

**On the Convergence of Pattern
Search Algorithms**

Virginia Torczon

**CRPC-TR93322
June 1993**

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

ON THE CONVERGENCE OF PATTERN SEARCH ALGORITHMS*

VIRGINIA TORCZON†

Abstract. This paper gives a unifying, abstract generalization of pattern search methods for solving nonlinear optimization problems. Pattern search methods are a class of direct search methods—methods that neither require nor explicitly approximate derivatives. We use the abstract description of pattern search methods to establish a global first-order stationary point convergence theory that neither requires the directional derivative nor enforces a notion of sufficient decrease. We also discuss the relationship between the convergence analysis for pattern search methods and the analysis for both line search and model trust region globalization strategies; in particular, the fact that we can relax the requirements on the acceptance of the step, at the expense of stronger conditions on the form of the step, and still guarantee global convergence.

Key words. unconstrained optimization, convergence analysis, direct search methods, model trust region methods, line search methods, globalization strategies, alternating directions, alternating variable search, axial relaxation, local variation, coordinate search, response surface methodology, evolutionary operation, pattern search, multidirectional search, downhill simplex search

AMS(MOS) subject classifications. 49D30, 65K05

1. Introduction. In this paper, we study methods that require neither the directional derivative nor an approximation to the directional derivative to solve the unconstrained minimization problem

$$\min_{x \in \mathbf{R}^n} f(x),$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$. Such methods generally are referred to as *direct search* methods.

The purpose of this paper is to define and analyze a generalization of *pattern search* methods, a particular subclass of direct search methods. We give a global first-order stationary point convergence theory for pattern search methods, which, to our knowledge, provides the first known convergence result for some of these methods, and the first general convergence theory for all of them.

A unique feature of the convergence analysis we present is that we are able to guarantee first-order stationary point convergence without an explicit representation of the gradient or the directional derivative. In particular, we can prove convergence for pattern search methods even though they do not explicitly enforce fraction of Cauchy decrease, the Armijo–Goldstein–Wolfe conditions, or some other notion of sufficient decrease, on their iterates. However, the global convergence analysis of these methods also shows that they share several important features with both line search and model trust region methods. We believe this to be somewhat subtle and unexpected.

* This research was sponsored by the Air Force Office of Scientific Research grant 89-0363, the United States Air Force grant F49629-92-J-0203, and the Department of Energy grant DE-FG005-86ER25017. This research was also supported in part by the Geophysical Parallel Computation Project under State of Texas Contract #1059.

† Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892.

The important ideas for the convergence analysis for pattern search methods come from the convergence analysis developed by Torczon [20] for the *multidirectional search algorithm* of Dennis and Torczon [9, 19]. The main contribution of this paper is a concise abstraction of the key ingredients necessary for a more general convergence theory.

For some time we have been aware that the same style of argument used to prove global convergence for the multidirectional search algorithm could be applied, individually, to such classical algorithms as *coordinate search*, with fixed step sizes, variants of *response surface methodology*, first developed by Box and Wilson [4] and later popularized by Box [2, 3], and the original *pattern search algorithm* of Hooke and Jeeves [10]. The challenge was to develop an abstraction that both allowed for a general convergence theory and explained why such algorithms, often viewed as disparate direct search methods, could be analyzed using the same techniques. The goal was to show that these methods were special cases of a generalized pattern search method. The convergence analysis we present will make it clear why these methods are as robust as their proponents have long claimed, while clarifying some of the limitations that have long been ascribed to these methods. In addition, now that the key ingredients these methods share have been identified, it is possible to develop new pattern search methods for which the analysis holds.

The abstraction we present should make the important elements of the global convergence theory much clearer than those found in [20]. Furthermore, we hope that the unexpected parallels with classical convergence analysis for both line search and model trust region methods now will be more evident. This paper also includes a new analytic argument for the proposition, found in [20], stating that if the sequence of iterates is uniformly bounded away from the set of stationary points of the function, then the multidirectional search algorithm can visit only a finite number of points. (This result has also been extended to generalized pattern search methods.) In addition, we include a correction to the specification for the scaling factors found in [20].

In the next section we will establish the notation and general specification of pattern search methods. In §3, we will prove that if the function to be minimized is continuously differentiable, then pattern search methods guarantee weak first-order stationary point convergence. In §4 we will discuss the relationship between the convergence theory for pattern search methods and the convergence theory for line search and model trust region methods. In §5 we will show that the classical pattern search methods mentioned above, as well as the newer multidirectional search algorithm of Dennis and Torczon, conform to the general specification for pattern search methods. In §6, we give some concluding remarks; §7 contains technical results needed for the proofs of §3.

Notation. We will denote by \mathbf{R} , \mathbf{Q} , \mathbf{Z} , and \mathbf{N} the sets of real, rational, integer, and natural numbers, respectively.

All norms will be Euclidean vector norms or the associated operator norm. We also define $L(x_0) = \{x : f(x) \leq f(x_0)\}$ and

$$(1) \quad X_* = \{x : \nabla f(x) = 0\}.$$

2. The Generalized Pattern Search Method. We begin by introducing the following abstraction of pattern search methods. We defer to §5 demonstrations that the pattern search methods mentioned above fall comfortably within this abstraction.

2.1. The Pattern. To define a pattern we need two components, a basis matrix and a generating matrix.

The basis matrix can be any nonsingular matrix $B \in \mathbf{R}^{n \times n}$.

The generating matrix is a matrix $C_k \in \mathbf{Z}^{n \times p}$, where $p > 2n$. We partition the generating matrix into components

$$(2) \quad C_k = [M_k \quad -M_k \quad L_k] = [\Gamma_k \quad L_k].$$

We require that $M_k \in \mathbf{M} \subset \mathbf{Z}^{n \times n}$, where \mathbf{M} is a finite set of nonsingular matrices, and that $L_k \in \mathbf{Z}^{n \times (p-2n)}$ and contains at least one column, the column of zeros.

A *pattern* P_k is then defined by the columns of the matrix $P_k = BC_k$. Because both B and C_k have rank n , the columns of P_k span \mathbf{R}^n . For convenience, we use the partition of the generating matrix C_k given in (2) to partition P_k as follows:

$$(3) \quad P_k = BC_k = [BM_k \quad -BM_k \quad BL_k] = [B\Gamma_k \quad BL_k].$$

Given $\Delta_k \in \mathbf{R}$, $\Delta_k > 0$, we define a *trial step* s_k^i to be any vector of the form

$$(4) \quad s_k^i = \Delta_k Bc_k^i,$$

where c_k^i denotes a column of $C_k = [c_k^1 \cdots c_k^p]$. Note that Bc_k^i determines the direction of the step, while Δ_k serves as a step length parameter.

At iteration k , we define a *trial point* as any point of the form $x_k^i = x_k + s_k^i$, where x_k is the current iterate.

The Exploratory Moves. Pattern search methods proceed by conducting a series of *exploratory moves* about the current iterate before declaring a new iterate and updating the associated information. These moves can be viewed as sampling the function about the current iterate x_k in a well-defined deterministic fashion in search of a new iterate $x_{k+1} = x_k + s_k$ with a lower function value. The individual pattern search methods are distinguished, in part, by the manner in which these exploratory moves are conducted. To allow the broadest possible choice of exploratory moves, and yet still maintain the properties required to prove convergence for the pattern search methods, we place two requirements on the exploratory moves associated with any particular pattern search method. These requirements are given in the following Hypotheses on Exploratory Moves. (Please note an abuse of notation that is nonetheless convenient. Throughout this paper, if A is a matrix, then the notation $y \in A$ will mean the vector y is contained in the set of columns of A .)

The choice of exploratory moves must ensure two things:

1. The direction of any step s_k accepted at iteration k is defined by the pattern P_k and its length is determined by Δ_k . Note that s_k may be contained in $\Delta_k BL_k$ rather than in $\Delta_k B\Gamma_k$.

Hypotheses on Exploratory Moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k [B\Gamma_k BL_k]$.
 2. If $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.
2. If simple decrease on the function value at the current iterate can be found among *any* of the $2n$ trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves must produce a step s_k that also gives simple decrease on the function value at the current iterate. Note that $f(x_k + s_k)$ need not be less than or equal to $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\}$; s_k only need satisfy $f(x_k + s_k) < f(x_k)$ if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$.

These are the properties of the exploratory moves that enable us to prove weak first-order stationary point convergence by requiring only simple decrease on f , while avoiding the necessity of enforcing either a fraction of Cauchy decrease or the Armijo–Goldstein–Wolfe conditions on the iterates. To obtain a stronger result we need to place stronger hypotheses on the exploratory moves, as well as placing a boundedness condition on the columns of the generating matrices. These extensions will be discussed further in §3.3.2.

The Generalized Pattern Search Method. We now specify the generalized pattern search method for unconstrained minimization.

Algorithm 1. The Generalized Pattern Search Method.

Let $x_0 \in \mathbf{R}^n$ and $\Delta_0 > 0$ be given.

For $k = 0, 1, \dots$,

- a) Compute $f(x_k)$.
- b) Determine a step s_k using an *exploratory moves* algorithm.
- c) Compute $\rho_k = f(x_k) - f(x_k + s_k)$.
- d) If $\rho_k > 0$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
- e) Update C_k and Δ_k .

To define a pattern search method, it is necessary to specify the basis matrix B , the generating matrix C_k , the result s_k of the exploratory moves, and the algorithms for updating C_k and Δ_k .

The Updates. The aim of the updating algorithm for Δ_k is to force $\rho_k > 0$. An iteration with $\rho_k > 0$ is *successful*; otherwise, the iteration is *unsuccessful*. Again we note that to accept a step we only require *simple*, as opposed to *sufficient*, decrease.

Algorithm 2. Updating Δ_k .

Given $\tau \in \mathbf{Q}$, let $\theta = \tau^{w_0}$ and $\lambda_k \in \Lambda = \{\tau^{w_1}, \dots, \tau^{w_L}\}$, where $\tau > 1$ and $\{w_0, w_1, \dots, w_L\} \subset \mathbf{Z}$, $L \equiv |\Lambda| < +\infty$, $w_0 < 0$, and $w_i \geq 0$, $i = 1, \dots, L$.

- a) If $\rho_k \leq 0$ then $\Delta_{k+1} = \theta \Delta_k$.
- b) If $\rho_k > 0$ then $\Delta_{k+1} = \lambda_k \Delta_k$.

The conditions on θ and Λ ensure that $0 < \theta < 1$ and $\lambda_i \geq 1$ for all $\lambda_i \in \Lambda$. Thus, if an iteration is successful it may be possible to increase the step length parameter Δ_k ,

but Δ_k is not allowed to decrease. Not surprisingly, this is crucial to the success of the analysis. Also crucial to the analysis is the relationship (overlooked in [20]) between θ and the elements of Λ .

The algorithm for updating C_k will depend on the pattern search method. For theoretical purposes, it is sufficient to choose the columns of C_k so that they satisfy (2) and the conditions we have placed on the matrices $M_k \in \mathbf{M} \subset \mathbf{Z}^{n \times n}$ and $L_k \in \mathbf{Z}^{n \times (p-2n)}$.

3. The Convergence Theory. Having set up the machinery to define pattern search methods, we are now ready to analyze these methods. This analysis will produce theorems of several types. The first, developed in §3.1, demonstrates an algebraic fact about the nature of pattern search methods that requires *no* assumption on the function f . This theorem is critical to the proof of the convergence results for it shows that we only need require simple decrease in f to ensure global convergence. The second theorem, developed in §3.2, describes the limiting behavior of the step length control parameter Δ_k if we place only a very mild condition on the function f and exploit the interaction of the simple decrease condition for the generalized pattern search method with the algorithm for updating Δ_k . Finally, the third and fourth theorems, developed in §3.3, give the results for first-order stationary point convergence. The first theorem guarantees weak first-order stationary point convergence for any generalized pattern search method that satisfies the specifications given in §2. This is significant since the theorem applies to all the pattern search methods we will discuss in §5 *without* the need to impose any modifications on the methods as originally stated. The second theorem is equivalent to convergence results for line search and trust region globalization strategies. We can guarantee first-order stationary point convergence, but to do so requires placing stronger conditions on the specifications for generalized pattern search methods. These stronger conditions are immediately satisfied by only one of the pattern search methods we will present (that due to G. E. P. Box and Wilson [4]). We could certainly impose these stronger conditions on the remaining pattern search methods presented in §5—none of them are unreasonable to suggest or to enforce—but we would do so at the expense of attractive algorithmic features found in the original methods.

3.1. The Algebraic Structure of the Iterates. The results found in this section are *purely algebraic facts about the nature of pattern search methods*; they are also *independent of the function to be optimized*. It is the algebraic structure of the iterates that allows us to prove global convergence for pattern search methods without imposing a notion of either sufficient or Cauchy decrease on the iterates.

We begin by showing in what sense Δ_k is a step length parameter.

LEMMA 3.1. *There exists a constant $\zeta_* > 0$, independent of k , such that for any trial step $s_k^i \neq 0$ produced by a generalized pattern search method (Algorithm 1) we have*

$$\|s_k^i\| \geq \zeta_* \Delta_k.$$

Proof. From (4) we have $s_k^i = \Delta_k B c_k^i$. The conditions we have placed on the generating matrix C_k insure that $c_k^i \in \mathbf{Z}^n$.

