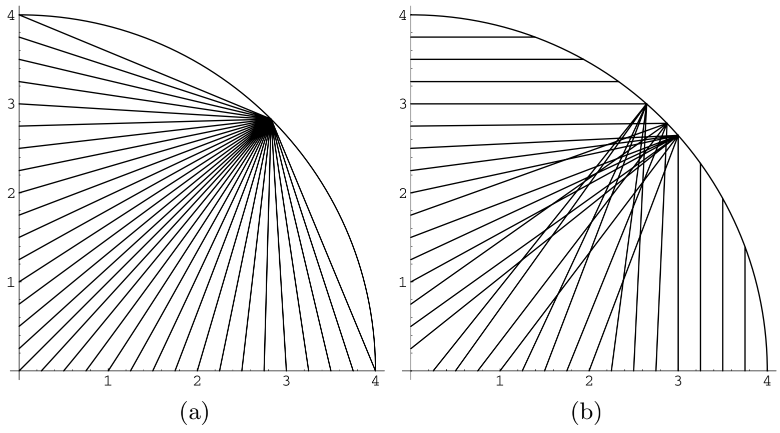

Fig. 2. Illustration of the convergence of Algorithm A for Example I when (a) $D_t = D = \{\pm(1,1), \pm(1,-1)\}$ and (b) $D_t = D = \{\pm(1,0), \pm(0,1)\}$. The two axes represent the coordinates of the solutions in this two-dimensional space. The lines in these figures connect the initial point and final feasible point. The lack of symmetry in (b) is due to the fact that ties are broken arbitrarily.

for which the optimal solution is $x^* = (a^*, b^*) = (2\sqrt{2}, 2\sqrt{2})$. Consider the behavior of Algorithm A using the search pattern $D_t = D = \{\pm(1,1), \pm(1,-1)\}$. We simplify our presentation by assuming that $\mu_F = \mu_I = 1$ and $P = 8$, so all mutation steps are generated in each iteration. We consider feasible starting points, so in initial iterations P is effectively equal to 4.

Figure 2a illustrates the convergence behavior of Algorithm A when started from a set of points along the x- and y-axes. The lines in this figure connect an initial point and final feasible point, and it is clear that in every case the F-EPSA converges to the optimal solution. Note that $-\nabla f(x) = (b, a)$. Now suppose that Algorithm A generates a limit point $\hat{z} = (a, b)$ on the constraint boundary for which $b > a$. It follows that the directions $(-1,-1)$ and $(1,-1)$ are the associated directions at this limit point (because the constraint violation function is constant in these directions), and they define the cone C_s. The polar cone C_s^o is defined by the directions $(-1,1)$ and $(1,1)$. However, at $\hat{z}$ we have $-\nabla f(\hat{z}) = (b, a)$ which is not in this cone if $b > a$ (note that $(-1,1)^T(b,a) = a - b < 0$). Consequently, the only limit point that Algorithm A could generate that satisfies the conditions of Theorem 1 is the point (a^*, b^*).

The contrast between these two search patterns highlights the degree to which the choice of search pattern can impact how closely Algorithm A converges to constrained stationary points. If the search pattern is selected well, you may be able to ensure that a constrained stationary point is generated, but if the search pattern is selected poorly then any point on the nonlinear constraint boundary may be a limit point. Furthermore, it is clear that if the search pattern $D_t = D = \{\pm(1,1), \pm(1,-1)\}$ were perturbed slightly then this F-EPSA could converge to points other than the constrained stationary point. Consequently, this method may be sensitive to numerical instabilities such as round-off errors.

3.2 Example II

Consider the problem

$$\begin{array}{rl} \min & f(a,b) \\ \text{s.t.} & \frac{1}{5}a+\frac{4}{5}\geq b \\ & 5a-4\leq b \\ & 0\leq a,b \end{array}$$

where $f:\mathbf{R}^2\rightarrow\mathbf{R}$ is an arbitrary function. The solution to this problem lies within the feasible region, but we consider the convergence of Algorithm A starting from an initial point (λ,λ) for some $\lambda>1$. The following analysis shows that Algorithm A converges to the point $(1,1)$ on the constraint boundary, regardless of whether this is a constrained stationary point. In fact, all iterates remain infeasible on this problem, so f could even be minimized at a strictly feasible point. Again, we assume that $\mu_F=\mu_I=1$, and that all mutation steps are generated in each iteration (so we are taking the best of all neighboring points).