Let $\sigma_n(B)$ denote the smallest singular value of B . Then

$$\|s_k^i\| = \Delta_k \|Bc_k^i\| \geq \Delta_k \sigma_n(B) \|c_k^i\| \geq \Delta_k \sigma_n(B).$$

The last inequality holds because at least one of the components of c_k^i is a nonzero integer, and hence $\|c_k^i\| \geq 1$. \square

From Lemma 3.1 we can see that the role of Δ_k as a step length parameter is to regulate backtracking and thus prevent excessively short steps.

THEOREM 3.2. *Any iterate x_N produced by a generalized pattern search method (Algorithm 1) can be expressed in the following form:*

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where

- x_0 is the initial iterate,
- $\beta/\alpha \equiv \tau$, with $\alpha, \beta \in \mathbf{N}$ and relatively prime, and τ is as defined in the algorithm for updating Δ_k (Algorithm 2),
- r_{LB} and r_{UB} depend on N ,
- Δ_0 is the initial choice for the step length control parameter,
- B is the basis matrix, and
- $z_k \in \mathbf{Z}^n$, $k = 0, \dots, N-1$.

Proof. The generalized pattern search algorithm, as stated in Algorithm 1, guarantees that any iterate x_N is of the form

$$(5) \quad x_N = x_0 + \sum_{k=0}^{N-1} s_k.$$

(We adopt the convention that $s_k = 0$ if iteration k is unsuccessful.) We also know that the step s_k must come from the set of trial steps s_k^i , $i = 1, \dots, p$. The trial steps are of the form $s_k^i = \Delta_k Bc_k^i$.

Consider the step length parameter Δ_k . For any $k \geq 0$, the update for Δ_k given in Algorithm 2 guarantees that Δ_k is of the form

$$(6) \quad \Delta_k = \theta^{p_k^0} \lambda_1^{p_k^1} \lambda_2^{p_k^2} \dots \lambda_L^{p_k^L} \Delta_0,$$

where $p_k^i \in \mathbf{Z}$ and $p_k^i \geq 0$. (Recall that $L = |\Lambda|$.) We have also placed the following restrictions on the form of θ and λ_i : for a given $\tau \in \mathbf{Q}$, $\tau > 1$, and $\{w_0, w_1, \dots, w_L\} \subset \mathbf{Z}$,

$$\theta = \tau^{w_0}, \quad w_0 < 0$$

and

$$\lambda_i = \tau^{w_i}, \quad w_i \geq 0, \quad i = 1, \dots, L.$$

We can thus rewrite (6) as:

$$(7) \quad \Delta_k = (\tau^{w_0})^{p_k^0} (\tau^{w_1})^{p_k^1} (\tau^{w_2})^{p_k^2} \dots (\tau^{w_L})^{p_k^L} \Delta_0 = \tau^{r_k} \Delta_0,$$

where $r_k \in \mathbf{Z}$. Let

$$(8) \quad r_{LB} = \min_{0 \leq k < N} \{r_k\} \quad r_{UB} = \max_{0 \leq k < N} \{r_k\}.$$

Then from (5) and (7) we have

$$x_N = x_0 + \sum_{k=0}^{N-1} \Delta_k B c_k = x_0 + \Delta_0 B \sum_{k=0}^{N-1} \tau^{r_k} c_k.$$

Since τ is rational, we can express τ as $\tau = \frac{\beta}{\alpha}$, where $\alpha, \beta \in \mathbf{N}$ are relatively prime. Then, using (8),

$$(9) \quad x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}} \right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where $z_k \in \mathbf{Z}^n$. \square

Theorem 3.2 synthesizes the requirements we have placed on the pattern, the definition of the trial steps, and the algorithm for updating Δ_k . Note that this means that for a fixed N , all the iterates lie on a translated integer lattice generated by x_0 and the columns of $\beta^{r_{LB}} \alpha^{-r_{UB}} \Delta_0 B$.

3.2. The Limiting Behavior of the Step Length Control Parameter. The next theorem combines the strict algebraic structure of the iterates with the simple decrease condition of the generalized pattern search algorithm, along with the algorithm for updating Δ_k , to give us a useful fact about the limiting behavior of Δ_k .

THEOREM 3.3. *Assume that $L(x_0)$ is compact. Then $\liminf_{k \rightarrow +\infty} \Delta_k = 0$.*

Proof. The proof is by contradiction. Suppose $0 < \Delta_{LB} \leq \Delta_k$ for all k . From (7) we know that Δ_k can be written as $\Delta_k = \tau^{r_k} \Delta_0$, where $r_k \in \mathbf{Z}$.

The hypothesis that $\Delta_{LB} \leq \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. Meanwhile, we also know that the sequence $\{\Delta_k\}$ is bounded above because all the iterates x_k must lie inside the set $L(x_0)$ and the latter set is compact; Lemma 3.1 then guarantees an upper bound $\Delta_{UB} < +\infty$ for $\{\Delta_k\}$. This, in turn, means that the sequence $\{\tau^{r_k}\}$ is bounded above. Consequently, the sequence $\{r_k\}$ is a finite set. Equivalently, the sequence $\{r_k\}$ is bounded above and below.

Let

$$(10) \quad r_{LB} = \min_{0 \leq k < +\infty} \{r_k\} \quad r_{UB} = \max_{0 \leq k < +\infty} \{r_k\}.$$

Then (9) now holds for the bounds given in (10), rather than (8), and we see that for all k , x_k lies in the translated integer lattice G generated by x_0 and the columns of $\beta^{r_{LB}} \alpha^{-r_{UB}} \Delta_0 B$.

The intersection of the compact set $L(x_0)$ with the translated integer lattice G is finite. Thus, there must exist at least one point x_* in the lattice for which $x_k = x_*$ for infinitely many k .

We now appeal to the simple decrease condition in the generalized pattern search method (Algorithm 1, d), which guarantees that a lattice point cannot be revisited

infinitely many times since we accept a new step s_k if and only if $f(x_k) > f(x_k + s_k)$. Thus there exists an N such that for all $k \geq N$, $x_k = x_*$, which implies that $\rho_k = 0$.

We now appeal to the algorithm for updating Δ_k (Algorithm 2, a) to see that $\Delta_k \rightarrow 0$, thus leading to a contradiction. \square

3.3. First-Order Stationary Point Convergence. Throughout the discussion in this section, we assume that f is continuously differentiable on $L(x_0)$; however, this assumption can be weakened, as we shall discuss further in §4.

3.3.1. The General Result. To prove Theorem 3.5 we need Proposition 3.4. We defer the proof of this proposition to §7 in part because we wish to discuss there several other issues that are tangential to the proof of Theorem 3.5. It is also the case that the proof of this result is similar in style to that given for the equivalent result found in [20], though now restated more succinctly in terms of the machinery we developed in §2.

PROPOSITION 3.4. *Assume that $L(x_0)$ is compact, that f is continuously differentiable on $L(x_0)$, and that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then there exists a constant $\Delta_{LB} > 0$ such that for all k , $\Delta_k > \Delta_{LB}$.*

We emphasize that the existence of a lower bound Δ_{LB} for Δ_k is guaranteed *only* under the null hypothesis that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$.

THEOREM 3.5. *Assume that $L(x_0)$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method (Algorithm 1),*

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Proof. The proof is by contradiction. Suppose that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then Proposition 3.4 tells us that there exists $\Delta_{LB} > 0$ such that $\Delta_k \geq \Delta_{LB}$. But this contradicts Theorem 3.3. \square

3.3.2. The Stronger Result. We can strengthen the result given in Theorem 3.5 at the expense of wider applicability. To begin with, we must add three further restrictions: one on the pattern matrix, one on the hypotheses on the exploratory moves, and one on the limiting behavior of the step length control parameter Δ_k .

First, we must ensure that the columns of the generating matrix C_k are bounded in norm, i.e., that there exists a constant $\mathcal{C} > 0$ such that for all k , $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \dots, p$. Given this bound, we can place an upper bound, in terms of Δ_k , on the norm of any trial step s_k^i .

LEMMA 3.6. *Given a constant $\mathcal{C} > 0$ such that for all k , $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \dots, p$, there exists a constant $\psi_* > 0$, independent of k , such that for any trial step s_k^i produced by a generalized pattern search method (Algorithm 1) we have*

$$\Delta_k \geq \psi_* \|s_k^i\|.$$

Proof. From (4) we have $s_k^i = \Delta_k B c_k^i$. Then $\|s_k^i\| = \Delta_k \|B c_k^i\| \leq \Delta_k \|B\| \|c_k^i\| \leq \Delta_k C \|\mathcal{B}\|$. Set $\psi_* = \frac{1}{C \|\mathcal{B}\|}$. \square

Note that the columns of $M_k \in \mathbf{M}$ are bounded by the assumption that $|\mathbf{M}| < +\infty$; we use this fact in the proof of Proposition 7.4. The stronger boundedness condition on the columns of $C_k = [M_k - M_k \ L_k]$ is needed to monitor the behavior of L_k .

Second, we must replace the original Hypotheses on Exploratory Moves with a stronger version, as given below. Together, Lemma 3.6 and the Strong Hypotheses on

Strong Hypotheses on Exploratory Moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k [B \Gamma_k \ B L_k]$.
2. If $\min\{f(x_k + y), y \in \Delta_k B \Gamma_k\} < f(x_k)$, then $f(x_k + s_k) \leq \min\{f(x_k + y), y \in \Delta_k B \Gamma_k\}$.

Exploratory Moves allow us to tie decrease in f to the norm of the gradient when the step sizes get small enough. This is the import of Corollary 7.5, which is given in §7.

Third, we require that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. We can use the algorithm for updating Δ_k (Algorithm 2) to ensure that this condition holds. For instance, we can force Δ_k to be nonincreasing by requiring $w_i = 0, i = 1, \dots, L$, which when taken together with Theorem 3.3 guarantees that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. All the algorithms we will consider in §5, except the multidirectional search algorithm, enforce this condition by limiting $\Lambda = \{1\} \equiv \{\tau^0\}$. However, it is not necessary to force the steps to be nonincreasing; we need only require that in the limit the step length control parameter goes to zero, which, in conjunction with Lemmas 3.1 and 3.6, has the effect of ultimately forcing the steps to zero.

THEOREM 3.7. *Assume that $L(x_0)$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. In addition, assume that the columns of the generating matrix are bounded in norm, that the generalized pattern search method (Algorithm 1) enforces the Strong Hypotheses on Exploratory Moves, and that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method,*

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Proof. The proof is by contradiction. Suppose $\limsup_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Let $\varepsilon > 0$ be such that there exists a subsequence $\|\nabla f(x_{m_i})\| \geq \varepsilon$. Since

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0,$$

given any $0 < \eta < \varepsilon$, there exists an associated subsequence l_i such that

$$\|\nabla f(x_k)\| > \eta \text{ for } m_i \leq k < l_i, \quad \|\nabla f(x_{l_i})\| < \eta.$$

Then, since $\Delta_k \rightarrow 0$, we can appeal to Corollary 7.5 to obtain for $m_i \leq k < l_i$, i sufficiently large,

$$f(x_k) - f(x_{k+1}) \geq \sigma \|\nabla f(x_k)\| \|s_k\| \geq \sigma \eta \|s_k\|,$$

where $\sigma > 0$. Then the telescoping sum:

$$(f(x_{m_i}) - f(x_{m_i+1})) + (f(x_{m_i+1}) - f(x_{m_i+2})) + \cdots + (f(x_{l_i-1}) - f(x_{l_i})) \geq \sum_{k=m_i}^{l_i} \sigma \eta \|s_k\|$$

gives us

$$f(x_{m_i}) - f(x_{l_i}) \geq \sum_{k=m_i}^{l_i} \sigma \eta \|s_k\| \geq c' \|x_{m_i} - x_{l_i}\|.$$

Since $f(x_{m_i}) - f(x_{l_i}) \rightarrow 0$ as $i \rightarrow +\infty$, $\|x_{m_i} - x_{l_i}\| \rightarrow 0$ as $i \rightarrow +\infty$. Then, because ∇f is uniformly continuous,

$$\|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| < \eta,$$

for i sufficiently large. However,

$$(11) \quad \|\nabla f(x_{m_i})\| \leq \|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| + \|\nabla f(x_{l_i})\| \leq 2\eta.$$

Since (11) must hold for any η , $0 < \eta < \varepsilon$, we have a contradiction (e.g., try $\eta = \frac{\varepsilon}{4}$). \square

The proof of Theorem 3.7 is almost identical to that of an equivalent result for trust region methods that was first given by Thomas [18] and which is included, in a more general form, in the survey by Moré [11].

One final note: the hypotheses of Theorem 3.7 suggest that in the absence of any explicit higher-order information about the function to be minimized, it makes sense to terminate a generalized pattern search algorithm when Δ_k is less than some reasonably small tolerance. In fact, this is a common stopping condition for algorithms of this sort and the one implemented for the multidirectional search algorithm [21].

4. Parallels with Line Search and Model Trust Region Theory. The global convergence analysis we have just presented for pattern search methods shares similarities with the global convergence analysis for both line search and model trust region methods.

Parallels with the line search theory are perhaps most obvious and are discussed in [20]. The outline for the convergence theory follows the outline for global convergence theorems as detailed by Ortega and Rheinboldt [14] and reviewed in the survey by Nocedal [13]: we consider iterations of the form $x_{k+1} = x_k + s_k$ where $s_k \in \Delta_k P_k$; the columns of P_k determine the search directions and Δ_k serves as a step length parameter. In §7.1 we show that pattern search methods are *descent methods* (as defined in either [8] or [14]) by showing that if x_k is not a stationary point of the function then the generalized pattern search method guarantees decrease in the value of the objective function in a finite number of iterations. We then provide, in §7.2, a measure of the goodness of the search direction by showing that pattern search methods are *gradient-related methods* (as defined in [14]). Finally, in Theorem 3.3 and Proposition 3.4, we consider the length of the step.

Given these major components for the convergence analysis, the one that is most unusual is that involving the step length control. For a standard line search iteration, the

strategy for choosing the step relies on satisfying the Armijo–Goldstein–Wolfe conditions for *sufficient* decrease: the step length Δ_k is chosen to satisfy

$$f(x_k + \Delta_k p_k) \leq f(x_k) + \sigma_1 \Delta_k \nabla f(x_k)^T p_k$$

$$\nabla f(x_k + \Delta_k p_k)^T p_k \geq \sigma_2 \nabla f(x_k)^T p_k,$$

where p_k is the search direction and the constants σ_1 and σ_2 are chosen to satisfy $0 < \sigma_1 < \sigma_2 < 1$. The first condition prevents very small decreases in the function values relative to the lengths of the steps taken. The second condition prevents steps that are too small relative to the initial rate of decrease of f . The standard convergence analysis then proceeds on the assumption that these conditions for sufficient decrease have been satisfied.

The obvious difficulty that arises in any attempt to extend the notion of sufficient decrease to an iteration of any direct search method is that enforcement of the Armijo–Goldstein–Wolfe conditions requires the directional derivative—information that the direct search methods do not otherwise require.¹

The point of both Theorem 3.2 and Theorem 3.3 is that the Armijo–Goldstein–Wolfe conditions need not be enforced to guarantee weak first-order stationary point convergence. The reason is that pattern search methods place strict limitations on the choice of both the search directions and the step lengths. The set of search directions is limited to the columns of P_k . The step length parameter Δ_k must be a multiple of Δ_0 , θ , and $\lambda \in \Lambda$. These restrictions guarantee an underlying lattice structure as stated in Theorem 3.2. Once again we stress that this lattice structure is a purely algebraic fact about the nature of pattern search methods.

What does this algebraic structure mean for the convergence analysis? The answer lies in the simple control mechanisms imposed by the Hypotheses on the Exploratory Moves and the update algorithm for Δ_k (Algorithm 2). The Hypotheses on the Exploratory Moves require that *if* simple decrease can be found for some one of the steps defined by Γ_k , then the exploratory moves must return a step that gives simple decrease. This step can be defined by *any* column of the generating matrix. If we are lucky and guess right, we may only have to consider a single step at any given iteration; however, in the worst case we may have to consider all $2n$ steps defined by Γ_k . The role of the algorithm for updating Δ_k is to ensure that Δ_k is reduced *only* when the exploratory moves algorithm fails to produce a step that gives simple decrease on the value of the objective function.

The combination of these two mechanisms:

- Hypotheses on the Exploratory Moves
- Algorithm 2 for updating Δ_k

introduces backtracking into the generalized pattern search methods, which prevents steps that are too short. This is the import of Lemma 3.1.

¹ The analysis in [22] enforces a notion of sufficient decrease by imposing on the methods an “error-controlling sequence” that does not exist in the original algorithms. No suggestions are provided on how to construct such a sequence in practice.

The backtracking feature of the generalized pattern search methods is also critical because it guarantees that descent will be realized if the current iterate is not a stationary point of the function. The matrix Γ_k guarantees that a descent direction exists, even if we are not sure which of the $2n$ directions it may be. Thus we are assured that the pattern search methods will produce descent once Δ_k becomes small enough. However, the second of the Hypotheses on Exploratory Moves also prevents Δ_k from becoming unnecessarily fine if decrease can be found for one of the trial steps defined by $\Delta_k \Gamma_k$ and thereby ensures that $\{x_k\}$ cannot bog down at points not included in X_* .

The net effect of the algebraic structure of the pattern search methods, when combined with the Hypothesis on the Exploratory Moves and the algorithm for updating Δ_k , is to ensure that the pathologies which might otherwise occur were the Armijo–Goldstein–Wolfe conditions to be ignored (see [8]) cannot happen. These simple, if non-standard, mechanisms prevent the well-known pathologies that can arise if the length of the steps is not monitored.

Parallels with the global convergence theory for model trust region methods are perhaps less obvious. In fact, there is some temptation to call the pattern search methods *sample trust region* methods. This arises from the observation that in the basic version of each of the algorithms covered in §5, all possible steps can be seen to lie on the boundary of a trust region in a weighted l_∞ norm. We weight the norm by using the inverse of the basis matrix B . (Recall that the basis matrix is required to be nonsingular.) Thus, for all the pattern search methods (ignoring the acceleration steps found in the Hooke and Jeeves algorithm and the multidirectional search algorithm, which lie outside the trust region radius) all steps satisfy $\|B^{-1}s_k^i\|_\infty = \Delta_k$. In essence, Δ_k gives the radius of the region within which we trust our sampling to be effective. If the sampling does not produce simple decrease, we reduce the radius of the trust region.

It is worth comparing the global first-order stationary point convergence results for trust region methods and generalized pattern search methods. Roughly speaking, the result due to Powell [15] says that

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

assuming that ∇f is continuous and the Hessian (or its approximation) remains uniformly bounded. Thomas [18] assumes, in addition, that ∇f is uniformly continuous to prove that

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

Neither result requires $L(x_0)$ to be compact; instead, f must be bounded below. We have shown comparable results for pattern search methods in Theorems 3.5 and 3.7 under the hypothesis that $L(x_0)$ is compact. The gap between Theorem 3.5 and Theorem 3.7 is due to the fact that while model trust regions link decrease in f to the norm of the gradient, the generalized pattern search methods do not explicitly enforce such a condition. Only when we place stronger conditions on the generating matrix, the

Hypotheses on the Exploratory Moves, and the limiting behavior of Δ_k , which together have the effect of *implicitly* linking decrease in f to the norm of the gradient, can we obtain the stronger convergence result. Nonetheless, it is perhaps surprising that we can prove any such convergence theorem for pattern search methods.

Having noted the unexpected similarities between pattern search methods and line search and trust region methods, we now note one critical difference: the step control mechanism for pattern search methods, however modest, is always active. Line search and model trust region strategies are extensions to Newton or quasi-Newton methods motivated by a need to ensure global convergence. These extensions are designed so that the step length control mechanism introduced by the line search or model trust region strategy “turns off” and the full Newton or quasi-Newton step is accepted once the local quadratic model of the function is deemed reliable. Thus, the fast local convergence properties of these algorithms are assured. However, at best the pattern search methods are relying on implicit information about the gradient; they do not construct local quadratic models of the function. The only natural notion of a full step is that given by Δ_k . As has long been recognized [16], this means that pattern search methods do not enjoy fast local convergence properties. When using these methods one trades speed of convergence for robustness and wider applicability.

One other point worth making is that pattern search methods are well-defined even when the function to be minimized is not differentiable. It is trivial to extend Theorem 3.5 to handle functions that are nondifferentiable; we need only modify the set X_* . This is reassuring in light of the fact that those who rely on these methods typically only assume that the function is continuous. The modifications required are discussed in further detail in [20]. The only qualification to be made is that the resulting theorem is not a general result for the nonsmooth case; rather, it is an extension of the result for the smooth case. Because the uniform continuity of the gradient over

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

is necessary for the proof of Theorem 3.5, the definition of X_* given in (1) must include not only the stationary points of the function, but also the points where the function is nondifferentiable and where the gradient exists but is not continuous.

5. The Particular Pattern Search Methods. In §2 we stated the conditions an algorithm must satisfy to be a pattern search method. We now illustrate these conditions by considering the following specific algorithms:

- coordinate search with fixed step lengths,
- searches based on two-level factorial designs (see [4] and [5]),
- the original pattern search method of Hooke and Jeeves [10], and
- the multidirectional search algorithm of Dennis and Torczon ([9] and [19]).

We will show that these algorithms satisfy the conditions that define pattern search methods and thus are special cases of the generalized pattern search method presented as Algorithm 1. Then we can appeal to Theorem 3.5 to claim weak first-order stationary point convergence for these methods.

There are undoubtedly other algorithms for which the abstraction and accompanying analysis holds—including various modifications to the algorithms presented—but we shall confine our investigation to these, the best known of the pattern search methods, to illustrate the power of our abstract approach to pattern search methods.

5.1. Coordinate Search with Fixed Step Lengths. The method of coordinate search is the simplest and most obvious of all the pattern search methods. Davidon describes it concisely in the opening of his belated preface to Argonne National Laboratory Research and Development Report 5990 [7]:

Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure....

This simple search method enjoys many names, among them *alternating directions*, *alternating variable search*, *axial relaxation*, and *local variation*. We shall refer to it as *coordinate search*.

Perhaps less obvious is that coordinate search is a pattern search method. To see this, we begin by considering all possible outcomes for a single iteration of coordinate search when $n = 2$, as shown in Fig. 1. We mark the current iterate x_k . The x_k^i 's denote trial points considered during the course of the iteration. The next iterate x_{k+1} is marked. Solid circles indicate successful intermediate steps taken during the course of the iteration while open circles indicate points at which the function was evaluated but that did not produce further decrease in the value of the objective function. Thus, in the first scenario shown a step from x_k to x_k^1 resulted in a decrease in the objective function, so the step from x_k^1 to x_{k+1} was tried and led to a further decrease in the objective function value. The iteration was then terminated with a new point x_{k+1} that satisfies the simple decrease condition $f(x_{k+1}) < f(x_k)$. In the worst case, the last scenario shown, $2n$ trial points were evaluated (x_k^1 , $x_k^{1'}$, x_k^2 , and $x_k^{2'}$) without producing decrease in the function value at the current iterate x_k . In this case, $x_{k+1} = x_k$ and the step size must be reduced for the next iteration.

We now show this algorithm is an instance of a generalized pattern search method.

5.1.1. The Matrices. Coordinate search is usually defined so that the basis matrix is the identity matrix; i.e., $B = I$. However, knowledge of the problem may lead to a different choice for the basis matrix. It may make sense to search using a different coordinate system. For instance, if the variables are known to differ by several orders of magnitude, this can be taken into account in the choice of the basis matrix (though, as we will see in §7.2, this may have a significant effect on the behavior of the method).

The generating matrix for coordinate search is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combina-

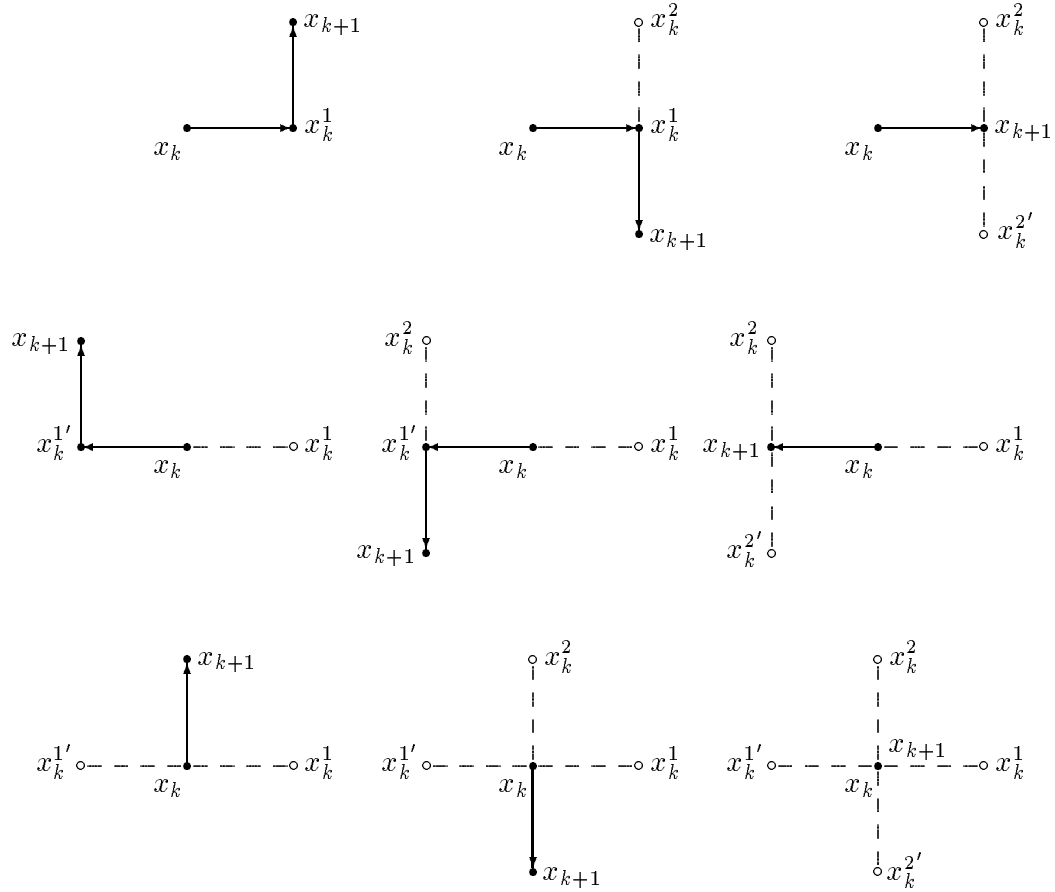


FIG. 1. All possible subsets of the steps for coordinate search in \mathbf{R}^2 .

tions of $\{-1, 0, 1\}$. Thus, C has $p = 3^n$ columns. In particular, the columns of C will contain both I and $-I$, as well as a column of zeros. We define $M = I$; L consists of the remaining $3^n - 2n$ columns of C . Since C is fixed across all iterations of the method, there is no need for an update algorithm.