Figure 3 illustrates the initial point and the three search directions in the search pattern used in this example. From a point (a,b) the solution set steps during the search are 120 degrees apart from one another, given by

$$\left(a+\Delta\cos\left(\frac{\pi}{4}\right),b+\Delta\sin\left(\frac{\pi}{4}\right)\right)=\left(a+\Delta\sqrt{2}/2,b+\Delta\sqrt{2}/2\right)$$
$$\left(a+\Delta\cos\left(\frac{-5\pi}{12}\right),b+\Delta\sin\left(\frac{-5\pi}{12}\right)\right)=\left(a+\Delta\sqrt{2}\,\omega_2,b-\Delta\sqrt{2}\,\omega_1\right)$$
$$\left(a+\Delta\cos\left(\frac{11\pi}{12}\right),b+\Delta\sin\left(\frac{11\pi}{12}\right)\right)=\left(a-\Delta\sqrt{2}\,\omega_1,b+\Delta\sqrt{2}\,\omega_2\right)$$

where $\omega_1=(\sqrt{3}+1)/4$ and $\omega_2=(\sqrt{3}-1)/4$. We label these points $\overline{a}$, $\overline{b}$ and $\overline{c}$ respectively, and we label the initial point $\overline{x}$.

We denote constraint (1) to be $b\geq\frac{1}{5}a+\frac{4}{5}$ and constraint (2) to be $b\leq 5a-4$. Let $D_1^{\overline{x}}$ be the shortest squared distance from $\overline{x}$ to constraint (1), and let $D_2^{\overline{x}}$ be the shortest squared distance from $\overline{x}$ to constraint (2). We define similar values for $\overline{a}$, $\overline{b}$, and $\overline{c}$. To compute these values, we need to be able to compute the shortest squared distance from a point to the constraints that point is violating. The following lemma defines the point on a line that is closest to a given point.

Lemma 1. *The shortest squared distance between a point* (r,s) *and a line* $y=mx+b$ *is at* $x=\frac{r+(s-b)m}{m^2+1}$.

Proof. Let $f(x)=(x-r)^2+(mx+b-s)^2$, which is the squared distance between (r,s) and the point on the line $(x,mx+b)$. To find the minimal distance, we minimize $f(x)$, which occurs when $f'(x)=0$. $f'(x)=2(x-r)+2m(mx+b-s)$, which is zero at $x=\frac{r+(s-b)m}{m^2+1}$.

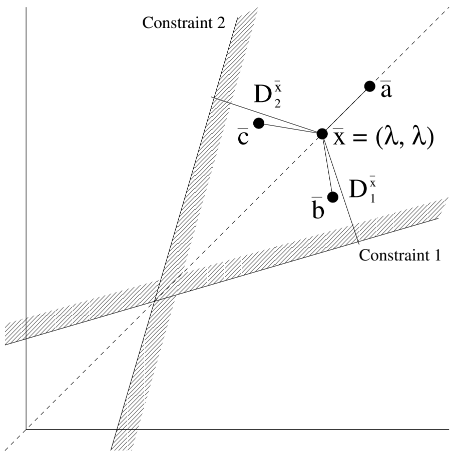

Fig. 3. An illustration of the initial point and associated search directions in Example II.

The following corollary follows directly from Lemma 1.

Corollary 1 *The squared distance from* (r, s) *to constraint (1) is* $\frac{(4+r-5s)^2}{26}$*, and the distance to constraint (2) is* $\frac{(4-5r+s)^2}{26}$.

The following theorem demonstrates that Algorithm A converges to the feasible point $(1, 1)$, though all iterates remain infeasible.

Theorem 2. *If* $z_0 = (\lambda, \lambda)$ *for some* $\lambda > 1$*, then no feasible point is generated and* $\lim_{t\to\infty} z_t = (1, 1)$.

Figure 4 illustrates the four different *algorithmic states* that can occur when Algorithm A searches from a point (λ, λ). The four lemmas in Appendix A demonstrate how the search progresses as follows:

(a) The point z_t is at some point (λ', λ') and the points $\bar{b}$ and $\bar{c}$ are infeasible with respect to both constraints. Lemma 2 shows that either (1) the step length is reduced or (2) either $z_{t+1} = \bar{b}$ or $z_{t+1} = \bar{c}$.
(b) The point z_t is at some point (λ', λ') and the points $\bar{b}$ and $\bar{c}$ are feasible with respect to constraint (1) and constraint (2) respectively. Lemma 3 shows that the step length is reduced.
(c) The previous algorithmic state was state (a), and the point $\bar{b}$ is infeasible with respect to both constraints. Lemma 4 shows that $z_{t+1} = (\lambda', \lambda')$.
(d) The previous algorithmic state was state (a), and the point $\bar{b}$ is feasible with respect to constraint (1). Lemma 5 shows that $z_{t+1} = (\lambda', \lambda')$.