For $n = 2$ we have

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

Thus, when $n = 2$, all possible trial points defined by the pattern $P = BC$, for a given step length Δ_k , can be seen in Fig. 2. Note that the pattern includes all the possible trial points enumerated in Fig. 1.

5.1.2. The Exploratory Moves. The exploratory moves for coordinate search are given in Algorithm 3, where the e_i 's denote the unit coordinate vectors.

The exploratory moves are executed sequentially in the sense that the selection of the next trial step is based on the success or failure of the previous trial step. Thus, while there are 3^n possible trial steps, we may compute as few as n trial steps, but we compute no more than $2n$ at any given iteration, as we saw in Fig. 1.

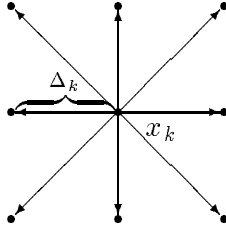


FIG. 2. The pattern for coordinate search in \mathbf{R}^2 with a given step length Δ_k .

Algorithm 3. Exploratory Moves Algorithm for Coordinate Search.

Given x_k , Δ_k , $f(x_k)$, and B , set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, n$ do

- a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Otherwise,

- i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

From the perspective of the theory, there are two conditions that need to be met by the exploratory moves algorithm. First, as Figs. 1 and 2 illustrate, all possible trial steps are contained in $\Delta_k P$.

The second condition on the exploratory moves is the more interesting; coordinate search demonstrates the laxity of this second hypothesis. For instance, in the first scenario shown in Fig. 1, decrease in the objective function was realized for the first trial step

$$s_k^1 = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

so the second trial step

$$s_k^2 = \Delta_k I \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

was tried, and accepted. It is certainly possible that greater decrease in the value of the objective function might have been realized for the trial step

$$s_k' = \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which is defined by a column in the matrix M (the step s_k^2 is defined by a column in the matrix L), but s_k' is not tried when simple decrease is realized by the step s_k^1 . However, *in the worst case*, as seen in Fig. 1, the algorithm for coordinate search ensures that all $2n$ steps defined by $\Delta_k B \Gamma = \Delta_k B [M - M] = \Delta_k B [I - I]$ are tried before returning the step $s_k = 0$. In other words, the exploratory moves given in Algorithm 3 will examine all $2n$ steps defined by $\Delta_k B \Gamma$ *unless* a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

5.1.3. Updating the Step Length. The update for Δ_k is exactly as given in Algorithm 2. As noted by Davidon, the usual practice is to continue with steps of the same magnitude until no further decrease in the objective function is realized, at which point the step size is halved. This corresponds to setting $\theta = 1/2$ and $\Lambda = \{1\}$. Thus, $\tau = 2$, $w_0 = -1$, and $w_1 = 0$.

This suffices to verify that coordinate search with fixed step length is a pattern search method. Theorem 3.5 thus holds. The exploratory moves algorithm for coordinate search would need to be modified to satisfy the Strong Hypotheses on Exploratory Moves for the conditions of Theorem 3.7 to be met, though this is a straightforward modification.

5.2. Response Surface Methodology. In 1951, Box and Wilson [4] introduced the notion of “response surface methodology” as a way to investigate an objective function by performing function evaluations at the vertices of some geometric configuration in the space of independent variables. This paper prepared the way for the development of direct search methods, in general, and what we now call pattern search methods, in particular.

In its earliest forms, response surface methodology was based on two-level factorial designs: evaluate the function at the vertices of a hypercube centered about the current iterate. (In fact, Box refers to this as one of a variety of “pattern of variants” [3].) If simple decrease in the value of the objective function is observed at one of the vertices, it becomes the new iterate. Otherwise, the lengths of the edges in the hypercube are halved and the process is repeated. Further discussions of the basic approach can be found in [5] and [17].

5.2.1. The Matrices. As with coordinate search, the usual choice for the basis matrix is $B = I$, though, as with coordinate search, other choices may be made to reflect information known about the problem to be solved.

The generating matrix for response surface methodology is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combinations of $\{-1, 1\}$; to this we append a column of zeros. Thus C has $p = 2^n + 1$ columns.

We take M to be any linearly independent subset of n columns of C ; $-M$ necessarily will be contained in C . Once again, L is fixed and consists of the remaining $(2^n + 1) - 2n$ columns of C .

Since the generating matrix is fixed, there is no need for an algorithm to update C .

5.2.2. The Exploratory Moves. The exploratory moves given in Algorithm 4 are simultaneous in the sense that every possible trial step $s_k^i \in \Delta_k P = \Delta_k BC$ is computed at each iteration. It is then the case that every trial step s_k^i is contained in $\Delta_k P$. The second observation of note is that since

$$s_k = \arg \min_{s_k^i \in \Delta_k P} \{f(x_k + s_k^i)\},$$

then, if $\min\{f(x_k + y), y \in \Delta_k B\Gamma\} < f(x_k)$, we have $f(x_k + s_k) < f(x_k)$, regardless of our choice of M (and thus, by extension, our choice of Γ). Furthermore, we are guaranteed that the Strong Hypotheses on Exploratory Moves are satisfied.

Algorithm 4. Exploratory Moves Algorithm for Response Surface Methodology.

Given x_k , Δ_k , $f(x_k)$, B , and $C = [c^1 \cdots c^p]$, set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, 2^n$ do

a) $s_k^i = \Delta_k B c^i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

5.2.3. Updating the Step Length. The algorithm for updating Δ_k is exactly as given in Algorithm 2, with θ usually set to $1/2$ and $\Lambda = \{1\}$.

Since we have shown that response surface methodology satisfies all the necessary requirements, we can therefore conclude that it, too, is a pattern search method, so Theorem 3.5 holds. The algorithm, as stated above, also satisfies the conditions of Theorem 3.7.

5.3. Hooke and Jeeves' Pattern Search Algorithm. In addition to introducing the general notion of a “direct search” method, Hooke and Jeeves introduced the pattern search method—a specific kind of search strategy—in their 1961 paper [10]. The pattern search of Hooke and Jeeves is essentially a variant of coordinate search that incorporates a *pattern step* in an attempt to accelerate the progress of the algorithm by exploiting information gained from the search during previous successful iterations.

The Hooke and Jeeves pattern search algorithm is opportunistic. If the previous iteration was successful (i.e., $\rho_{k-1} > 0$), then the current iteration begins by conducting coordinate search about a speculative iterate $x_k + (x_k - x_{k-1})$, rather than about the current iterate x_k . This is the pattern step. The idea is to investigate whether further progress is possible in the general direction $x_k - x_{k-1}$ (since, if $x_k \neq x_{k-1}$, then $x_k - x_{k-1}$ is clearly a promising direction).

To make this a little clearer, we consider the example shown in Fig. 3. Given

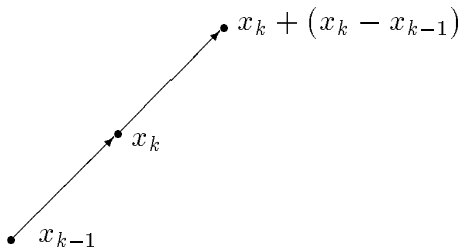


FIG. 3. The pattern step in \mathbf{R}^2 , given $x_k \neq x_{k-1}$, $k > 0$.

x_{k-1} and x_k (we assume, for now, that $k > 0$ and that $x_k \neq x_{k-1}$), the pattern search algorithm takes the step $x_k - x_{k-1}$ from x_k . The function is evaluated at this trial step and it is accepted, temporarily, *even if* $f(x_k + (x_k - x_{k-1})) \geq f(x_k)$. The Hooke and Jeeves pattern search algorithm then proceeds to conduct coordinate search about the

temporary iterate $x_k + (x_k - x_{k-1})$. Thus, in \mathbf{R}^2 , the exploratory moves are exactly as shown in Fig. 1, but with $x_k + (x_k - x_{k-1})$ substituted for x_k .

If coordinate search about the temporary iterate $x_k + (x_k - x_{k-1})$ is successful, then the point returned by coordinate search about the temporary iterate is accepted as the new iterate x_{k+1} . If not, i.e., $f((x_k + (x_k - x_{k-1})) + s_k) \geq f(x_k)$, then the pattern step is deemed unsuccessful, and the method reduces to coordinate search about x_k . For the two dimensional case, then, the exploratory moves would simply resort to the possibilities shown in Fig. 1.

If the previous iteration was not successful, so $x_k = x_{k-1}$ and $(x_k - x_{k-1}) = 0$, then the iteration is limited to coordinate search about x_k . In this instance, though, the updating algorithm for Δ_k will have reduced the size of the step (i.e., $\Delta_k = \theta\Delta_{k-1}$). The algorithm does not execute the pattern step when $k = 0$.

To express the pattern search algorithm within the framework we have developed, we use all the machinery required for coordinate search. Once again, the basis matrix is usually defined to be $B = I$. We append to the generating matrix another set of 3^n columns to capture the effect of the pattern step and we change the exploratory moves algorithm, as detailed below.

5.3.1. The Generating Matrix. Recall that the generating matrix for coordinate search consists of all possible combinations of $\{-1, 0, 1\}$ and is never changed. For the Hooke and Jeeves pattern search method, we allow the generating matrix to change from iteration to iteration to capture the effect of the pattern step. We append another set of 3^n columns, consisting of all possible combinations of $\{-1, 0, 1\}$, to the initial generating matrix for coordinate search. Thus C_k has $p = 2 \cdot 3^n$ columns. The additional 3^n columns allow us to express the effect of the pattern step with respect to x_k , rather than with respect to the temporary iterate $x_k + (x_k - x_{k-1})$, which is how the Hooke and Jeeves pattern search method usually is described. The matrix M is unchanged; $M = I$. Now, however, $L_k \in \mathbf{Z}^{n \times (p-2n)}$ is allowed to vary, though only in the 3^n columns associated with the pattern step. For $n = 2$,

$$(12) \quad C_0 = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

For notational convenience, we require that the last column of C_0 , which we denote as c_0^p , be the column of zeros. In both the algorithm for updating C_k (Algorithm 5) and the algorithm for the exploratory moves (Algorithm 6), we use the column c_k^p to measure the accumulation of a sequence of successful pattern steps. This can be seen, in (13), for our example from Fig. 3. In this example, we have the generating matrix

$$(13) \quad C_k = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 2 & 1 & 0 & 1 & 2 & 2 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 1 & 2 & 1 & 0 & 2 & 0 & 0 & 2 & 1 \end{bmatrix}.$$

The pattern step $(x_k - x_{k-1})$ is represented by the vector $(1 \ 1)^T$, seen in the last column of C_k . Note that the only difference between the columns of C_0 given in (12) and the columns of C_k given in (13) is that $(1 \ 1)^T$ has been added to the last 3^2 columns of C_k .

The algorithm for updating the generating matrix updates the last 3^n columns of C_k ; the first 3^n columns remain unchanged, as in coordinate search. The purpose of the updating algorithm is to incorporate the result of the search at the current iteration into the pattern for the next iteration. This is done using Algorithm 5. Note the distinguished role of c_k^p , the last column of C_k , which represents the pattern step $(x_k - x_{k-1})$.

Algorithm 5. Updating C_k .

For $i = 3^n + 1, \dots, 2 \cdot 3^n$ do

$$c_{k+1}^i = c_k^i + (1/\Delta_k)s_k - c_k^p.$$

Return.

Since $(1/\Delta_k)s_k$ is necessarily a column of C_k , and $C_0 \in \mathbf{Z}^{n \times p}$, an argument by induction shows that the update algorithm for C_k ensures that the columns of C_k always consist of integers.

5.3.2. The Exploratory Moves. In Algorithm 6, the e_i 's denote the unit coordinate vectors and c_k^p denotes the last column of C_k . We set $\rho_{-1} = 0$ so that ρ_{k-1} is defined when $k = 0$.

A useful example for working through the logic of the algorithm can be found in [1], though the presentation and notation differ somewhat from that given here.

Algorithm 6. Exploratory Moves Algorithm for Hooke and Jeeves.

Given x_k , Δ_k , $f(x_k)$, B , and ρ_{k-1} , set $\rho_k = \rho_{k-1}$ and $min = f(x_k)$.

If $\rho_k > 0$ then set $s_k = \Delta_k B c_k^p$, $\rho_k = f(x_k) - f(x_k + s_k)$, and $min = f(x_k + s_k)$.

For $i = 1, \dots, n$ do

a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Otherwise,

i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

If $\rho_k \leq 0$ then set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, n$ do

a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Otherwise,

i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

All possible steps are contained in $\Delta_k P_k$ since C_k contains columns that represent the ‘‘pattern steps’’ tried at the beginning of the iteration. And, once again, the exploratory moves given in Algorithm 6 will examine all $2n$ steps defined by $\Delta_k B \Gamma$ unless a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

Since we have shown that the pattern search algorithm of Hooke and Jeeves satisfies all the necessary requirements, we can therefore conclude that it, too, is a special case of the *generalized* pattern search method and Theorem 3.5 holds.

5.4. Multidirectional Search. The multidirectional search algorithm was introduced by Dennis and Torczon in 1989 [19] as a first step towards a general purpose optimization algorithm with promising properties for parallel computation. While subsequent work led to a class of algorithms, based on the multidirectional search algorithm, that allows for more flexible computation ([9] and [21]) one of the unanticipated results of the original research was a weak first-order stationary point convergence theorem for the multidirectional search algorithm [20].

The multidirectional search algorithm is a simplex-based algorithm. The pattern of points can be expressed as a simplex (i.e., $n + 1$ points, or vertices) based at the current iterate; as such, multidirectional search owes much in its conception to its predecessors, the simplex design algorithm of Spendley, Hext, and Himsworth [16] and the simplex algorithm of Nelder and Mead [12]. However, multidirectional search is a different algorithm—particularly from a theoretical standpoint. Convergence for the Spendley, Hext and Himsworth algorithm can be shown only with some modification of the original algorithm, and then only under the additional assumption that the function f is convex. There are numerical examples to demonstrate that the Nelder–Mead simplex algorithm may fail to converge to a stationary point of the function because the uniform linear independence property (discussed in §7.2), which plays a key role in the convergence analysis, cannot be guaranteed to hold [19].

The multidirectional search algorithm is described in detail in both [9] and [20]. The formulation given here is different and, in fact, introduces some redundancy that can be eliminated when actually implementing the algorithm. However, the way of expressing the algorithm that we will use here allows us to make clear the similarities between this and other pattern search methods.

5.4.1. The Matrices. It is most natural to express multidirectional search in terms of *multiple* basis matrices B_k and a *fixed* generating matrix C , which is at odds with our definition for generalized pattern search methods. As we shall see, however, it is possible to convert the more natural specification to one that conforms to our requirements for a pattern search method.

The multidirectional search algorithm centers around a family of basis matrices \mathbf{B} that consists of all matrices representing the edges adjacent to each vertex in a nondegenerate n -dimensional simplex that the user is allowed to specify. Since the ordering of the columns is not unique, and typically not preserved in the implementation of the method, we consider all possible representations of the columns of the matrices associated with the edges adjacent to the $(n + 1)$ vertices of the simplex. We then add the negatives of these $(n + 1)!$ basis matrices to account for the effect of the *reflection* step allowed by the multidirectional search algorithm. Thus the cardinality of the set \mathbf{B} is $|\mathbf{B}| = 2(n + 1)!$.

Fortunately, there is no need to actually construct and “store” this unwieldy number

of basis matrices to initialize the method. We can update the basis matrix after each iteration k by reconstructing the new basis matrix B_{k+1} , given the outcome of the exploratory moves, from the trial points x_k^i , $i = 1, \dots, n$, considered during the course of the exploratory moves. This procedure is given in Algorithm 8. The scalar *scale* is chosen during the course of the exploratory moves (see Algorithm 7) to ensure that $B_{k+1} \in \mathbf{B}$ by factoring out any change in the size of the simplex introduced by a change in Δ_k . This has the further effect of preserving the role of Δ_k as a step length parameter.

Algorithm 8. Updating B_k .

Given B_k , *scale*, *best*, and x_k^i for $i = 0, \dots, n$, denote $B_k = [b_k^i]$, $i = 1, \dots, n$.

If $\rho_k > 0$ then

For $i = 0, \dots, (\text{best} - 1)$ do

$$b_{k+1}^{i+1} = \text{scale} * (x_k^i - x_k^{\text{best}}).$$

For $i = (\text{best} + 1), \dots, n$ do

$$b_{k+1}^i = \text{scale} * (x_k^i - x_k^{\text{best}}).$$

Otherwise

For $i = 1, \dots, n$ do

$$b_{k+1}^i = b_k^i.$$

Return.

Given this use of a family of basis matrices to help define the multidirectional search algorithm, the generating matrix is then the fixed matrix $C = [I \ -I \ -\mu I \ 0]$. Thus, C contains $p = 3n + 1$ columns, with $M = I$. To ensure that $C \in \mathbf{Z}^{n \times p}$, we require $\mu \in \mathbf{Z}$. Furthermore, to ensure that the role of Δ_k as a step length parameter is not lost with the introduction of the *expansion* step represented by $-\mu I$, we require $\mu \in \Lambda$. The algorithm is defined so that $\Lambda = \{\tau^{w_1}, \tau^{w_2}\}$, with $\mu = \tau^{w_2}$. This requires the further restriction that $\tau \in \mathbf{N}$. Again, this is not an onerous restriction. Multidirectional search usually is specified so that $\tau = 2$, $w_2 = 1$, and thus $\mu = 2$.

Now, to bring this notation into conformity with our definition for a generalized pattern search method, observe that we can represent all possible basis matrices $B_\nu \in \mathbf{B}$ in terms of a single reference matrix $B \in \mathbf{B}$ so that

$$B_\nu = B \hat{B}_\nu, \quad \nu = 1, \dots, |\mathbf{B}|.$$

A convenient feature of using the edges of a simplex to form the set of basis matrices is that the matrices \hat{B}_ν consist only of elements from the set $\{-1, 0, 1\}$. The matrices \hat{B}_ν are necessarily nonsingular because of the nondegeneracy of the simplex. We use $\hat{\mathbf{B}}$ to represent the set of matrices \hat{B}_ν and observe that since \mathbf{B} is a finite set, the set $\hat{\mathbf{B}}$ is also finite.

We then observe that

$$P_k = B_k C = B_k [I \ -I \ -\mu I \ 0] \equiv B [\hat{B}_k \ -\hat{B}_k \ -\mu \hat{B}_k \ 0] = B C_k.$$

Thus we can define the pattern in terms of the single reference matrix B and the redefined generating matrix

$$C_k = [\hat{B}_k \ -\hat{B}_k \ -\mu \hat{B}_k \ 0],$$

with $M_k = \hat{B}_k$ and $\mathbf{M} \equiv \hat{\mathbf{B}}$. Since $\mu \in \mathbf{Z}$, $L_k \in \mathbf{Z}^{n \times (n+1)}$ and the column of zeros has been added explicitly.

5.4.2. The Exploratory Moves. The exploratory moves for the multidirectional search method are given in Algorithm 7; the e_i 's denote the unit coordinate vectors. We use the notion of $B_k \in \mathbf{B}$ for consistency with the update algorithm given in Algorithm 6, but we could just as easily substitute $B\hat{B}_k$ for B_k in the algorithm given below.

Algorithm 7. Exploratory Moves Algorithm for Multidirectional Search.

Given x_k , Δ_k , $f(x_k)$, B_k , and $\mu = \tau^{w_2} \in \mathbf{N}$, set $s_k = 0$, $\rho_k = 0$, $min = f(x_k)$, $\lambda_k = 1$, $scale = 1/\Delta_k$, $best = 0$, and $x_k^0 = x_k$.

For $i = 1, \dots, n$ do

a) $s_k^i = \Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, and $best = i$.

If $\rho_k \leq 0$ then

For $i = 1, \dots, n$ do

a) $s_k^i = -\Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, and $best = i$.

If $\rho_k > 0$ then set $scale = 1/\mu\Delta_k$.

For $i = 1, \dots, n$ do

a) $s_k^i = -\mu\Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, $best = i$, and $\lambda_k = \mu$.

Return.

Clearly, $s_k \in \Delta_k P_k$. Since the exploratory moves algorithm will consider all steps of the form $\Delta_k B\Gamma_k$, *unless* simple decrease is found after examining only the steps defined by $\Delta_k B M_k$, this guarantees we satisfy the condition that if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

5.4.3. Updating the Step Length. The algorithm for updating Δ_k is that given in Algorithm 2. In this case, while θ usually is set to $1/2$ so that $\tau = 2$, $w_0 = -1$, and $w_1 = 0$, we also include an expansion factor $\mu = \tau^{w_2}$, where w_2 usually equals one. Thus $\Lambda = \{1, \mu\}$, where μ is usually 2. The choice of $\lambda_k \in \Lambda$ is made during the execution of the exploratory moves.

Since we have shown that the multidirectional search algorithm satisfies all the necessary requirements, we conclude that it is also a pattern search method and thus Theorem 3.5 applies. Note that since we allow $\mu > 1$, which is a useful algorithmic feature, we cannot guarantee that $\lim_{k \rightarrow +\infty} \Delta_k = 0$ and so Theorem 3.7 does not automatically apply.

6. Conclusions. We have presented a framework in which one can analyze pattern search methods. This framework abstracts and quantifies the similarities of the classical pattern search methods and enables us to prove weak first-order stationary

point convergence for this class of algorithms. We also specify the conditions under which first-order stationary point convergence can be shown to hold.

These convergence results are perhaps surprising, given the simplicity of pattern search methods, but derive from the algebraic rigidity imposed on the iterates produced by pattern search methods. This is gratifying, since while this rigidity originally was introduced as a heuristic for directing the local search, it turns out to be the key to proving convergence as well. This analysis also highlights just how weak the conditions on the acceptance of the step can be and yet still allow a global convergence analysis, an observation that may prove useful in the analysis of other classes of optimization methods.

7. Appendix. We deferred the proof of Proposition 3.4 for several reasons. First, many of the results in this section are generalizations of similar results to be found in [20]. The abstraction in §2 leads to more succinct proofs. Second, the proof of Proposition 3.4 is closely related to that of several other results presented in this section and requires us to introduce several additional notions before tackling its proof.

We return to our definition of the pattern as $P_k = BC_k$ to show that the pattern contains at least one direction of descent whenever $\nabla f(x_k) \neq 0$.

Recall that we require the columns of C_k to contain both M_k and $-M_k$. Thus, P_k can be partitioned as follows:

$$P_k = BC_k = B [M_k \quad -M_k \quad L_k] = B [\Gamma_k \quad L_k].$$

We now elaborate on these requirements. Since M_k is an $n \times n$ nonsingular matrix, and B is nonsingular, we are guaranteed that BM_k forms a basis for \mathbf{R}^n . Thus at any iteration k , if $\nabla f(x_k) \neq 0$ we are guaranteed that Bc_k^i will be a direction of descent for at least one column c_k^i contained in the block Γ_k .

7.1. Descent Methods. Of course, the existence of a trial step in a descent direction is not sufficient to guarantee that decrease in the value of the objective function will be realized. To guarantee that a pattern search method is a descent method, we need to guarantee that in a finite number of iterations the method will produce a positive step size Δ_k that achieves decrease on the objective function at the current iterate. We now show that this is the case.

LEMMA 7.1. *Suppose that f is continuously differentiable on $L(x_0)$. If $\nabla f(x_k) \neq 0$, then there exists $p \in \mathbf{Z}$, $p \geq 0$, such that $\rho_{k+p} > 0$ (i.e., the $(k+p)^{\text{th}}$ iteration is successful).*

Proof. A key hypothesis placed on the exploratory moves is that if descent can be found for some one of the trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves will return a step that produces descent.

Because BC_k has rank n , if $\nabla f(x_k) \neq 0$, then there exists at least one trial direction $d_k^i = Bc_k^i$, where $c_k^i \in \Gamma_k$, such that $\nabla f(x_k)^T d_k^i \neq 0$. But, since $-c_k^i \in \Gamma_k$, without loss of generality, $\nabla f(x_k)^T d_k^i < 0$. Thus, there exists an $h_k > 0$ such that for $0 < h \leq h_k$, $f(x_k + hd_k^i) < f(x_k)$.

If at iteration k , $\Delta_k > h_k$, then the iteration may be unsuccessful; that is, $\rho_k = f(x_k) - f(x_k + s_k) \leq 0$. When the iteration is unsuccessful, the generalized pattern

search method sets $x_{k+1} = x_k$ and the updating algorithm sets $\Delta_{k+1} = \theta\Delta_k$. Since θ is strictly less than one, there exists $p \in \mathbf{Z}$, $p \geq 0$, such that $\theta^p\Delta_k \leq h_k$. Thus we are guaranteed descent, i.e., a successful iteration, in at most p iterations. \square

7.2. Uniform Linear Independence. The pattern P_k guarantees the existence of at least one direction of descent whenever $\nabla f(x_k) \neq 0$. We now want to guarantee the existence of a bound on the angle between the direction of descent contained in $B\Gamma_k$ and the negative gradient at x_k (whenever $\nabla f(x_k) \neq 0$). We will show, in fact, that this bound is uniform across all iterations of the pattern search algorithm. To do so, we use the notion of *uniform linear independence* [14].