Note that steps (c) and (d) assume that the point $\bar{b}$ is selected from a point (λ, λ). There are equivalent algorithmic states for the case where $\bar{c}$ is selected. We now prove Theorem 2 using this decomposition of the search of Algorithm A.

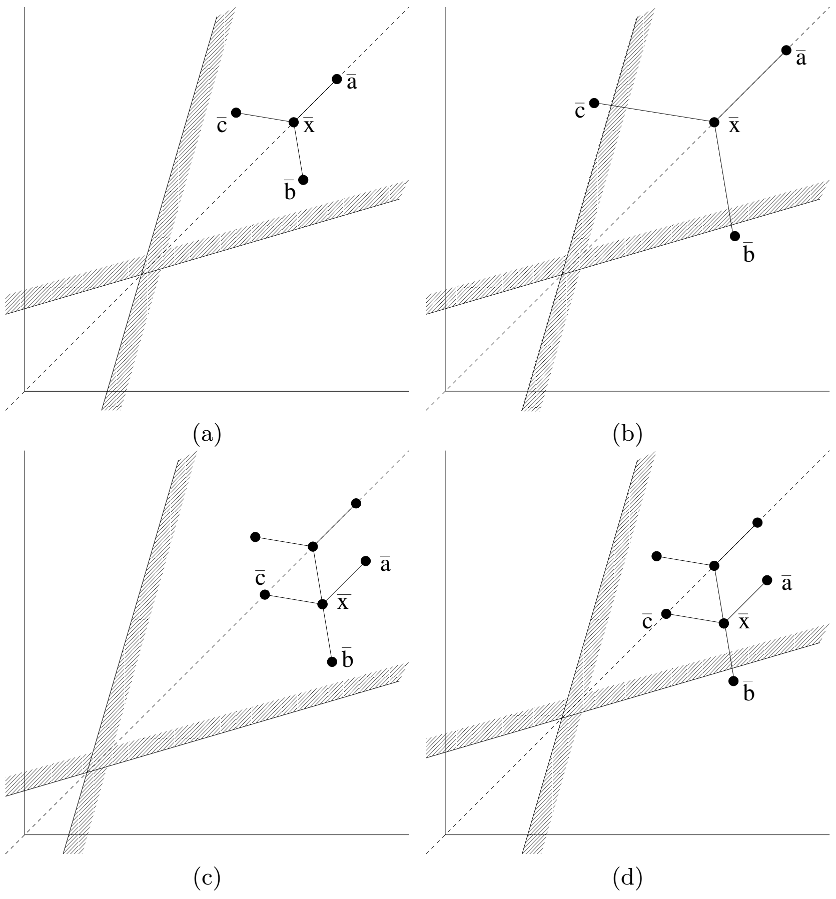

Fig. 4. The four different algorithmic states that can occur in Example II.

Proof (Proof of Theorem 2). Beginning at some point $z_0 = (\lambda, \lambda)$, $\lambda > 1$, it is clear from the results of Lemmas 2, 3, 4 and 5 that Algorithm A generates a sequence of points such that if $a_t \neq b_t$ then $a_{t+1} = b_{t+1}$. Thus there is an infinite subsequence K s.t. $a_k = b_k$, $\forall k \in K$. Let $\{(\lambda_k, \lambda_k)\}_{k \in K}$ denote this subsequence. We know that $\lambda_k > 1$ for all $k \in K$, so the sequence $\{\lambda_k\}_{k \in K}$ is monotonic and bounded below. Thus there exists a limit point of $\{(\lambda_k, \lambda_k)\}_{k \in K}$. Suppose that $\exists\ \lambda^* > 1$ s.t. (λ^*, λ^*) is this limit point. The sequence $(\lambda_k, \lambda_k)_{k \in K}$ is a refining subsequence because $\lim_{k \in K} \Delta_k = 0$, with associated positive spanning set defined by the fixed pattern used in this example. Now h is continuously differentiable, so h is strictly differentiable at $z^* = (\lambda^*, \lambda^*)$. Thus we know from Theorem 1 that $\nabla h(z^*) = 0$. But this is a contradiction, because ∇h is only zero at the point $(1, 1)$. Thus we conclude that $z^* = (1, 1)$.

This analysis demonstrates a less obvious limitation of Theorem 1 than is illustrated in Example I. Specifically, this example shows that although the convergence theory may ensure convergence to a feasible point, the local properties of that point may be poorly characterized. Although the specific choice of search pattern was crucial to our analysis, we conjecture that this is a more fundamental weakness of multi-objective EA search strategies for constrained optimization. By treating the objective function as one of two or more objectives, the search may proceed to find feasible solutions without regard to whether these solutions are interesting. Multi-objective EA strategies must also search in the neighborhood of the limit point to ensure that it is a constrained stationary point.