LEMMA 7.2. *For a pattern search algorithm, there exists a constant $\xi > 0$ such that for all $k \geq 0$ and $x \neq 0$,*

$$(14) \quad \max \left\{ \frac{|x^T(x_k^i - x_k)|}{\|x\|\|x_k^i - x_k\|}, i = 1, \dots, p \right\} \geq \xi.$$

Proof. To demonstrate the existence of ξ , we first consider the simplest possible case, $B = I$ and $C = [M \ -M \ 0] = [I \ -I \ 0]$, and use this to derive a bound for any choice of B and C_k that satisfies the conditions we have imposed.

LEMMA 7.3. *Suppose $\|y\| = 1$. Let*

$$|\cos \theta(y)| = \max_{1 \leq j \leq n} \{ |y^T e_j| \},$$

where the e_j 's are the unit coordinate vectors.

If $B = I$ and $C = [I \ -I \ 0]$, then

$$\min_{y \in \mathbf{R}^n} |\cos \theta(y)| = \frac{1}{\sqrt{n}}.$$

Proof. We have $|y^T e_j| = |y_j|$, where $y = (y_1, \dots, y_n)^T$. Since $\sum_{j=1}^n |y_j|^2 = 1$, we are guaranteed that $|y_j| \geq 1/\sqrt{n}$ for some j , so $|y^T e_j| \geq 1/\sqrt{n}$ for some j . Thus $|\cos \theta(y)| \geq 1/\sqrt{n}$.

Now note that $\cos \theta(y)$ attains this lower bound for any $y = \alpha_1 e_1 + \alpha_2 e_2 + \dots + \alpha_n e_n$, where $\alpha_j = \pm 1/\sqrt{n}$. \square

Thus, if the pattern search is restricted to the coordinate directions defined by $P = [I \ -I \ 0]$, $\xi = 1/\sqrt{n}$ gives the lower bound on the absolute value of the cosine of the angle between the gradient and a guaranteed direction of descent. We now use the bound for this particular case to derive a bound for the general case.

Assume a general basis matrix B and a general matrix $M_k \in \mathbf{M}$, where $|\mathbf{M}| < +\infty$. We adopt the notation $BM_k = [y_k^1 \ \dots \ y_k^n]$. Then for any $x \neq 0$ we have the following:

$$|\cos \theta| = \frac{|x^T y_k^j|}{\|x\|\|y_k^j\|} = \frac{|x^T BM_k e_j|}{\|x\|\|BM_k e_j\|} = \frac{|((BM_k)^T x)^T e_j|}{\|x\|\|BM_k e_j\|}.$$

If we set $w = (BM_k)^T x$, so that $x = (BM_k)^{-T} w$, we have

$$\begin{aligned} |\cos \theta| &= \frac{|w^T e_j|}{\|(BM_k)^{-T} w\| \|BM_k e_j\|} \geq \frac{|w^T e_j|}{\|(BM_k)^{-T}\| \|w\| \|BM_k\| \|e_j\|} \\ &= \frac{1}{\|(BM_k)^{-T}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) = \frac{1}{\|(BM_k)^{-1}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) \\ &\geq \frac{1}{\kappa(BM_k)} \frac{1}{\sqrt{n}}, \end{aligned}$$

where $\kappa(BM_k)$ is the condition number of the matrix BM_k . Thus, we have

$$|\cos \theta| \geq \frac{1}{\kappa(BM_k) \sqrt{n}} > 0.$$

To ensure a bound ξ that is independent of the choice of any particular matrix $M \in \mathbf{M}$, we simply observe that the set \mathbf{M} is required to be finite. Thus, ξ is taken to be

$$(15) \quad \xi = \min_{M \in \mathbf{M}} \left\{ \frac{1}{\kappa(BM) \sqrt{n}} \right\}.$$

□

The bound given in (15) points to two features that explain much about the behavior of pattern search methods. Since we never explicitly calculate—or approximate—the gradient, we are dependent on the fact that in the worst case at least one of our search directions is not orthogonal to the gradient; ξ gives us a bound on how far away we can be. Thus, as either the condition number of the product BM_k increases, or the dimension of the problem increases, our bound on the angle between the search direction and the gradient deteriorates. This suggests two things. First, we should be very careful in our choice of B and \mathbf{M} for any particular pattern search method. Second, we should not be surprised that these methods become less effective as the dimension of the problem increases.

Nevertheless, even though pattern search methods neither require nor explicitly approximate the gradient of the function, the uniform linear independence condition demonstrates that the pattern search methods are, in fact, *gradient-related methods*, as defined by Ortega and Rheinboldt [14], which is one reason why we can establish global first-order stationary point convergence.

7.3. The Descent Condition. Having introduced the notion of uniform linear independence with the bound ξ , we are now ready to show that pattern search methods reduce Δ_k only when necessary to find descent. This enables us to prove Proposition 3.4.

PROPOSITION 7.4. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on $L(x_0)$. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exists $\delta > 0$, independent of k , such that if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta$, then the k th iteration of a generalized pattern search method (Algorithm 1) will be successful (i.e., $\rho_k = f(x_k) - f(x_k + s_k) > 0$) and thus $\Delta_{k+1} \geq \Delta_k$.

Proof. We restrict our attention to the steps defined by the columns of $\Delta_k B\Gamma_k$. This is sufficient since the Hypotheses on Exploratory Moves ensure that a step s_k satisfying the simple decrease condition $\rho_k > 0$ must be returned if a trial step defined by a column of $\Delta_k B\Gamma_k$ satisfies the simple decrease condition.

We first need some measure of the relative lengths of the steps defined by M_k and its negative $-M_k$. We begin by defining

$$e^k = \min_{i=1, \dots, 2n} \|x_k^i - x_k\| \equiv \min_{i=1, \dots, 2n} \|s_k^i\| \equiv \min_{j=1, \dots, n} \Delta_k \|Bc_k^j\|$$

and

$$E^k = \max_{i=1, \dots, 2n} \|x_k^i - x_k\| \equiv \max_{i=1, \dots, 2n} \|s_k^i\| \equiv \max_{j=1, \dots, n} \Delta_k \|Bc_k^j\|,$$

where c_k^i is restricted to be a column of $M_k \in \mathbf{M}$. (We assume that P_k is partitioned as in (3) so that the first $2n$ columns of P_k contain the columns of $[B\Gamma_k] = [BM_k - BM_k]$.) Since $|\mathbf{M}| < +\infty$, we can define η as

$$\eta = \min_{M \in \mathbf{M}} \frac{\min_{j=1, \dots, n} \|Bc^j\|}{\max_{j=1, \dots, n} \|Bc^j\|},$$

where c^j is a column of $M \in \mathbf{M}$. Observe that $0 < \eta \leq 1$. Then, for any $1 \leq i, j \leq 2n$,

$$\|x_k^i - x_k\| = \|s_k^i\| \leq E^k \leq \frac{1}{\eta} e^k \leq \frac{1}{\eta} \|s_k^j\| = \frac{1}{\eta} \|x_k^j - x_k\|.$$

We define the *contour* $C(x_0)$ to be $C(x_0) = \{x : f(x) = f(x_0)\}$. Since $x_0 \in \Omega_\epsilon$, Lemma 7.1 allows us to define $N = \min\{k : x_k \neq x_0\}$. We then define d to be $d = \text{dist}(L(x_N), C(x_0))$. Because $L(x_N)$ and $C(x_0)$ are compact and disjoint, we know that $d > 0$. We now make the following claim:

CLAIM. *Suppose $k \geq N$. If for some $j = 1, \dots, 2n$, $\|x_k^j - x_k\| \leq \eta d$, then $x_k^i \in L(x_0)$ for all $i = 1, \dots, 2n$.*

The proof of this claim is by straightforward application of the triangle inequality [20].

Let

$$\alpha = \min_{x \in \Omega_\epsilon} \|\nabla f(x)\|.$$

By design, $\alpha > 0$. Since ∇f is continuous on $L(x_0)$, ∇f is uniformly continuous on $L(x_0)$. Thus, there exists a constant $r > 0$, depending only on α and the ξ from (14), such that

$$\|\nabla f(x) - \nabla f(x_k)\| \leq \frac{\xi\alpha}{2} \text{ whenever } \|x - x_k\| \leq r \text{ (and } x \in L(x_0)\text{)}.$$

We define

$$(16) \quad \delta' = \min\{\eta d, r\}.$$

Now, assume $k \geq N$ and $x_k \in \Omega_\epsilon$. Choose a trial point x_k^i , $i = 1, \dots, 2n$, that satisfies both $\nabla f(x_k)^T(x_k^i - x_k) < 0$ and

$$\frac{|\nabla f(x_k)^T(x_k^i - x_k)|}{\|\nabla f(x_k)\| \|x_k^i - x_k\|} \geq \xi.$$

The definitions of Ω_ϵ and the pattern P_k , together with Lemma 7.2, guarantee the existence of at least one such x_k^i .

The Mean Value Theorem tells us that $f(x_k^i) - f(x_k) = \nabla f(\omega)^T(x_k^i - x_k)$ for some $\omega \in (x_k, x_k^i)$, whence

$$(17) \quad f(x_k^i) - f(x_k) = \nabla f(x_k)^T(x_k^i - x_k) + (\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k).$$

Consider the first term on the right-hand side of (17). Our choice of x_k^i gives us

$$|\nabla f(x_k)^T(x_k^i - x_k)| \geq \xi \|\nabla f(x_k)\| \|x_k^i - x_k\|.$$

Furthermore, since $\nabla f(x_k)^T(x_k^i - x_k) < 0$, we have

$$(18) \quad \nabla f(x_k)^T(x_k^i - x_k) \leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\|.$$

Now consider the second term on the right-hand side of (17). The Cauchy–Schwarz inequality gives us

$$(19) \quad |(\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k)| \leq \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\|.$$

Combine (18) and (19) to rewrite (17) as

$$\begin{aligned} f(x_k^i) - f(x_k) &\leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\| + \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\| \\ &= (-\xi \|\nabla f(x_k)\| + \|\nabla f(\omega) - \nabla f(x_k)\|) \|x_k^i - x_k\|. \end{aligned}$$

Since $\omega \in (x_k, x_k^i)$, once $\|x_k^i - x_k\| \leq \delta'$,

$$(20) \quad f(x_k^i) - f(x_k) \leq (-\xi \|\nabla f(x_k)\| + \frac{\xi}{2} \|\nabla f(x_k)\|) \|x_k^i - x_k\| < 0.$$

Thus, when $\|x_k^i - x_k\| \leq \delta'$, $f(x_k^i) < f(x_k)$. The Hypotheses on Exploratory Moves guarantee that if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$. Thus, $\rho_k = f(x_k) - f(x_k + s_k) > 0$ and the algorithms for updating Δ_k (Algorithm 2) ensures that $\Delta_{k+1} \geq \Delta_k$.

We invoke Lemma 3.1 to obtain

$$\delta' \geq \|x_k^i - x_k\| \equiv \|s_k^i\| \geq \zeta_* \Delta_k.$$

Thus, at any iteration $k \geq N$, if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta'/\zeta_*$, then $\Delta_{k+1} \geq \Delta_k$.

We close by noting that since iteration $k = N - 1$ is defined to be the first successful iteration, if we define

$$\delta = \min \left\{ \Delta_{N-1}, \frac{\delta'}{\zeta_*} \right\},$$

we are guaranteed that at any iteration k , if $x_k \in \Omega_\epsilon$ and $\delta \geq \Delta_k$, then $\Delta_{k+1} \geq \Delta_k$. \square

Proposition 7.4 guarantees that if Δ_k is small enough, a generalized pattern search method will realize simple decrease because there exists at least one step among the $2n$ steps defined by $\Delta_k B\Gamma_k$ that gives decrease as a function of the norm of the gradient at the current iterate, as shown in (20); the Hypotheses on Exploratory Moves then ensure that the exploratory moves algorithm must return a step that satisfies at least simple decrease. However, there are no guarantees that the step returned by an exploratory moves algorithm satisfies more than the simple decrease condition.

To tie the amount of actual decrease to the norm of the gradient, we must place much stronger conditions on the generalized pattern search method, as discussed in §3.3.2. Once we have done so, Corollary 7.5 follows more or less immediately from Proposition 7.4.

COROLLARY 7.5. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on $L(x_0)$. Suppose that the columns of the generating matrix are bounded in norm and that the generalized pattern search method (Algorithm 1) enforces the Strong Hypotheses on Exploratory Moves. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exist $\delta' > 0$ and $\sigma > 0$, independent of k , such that for all but finitely many k , if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta'$, then

$$f(x_{k+1}) \leq f(x_k) - \sigma \|\nabla f(x_k)\| \|s_k\| < f(x_k).$$

Proof. From Proposition 7.4, (20) says that for $k \geq N = \min\{k : x_k \neq x_0\}$ (Lemma 7.1 guarantees that $N < +\infty$), there exists at least one trial step $s_k^i \in \Delta_k B\Gamma_k$ such that once $\|s_k^i\| < \delta'$, where δ' is as defined in (16), we have

$$f(x_k^i) \leq f(x_k) - \frac{\xi}{2} \|\nabla f(x_k)\| \|s_k^i\| < f(x_k).$$

The Strong Hypotheses on the Exploratory Moves give us

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \|\nabla f(x_k)\| \|s_k^i\| < f(x_k).$$

Lemma 3.1 ensures that

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \zeta_* \Delta_k \|\nabla f(x_k)\| < f(x_k).$$

Lemma 3.6, which holds only when the columns of the generating matrix are bounded in norm, gives us

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \zeta_* \psi_* \|\nabla f(x_k)\| \|s_k\| < f(x_k).$$

We define $\sigma = \frac{\xi}{2} \zeta_* \psi_*$ to complete the proof. \square

We now prove Proposition 3.4.