4 Discussion

Clevenger, Ferguson and Hart [4] provide a complete description of the analysis of F-EPSAs. The analysis of F-EPSAs demonstrates that they generate interesting limit points with probability one on a general class of constrained optimization problems. To our knowledge, this is the only convergence theory for EAs on constrained problems that does not require the use of derivative information. Consequently, these results suggest that similar multi-objective methods will be effective in practice.

However, this result depends on the set of search directions D that are defined, since this ultimately limits the cone that contains $-\nabla f(\hat{x})$. The examples in Section 3 illustrate how the performance of Algorithm A is sensitive to the choice of D. We expect that in practice Algorithm A will perform a robust search for constrained local minima if a sufficiently rich set of search directions are employed. For example, the two examples in Section 3 employ a single pattern throughout their search, but it is easy to imagine how search patterns could be dynamically adapted using directions from a large, finite set D.

Acknowledgements. This work was performed at Sandia National Laboratories. Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

1. C. Audet and J. E. Dennis Jr. A pattern search filter method for nonlinear programming without derivatives. Department of Computational and Applied Mathematics, Rice University, Houston, Texas, *TR00-09*, 2000.
2. E. Camponogara and S. N. Talukdar. A genetic algorithm for constrained and multiobjective optimization. In Jarmo T. Alander, editor, *3rd Nordic Workshop on Genetic Algorithms and Their Applications,* 49-62, 1997.
3. F. H. Clarke. Optimization and nonsmooth analysis. *SIAM Classics in Applied Mathematics*, 5, 1990.

4. L. Clevenger, L. Ferguson, and W. E. Hart. A filter-based evolutionary algorithm for constrained optimization. (submitted).
5. C. A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering Systems, Gordon and Breach Science Publishers*, 17:319–346, 2000.
6. C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.
7. K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.
8. L. Ferguson and W. E. Hart. A filter-based evolutionary algorithm for constrained optimization (extended abstract). *Proc Workshop on Foundations of Evolutionary Algorithms*, pages 287–290, 2003.
9. R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wächter. Global convergence of trust-region SQP-filter algorithms for nonlinear programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.
10. R. Fletcher and S. Leyffer. On the global convergence of a filter-SQP algorithm. *SIAM Journal on Optimization*, 13(1):44–59, 2002.
11. W. E. Hart. Rethinking the design of real-coded evolutionary algorithms: Making discrete choices in continuous search domains. *Soft Computing Journal*, 2003. (to appear).
12. J. D. Knowles, D. Corne, and W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
13. A. E. Smith and D. W. Coit. Constraint handling techniques–penalty functions. *Handbook of Evolutionary Computation,* Oxford University Press and Institute of Physics Publishing, 1997.

A Analysis of Example II

Proofs of the following Lemmas can be found in Clevenger, Ferguson, and Hart [4].

Lemma 2. *If* $\overline{x} = (\lambda, \lambda)$ *for some* $\lambda > 1$ *and* $\overline{b} = (\lambda + \Delta\sqrt{2}\omega_2, \lambda - \Delta\sqrt{2}\omega_1))$ *violates both constraints, then* $h(\overline{x}) < h(\overline{a})$ *and for sufficiently small* Δ*,* $h(\overline{x}) > h(\overline{c}) = h(\overline{b})$*.*

Lemma 3. *If* $\overline{x} = (\lambda, \lambda)$ *for some* $\lambda > 1$ *where* $D_2^{\overline{b}} = 0$*, then* $h(\overline{b}) > h(\overline{x})$*.*

Lemma 4. *If* $\overline{x} = (\lambda + \Delta\sqrt{2}\omega_2, \lambda - \Delta\sqrt{2}\omega_1))$ *for some* $\lambda > 1$ *and* $\overline{b} = (\lambda + 2\Delta\sqrt{2}\omega_2, \lambda - 2\Delta\sqrt{2}\omega_1))$ *violates both constraints, then* $h(\overline{x}) < h(\overline{a})$*,* $h(\overline{c}) < h(\overline{x})$ *and* $h(\overline{c}) < h(\overline{b})$*.*

Lemma 5. *If* $\overline{x} = (\lambda + \Delta\sqrt{2}\omega_2, \lambda - \Delta\sqrt{2}\omega_1)$ *for some* $\lambda > 1$ *where* $D_1^{\overline{b}} = 0$*, then* $h(\overline{x}) < h(\overline{b})$*.*