Proof. (Proposition 3.4.) By assumption, $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then we can find N_1 and $\epsilon > 0$ such that for all $k \geq N_1$, $x_k \in \Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}$. Lemma 7.1 guarantees the existence of $N_2 = \min\{k : x_k \neq x_0\}$. Let $N = \max(N_1, N_2)$.

From Proposition 7.4 we are assured of $\delta > 0$ such that if $\Delta_k \leq \delta$, then the iteration will be successful. Given Δ_0 , there exists a constant $p \in \mathbf{Z}$, $p \geq 0$, such that $\theta^p \Delta_0 \leq \delta$. Thus, for $k \geq N$, $\theta^{p+1} \Delta_0 < \Delta_k$.

Set $\Delta_{LB} = \theta \min(\theta^p \Delta_0, \Delta_1, \dots, \Delta_{N-1})$. Then for all k , $\Delta_{LB} < \Delta_k$. \square

Acknowledgments. This paper benefited from conversations with J. E. Dennis, Michael Trosset, Luís Vicente, and especially Robert Michael Lewis. In particular, discussions with Michael Lewis were critical in the distillation of the abstraction for pattern search methods found in §2 and in the development of the analytic arguments for the algebraic structure of the iterates found in §3.1.

The review and comments made by two anonymous referees and Jorge Moré, the editor, are gratefully acknowledged. In particular, the observation by one of the referees that the pattern contains only vectors in a fixed lattice led to a more general result and a much more elegant presentation.

REFERENCES

- [1] M. AVRIEL, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [2] G. E. P. BOX, *The exploration and exploitation of response surfaces: Some general considerations and examples*, *Biometrics*, 10 (1954), pp. 16–60.
- [3] G. E. P. BOX, *Evolutionary operation: A method for increasing industrial productivity*, *Appl. Statist.*, 6 (1957), pp. 81–101.
- [4] G. E. P. BOX AND K. B. WILSON, *On the experimental attainment of optimum conditions*, *Journal of the Royal Statistical Society, Series B*, XIII (1951), pp. 1–45.
- [5] M. J. BOX, D. DAVIES, AND W. H. SWANN, *Non-Linear Optimization Techniques*, ICI Monograph No. 5, Oliver & Boyd, Edinburgh, 1969.
- [6] J. CÉA, *Optimisation : théorie et algorithmes*, Dunod, Paris, 1971.
- [7] W. C. DAVIDON, *Variable metric method for minimization*, *SIAM J. Optimization*, 1 (1991), pp. 1–17. The belated preface was received by the editors June 4, 1990; accepted for publication August 10, 1990. The rest of the article was originally published as Argonne National Laboratory Research and Development Report 5990, May 1959 (revised November 1959).
- [8] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [9] J. E. DENNIS, JR. AND V. TORCZON, *Direct search methods on parallel machines*, *SIAM J. Optimization*, 1 (1991), pp. 448–474.
- [10] R. HOOKE AND T. A. JEEVES, “Direct search” solution of numerical and statistical problems, *J. Assoc. Comput. Mach.*, 8 (1961), pp. 212–229.
- [11] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, in *Mathematical Programming, The State of the Art*, A. Bachem, M. Grottschel, and G. Korte, eds., Springer-Verlag, 1983, pp. 256–287.
- [12] J. A. NELDER AND R. MEAD, *A simplex method for function minimization*, *Comput. J.*, 7 (1965), pp. 308–313.
- [13] J. NOCEDAL, *Theory of algorithms for unconstrained optimization*, *Acta Numerica*, 1 (1992), pp. 199–242.
- [14] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several*

- Variables*, Academic Press, New York, 1970.
- [15] M. J. D. POWELL, *Convergence properties of a class of minimization algorithms*, in *Nonlinear Programming 2*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, 1975, pp. 1–27.
 - [16] W. SPENDLEY, G. R. HEXT, AND F. R. HIMSWORTH, *Sequential application of simplex designs in optimisation and evolutionary operation*, *Technometrics*, 4 (1962), pp. 441–461.
 - [17] W. H. SWANN, *Direct search methods*, in *Numerical Methods for Unconstrained Optimizations*, W. Murray, ed., Academic Press, London and New York, 1972, pp. 13–28.
 - [18] S. W. THOMAS, *Sequential estimation techniques for quasi-Newton algorithms*, Ph.D. thesis, Cornell University, Ithaca, NY, 1975.
 - [19] V. TORCZON, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1989; also available as Tech. Report 90–7, Department of Mathematical Sciences, Rice University, Houston, TX 77251–1892.
 - [20] ———, *On the convergence of the multidirectional search algorithm*, *SIAM J. Optimization*, 1 (1991), pp. 123–145.
 - [21] ———, *PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines*, Tech. Report 92–9, Department of Mathematical Sciences, Rice University, Houston, TX 77251–1892, 1992.
 - [22] Y. WEN-CI, *Positive basis and a class of direct search techniques*, *Scientia Sinica, Special Issue of Mathematics*, 1 (1979), pp. 53–67.

ON THE CONVERGENCE OF PATTERN SEARCH ALGORITHMS*

VIRGINIA TORCZON†

Abstract. This paper gives a unifying, abstract generalization of pattern search methods for solving nonlinear optimization problems. Pattern search methods are a class of direct search methods—methods that neither require nor explicitly approximate derivatives. We use the abstract description of pattern search methods to establish a global first-order stationary point convergence theory that neither requires the directional derivative nor enforces a notion of sufficient decrease. We also discuss the relationship between the convergence analysis for pattern search methods and the analysis for both line search and model trust region globalization strategies; in particular, the fact that we can relax the requirements on the acceptance of the step, at the expense of stronger conditions on the form of the step, and still guarantee global convergence.

Key words. unconstrained optimization, convergence analysis, direct search methods, model trust region methods, line search methods, globalization strategies, alternating directions, alternating variable search, axial relaxation, local variation, coordinate search, response surface methodology, evolutionary operation, pattern search, multidirectional search, downhill simplex search

AMS(MOS) subject classifications. 49D30, 65K05

1. Introduction. In this paper, we study methods that require neither the directional derivative nor an approximation to the directional derivative to solve the unconstrained minimization problem

$$\min_{x \in \mathbf{R}^n} f(x),$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$. Such methods generally are referred to as *direct search* methods.

The purpose of this paper is to define and analyze a generalization of *pattern search* methods, a particular subclass of direct search methods. We give a global first-order stationary point convergence theory for pattern search methods, which, to our knowledge, provides the first known convergence result for some of these methods, and the first general convergence theory for all of them.

A unique feature of the convergence analysis we present is that we are able to guarantee first-order stationary point convergence without an explicit representation of the gradient or the directional derivative. In particular, we can prove convergence for pattern search methods even though they do not explicitly enforce fraction of Cauchy decrease, the Armijo–Goldstein–Wolfe conditions, or some other notion of sufficient decrease, on their iterates. However, the global convergence analysis of these methods also shows that they share several important features with both line search and model trust region methods. We believe this to be somewhat subtle and unexpected.

* This research was sponsored by the Air Force Office of Scientific Research grant 89-0363, the United States Air Force grant F49629-92-J-0203, and the Department of Energy grant DE-FG005-86ER25017. This research was also supported in part by the Geophysical Parallel Computation Project under State of Texas Contract #1059.

† Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892.

The important ideas for the convergence analysis for pattern search methods come from the convergence analysis developed by Torczon [20] for the *multidirectional search algorithm* of Dennis and Torczon [9, 19]. The main contribution of this paper is a concise abstraction of the key ingredients necessary for a more general convergence theory.

For some time we have been aware that the same style of argument used to prove global convergence for the multidirectional search algorithm could be applied, individually, to such classical algorithms as *coordinate search*, with fixed step sizes, variants of *response surface methodology*, first developed by Box and Wilson [4] and later popularized by Box [2, 3], and the original *pattern search algorithm* of Hooke and Jeeves [10]. The challenge was to develop an abstraction that both allowed for a general convergence theory and explained why such algorithms, often viewed as disparate direct search methods, could be analyzed using the same techniques. The goal was to show that these methods were special cases of a generalized pattern search method. The convergence analysis we present will make it clear why these methods are as robust as their proponents have long claimed, while clarifying some of the limitations that have long been ascribed to these methods. In addition, now that the key ingredients these methods share have been identified, it is possible to develop new pattern search methods for which the analysis holds.

The abstraction we present should make the important elements of the global convergence theory much clearer than those found in [20]. Furthermore, we hope that the unexpected parallels with classical convergence analysis for both line search and model trust region methods now will be more evident. This paper also includes a new analytic argument for the proposition, found in [20], stating that if the sequence of iterates is uniformly bounded away from the set of stationary points of the function, then the multidirectional search algorithm can visit only a finite number of points. (This result has also been extended to generalized pattern search methods.) In addition, we include a correction to the specification for the scaling factors found in [20].

In the next section we will establish the notation and general specification of pattern search methods. In §3, we will prove that if the function to be minimized is continuously differentiable, then pattern search methods guarantee weak first-order stationary point convergence. In §4 we will discuss the relationship between the convergence theory for pattern search methods and the convergence theory for line search and model trust region methods. In §5 we will show that the classical pattern search methods mentioned above, as well as the newer multidirectional search algorithm of Dennis and Torczon, conform to the general specification for pattern search methods. In §6, we give some concluding remarks; §7 contains technical results needed for the proofs of §3.

Notation. We will denote by \mathbf{R} , \mathbf{Q} , \mathbf{Z} , and \mathbf{N} the sets of real, rational, integer, and natural numbers, respectively.

All norms will be Euclidean vector norms or the associated operator norm. We also define $L(x_0) = \{x : f(x) \leq f(x_0)\}$ and

$$(1) \quad X_* = \{x : \nabla f(x) = 0\}.$$

2. The Generalized Pattern Search Method. We begin by introducing the following abstraction of pattern search methods. We defer to §5 demonstrations that the pattern search methods mentioned above fall comfortably within this abstraction.

2.1. The Pattern. To define a pattern we need two components, a basis matrix and a generating matrix.

The basis matrix can be any nonsingular matrix $B \in \mathbf{R}^{n \times n}$.

The generating matrix is a matrix $C_k \in \mathbf{Z}^{n \times p}$, where $p > 2n$. We partition the generating matrix into components

$$(2) \quad C_k = [M_k \quad -M_k \quad L_k] = [\Gamma_k \quad L_k].$$

We require that $M_k \in \mathbf{M} \subset \mathbf{Z}^{n \times n}$, where \mathbf{M} is a finite set of nonsingular matrices, and that $L_k \in \mathbf{Z}^{n \times (p-2n)}$ and contains at least one column, the column of zeros.

A *pattern* P_k is then defined by the columns of the matrix $P_k = BC_k$. Because both B and C_k have rank n , the columns of P_k span \mathbf{R}^n . For convenience, we use the partition of the generating matrix C_k given in (2) to partition P_k as follows:

$$(3) \quad P_k = BC_k = [BM_k \quad -BM_k \quad BL_k] = [B\Gamma_k \quad BL_k].$$

Given $\Delta_k \in \mathbf{R}$, $\Delta_k > 0$, we define a *trial step* s_k^i to be any vector of the form

$$(4) \quad s_k^i = \Delta_k Bc_k^i,$$

where c_k^i denotes a column of $C_k = [c_k^1 \cdots c_k^p]$. Note that Bc_k^i determines the direction of the step, while Δ_k serves as a step length parameter.

At iteration k , we define a *trial point* as any point of the form $x_k^i = x_k + s_k^i$, where x_k is the current iterate.

The Exploratory Moves. Pattern search methods proceed by conducting a series of *exploratory moves* about the current iterate before declaring a new iterate and updating the associated information. These moves can be viewed as sampling the function about the current iterate x_k in a well-defined deterministic fashion in search of a new iterate $x_{k+1} = x_k + s_k$ with a lower function value. The individual pattern search methods are distinguished, in part, by the manner in which these exploratory moves are conducted. To allow the broadest possible choice of exploratory moves, and yet still maintain the properties required to prove convergence for the pattern search methods, we place two requirements on the exploratory moves associated with any particular pattern search method. These requirements are given in the following Hypotheses on Exploratory Moves. (Please note an abuse of notation that is nonetheless convenient. Throughout this paper, if A is a matrix, then the notation $y \in A$ will mean the vector y is contained in the set of columns of A .)

The choice of exploratory moves must ensure two things:

1. The direction of any step s_k accepted at iteration k is defined by the pattern P_k and its length is determined by Δ_k . Note that s_k may be contained in $\Delta_k BL_k$ rather than in $\Delta_k B\Gamma_k$.

Hypotheses on Exploratory Moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k [B\Gamma_k BL_k]$.
 2. If $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.
2. If simple decrease on the function value at the current iterate can be found among *any* of the $2n$ trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves must produce a step s_k that also gives simple decrease on the function value at the current iterate. Note that $f(x_k + s_k)$ need not be less than or equal to $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\}$; s_k only need satisfy $f(x_k + s_k) < f(x_k)$ if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$.

These are the properties of the exploratory moves that enable us to prove weak first-order stationary point convergence by requiring only simple decrease on f , while avoiding the necessity of enforcing either a fraction of Cauchy decrease or the Armijo–Goldstein–Wolfe conditions on the iterates. To obtain a stronger result we need to place stronger hypotheses on the exploratory moves, as well as placing a boundedness condition on the columns of the generating matrices. These extensions will be discussed further in §3.3.2.

The Generalized Pattern Search Method. We now specify the generalized pattern search method for unconstrained minimization.

Algorithm 1. The Generalized Pattern Search Method.

Let $x_0 \in \mathbf{R}^n$ and $\Delta_0 > 0$ be given.

For $k = 0, 1, \dots$,

- a) Compute $f(x_k)$.
- b) Determine a step s_k using an *exploratory moves* algorithm.
- c) Compute $\rho_k = f(x_k) - f(x_k + s_k)$.
- d) If $\rho_k > 0$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
- e) Update C_k and Δ_k .

To define a pattern search method, it is necessary to specify the basis matrix B , the generating matrix C_k , the result s_k of the exploratory moves, and the algorithms for updating C_k and Δ_k .

The Updates. The aim of the updating algorithm for Δ_k is to force $\rho_k > 0$. An iteration with $\rho_k > 0$ is *successful*; otherwise, the iteration is *unsuccessful*. Again we note that to accept a step we only require *simple*, as opposed to *sufficient*, decrease.

Algorithm 2. Updating Δ_k .

Given $\tau \in \mathbf{Q}$, let $\theta = \tau^{w_0}$ and $\lambda_k \in \Lambda = \{\tau^{w_1}, \dots, \tau^{w_L}\}$, where $\tau > 1$ and $\{w_0, w_1, \dots, w_L\} \subset \mathbf{Z}$, $L \equiv |\Lambda| < +\infty$, $w_0 < 0$, and $w_i \geq 0$, $i = 1, \dots, L$.

- a) If $\rho_k \leq 0$ then $\Delta_{k+1} = \theta \Delta_k$.
- b) If $\rho_k > 0$ then $\Delta_{k+1} = \lambda_k \Delta_k$.

The conditions on θ and Λ ensure that $0 < \theta < 1$ and $\lambda_i \geq 1$ for all $\lambda_i \in \Lambda$. Thus, if an iteration is successful it may be possible to increase the step length parameter Δ_k ,

but Δ_k is not allowed to decrease. Not surprisingly, this is crucial to the success of the analysis. Also crucial to the analysis is the relationship (overlooked in [20]) between θ and the elements of Λ .

The algorithm for updating C_k will depend on the pattern search method. For theoretical purposes, it is sufficient to choose the columns of C_k so that they satisfy (2) and the conditions we have placed on the matrices $M_k \in \mathbf{M} \subset \mathbf{Z}^{n \times n}$ and $L_k \in \mathbf{Z}^{n \times (p-2n)}$.

3. The Convergence Theory. Having set up the machinery to define pattern search methods, we are now ready to analyze these methods. This analysis will produce theorems of several types. The first, developed in §3.1, demonstrates an algebraic fact about the nature of pattern search methods that requires *no* assumption on the function f . This theorem is critical to the proof of the convergence results for it shows that we only need require simple decrease in f to ensure global convergence. The second theorem, developed in §3.2, describes the limiting behavior of the step length control parameter Δ_k if we place only a very mild condition on the function f and exploit the interaction of the simple decrease condition for the generalized pattern search method with the algorithm for updating Δ_k . Finally, the third and fourth theorems, developed in §3.3, give the results for first-order stationary point convergence. The first theorem guarantees weak first-order stationary point convergence for any generalized pattern search method that satisfies the specifications given in §2. This is significant since the theorem applies to all the pattern search methods we will discuss in §5 *without* the need to impose any modifications on the methods as originally stated. The second theorem is equivalent to convergence results for line search and trust region globalization strategies. We can guarantee first-order stationary point convergence, but to do so requires placing stronger conditions on the specifications for generalized pattern search methods. These stronger conditions are immediately satisfied by only one of the pattern search methods we will present (that due to G. E. P. Box and Wilson [4]). We could certainly impose these stronger conditions on the remaining pattern search methods presented in §5—none of them are unreasonable to suggest or to enforce—but we would do so at the expense of attractive algorithmic features found in the original methods.

3.1. The Algebraic Structure of the Iterates. The results found in this section are *purely algebraic facts about the nature of pattern search methods*; they are also *independent of the function to be optimized*. It is the algebraic structure of the iterates that allows us to prove global convergence for pattern search methods without imposing a notion of either sufficient or Cauchy decrease on the iterates.

We begin by showing in what sense Δ_k is a step length parameter.

LEMMA 3.1. *There exists a constant $\zeta_* > 0$, independent of k , such that for any trial step $s_k^i \neq 0$ produced by a generalized pattern search method (Algorithm 1) we have*

$$\|s_k^i\| \geq \zeta_* \Delta_k.$$

Proof. From (4) we have $s_k^i = \Delta_k B c_k^i$. The conditions we have placed on the generating matrix C_k insure that $c_k^i \in \mathbf{Z}^n$.

Let $\sigma_n(B)$ denote the smallest singular value of B . Then

$$\|s_k^i\| = \Delta_k \|Bc_k^i\| \geq \Delta_k \sigma_n(B) \|c_k^i\| \geq \Delta_k \sigma_n(B).$$

The last inequality holds because at least one of the components of c_k^i is a nonzero integer, and hence $\|c_k^i\| \geq 1$. \square

From Lemma 3.1 we can see that the role of Δ_k as a step length parameter is to regulate backtracking and thus prevent excessively short steps.

THEOREM 3.2. *Any iterate x_N produced by a generalized pattern search method (Algorithm 1) can be expressed in the following form:*

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where

- x_0 is the initial iterate,
- $\beta/\alpha \equiv \tau$, with $\alpha, \beta \in \mathbf{N}$ and relatively prime, and τ is as defined in the algorithm for updating Δ_k (Algorithm 2),
- r_{LB} and r_{UB} depend on N ,
- Δ_0 is the initial choice for the step length control parameter,
- B is the basis matrix, and
- $z_k \in \mathbf{Z}^n$, $k = 0, \dots, N-1$.

Proof. The generalized pattern search algorithm, as stated in Algorithm 1, guarantees that any iterate x_N is of the form

$$(5) \quad x_N = x_0 + \sum_{k=0}^{N-1} s_k.$$

(We adopt the convention that $s_k = 0$ if iteration k is unsuccessful.) We also know that the step s_k must come from the set of trial steps s_k^i , $i = 1, \dots, p$. The trial steps are of the form $s_k^i = \Delta_k Bc_k^i$.

Consider the step length parameter Δ_k . For any $k \geq 0$, the update for Δ_k given in Algorithm 2 guarantees that Δ_k is of the form

$$(6) \quad \Delta_k = \theta^{p_k^0} \lambda_1^{p_k^1} \lambda_2^{p_k^2} \dots \lambda_L^{p_k^L} \Delta_0,$$

where $p_k^i \in \mathbf{Z}$ and $p_k^i \geq 0$. (Recall that $L = |\Lambda|$.) We have also placed the following restrictions on the form of θ and λ_i : for a given $\tau \in \mathbf{Q}$, $\tau > 1$, and $\{w_0, w_1, \dots, w_L\} \subset \mathbf{Z}$,

$$\theta = \tau^{w_0}, \quad w_0 < 0$$

and

$$\lambda_i = \tau^{w_i}, \quad w_i \geq 0, \quad i = 1, \dots, L.$$

We can thus rewrite (6) as:

$$(7) \quad \Delta_k = (\tau^{w_0})^{p_k^0} (\tau^{w_1})^{p_k^1} (\tau^{w_2})^{p_k^2} \dots (\tau^{w_L})^{p_k^L} \Delta_0 = \tau^{r_k} \Delta_0,$$

where $r_k \in \mathbf{Z}$. Let

$$(8) \quad r_{LB} = \min_{0 \leq k < N} \{r_k\} \quad r_{UB} = \max_{0 \leq k < N} \{r_k\}.$$

Then from (5) and (7) we have

$$x_N = x_0 + \sum_{k=0}^{N-1} \Delta_k B c_k = x_0 + \Delta_0 B \sum_{k=0}^{N-1} \tau^{r_k} c_k.$$

Since τ is rational, we can express τ as $\tau = \frac{\beta}{\alpha}$, where $\alpha, \beta \in \mathbf{N}$ are relatively prime. Then, using (8),

$$(9) \quad x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}} \right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where $z_k \in \mathbf{Z}^n$. \square

Theorem 3.2 synthesizes the requirements we have placed on the pattern, the definition of the trial steps, and the algorithm for updating Δ_k . Note that this means that for a fixed N , all the iterates lie on a translated integer lattice generated by x_0 and the columns of $\beta^{r_{LB}} \alpha^{-r_{UB}} \Delta_0 B$.

3.2. The Limiting Behavior of the Step Length Control Parameter. The next theorem combines the strict algebraic structure of the iterates with the simple decrease condition of the generalized pattern search algorithm, along with the algorithm for updating Δ_k , to give us a useful fact about the limiting behavior of Δ_k .

THEOREM 3.3. *Assume that $L(x_0)$ is compact. Then $\liminf_{k \rightarrow +\infty} \Delta_k = 0$.*

Proof. The proof is by contradiction. Suppose $0 < \Delta_{LB} \leq \Delta_k$ for all k . From (7) we know that Δ_k can be written as $\Delta_k = \tau^{r_k} \Delta_0$, where $r_k \in \mathbf{Z}$.

The hypothesis that $\Delta_{LB} \leq \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. Meanwhile, we also know that the sequence $\{\Delta_k\}$ is bounded above because all the iterates x_k must lie inside the set $L(x_0)$ and the latter set is compact; Lemma 3.1 then guarantees an upper bound $\Delta_{UB} < +\infty$ for $\{\Delta_k\}$. This, in turn, means that the sequence $\{\tau^{r_k}\}$ is bounded above. Consequently, the sequence $\{\tau^{r_k}\}$ is a finite set. Equivalently, the sequence $\{r_k\}$ is bounded above and below.

Let

$$(10) \quad r_{LB} = \min_{0 \leq k < +\infty} \{r_k\} \quad r_{UB} = \max_{0 \leq k < +\infty} \{r_k\}.$$

Then (9) now holds for the bounds given in (10), rather than (8), and we see that for all k , x_k lies in the translated integer lattice G generated by x_0 and the columns of $\beta^{r_{LB}} \alpha^{-r_{UB}} \Delta_0 B$.

The intersection of the compact set $L(x_0)$ with the translated integer lattice G is finite. Thus, there must exist at least one point x_* in the lattice for which $x_k = x_*$ for infinitely many k .

We now appeal to the simple decrease condition in the generalized pattern search method (Algorithm 1, d), which guarantees that a lattice point cannot be revisited

infinitely many times since we accept a new step s_k if and only if $f(x_k) > f(x_k + s_k)$. Thus there exists an N such that for all $k \geq N$, $x_k = x_*$, which implies that $\rho_k = 0$.

We now appeal to the algorithm for updating Δ_k (Algorithm 2, a) to see that $\Delta_k \rightarrow 0$, thus leading to a contradiction. \square

3.3. First-Order Stationary Point Convergence. Throughout the discussion in this section, we assume that f is continuously differentiable on $L(x_0)$; however, this assumption can be weakened, as we shall discuss further in §4.

3.3.1. The General Result. To prove Theorem 3.5 we need Proposition 3.4. We defer the proof of this proposition to §7 in part because we wish to discuss there several other issues that are tangential to the proof of Theorem 3.5. It is also the case that the proof of this result is similar in style to that given for the equivalent result found in [20], though now restated more succinctly in terms of the machinery we developed in §2.

PROPOSITION 3.4. *Assume that $L(x_0)$ is compact, that f is continuously differentiable on $L(x_0)$, and that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then there exists a constant $\Delta_{LB} > 0$ such that for all k , $\Delta_k > \Delta_{LB}$.*

We emphasize that the existence of a lower bound Δ_{LB} for Δ_k is guaranteed *only* under the null hypothesis that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$.

THEOREM 3.5. *Assume that $L(x_0)$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method (Algorithm 1),*

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Proof. The proof is by contradiction. Suppose that $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then Proposition 3.4 tells us that there exists $\Delta_{LB} > 0$ such that $\Delta_k \geq \Delta_{LB}$. But this contradicts Theorem 3.3. \square

3.3.2. The Stronger Result. We can strengthen the result given in Theorem 3.5 at the expense of wider applicability. To begin with, we must add three further restrictions: one on the pattern matrix, one on the hypotheses on the exploratory moves, and one on the limiting behavior of the step length control parameter Δ_k .

First, we must ensure that the columns of the generating matrix C_k are bounded in norm, i.e., that there exists a constant $\mathcal{C} > 0$ such that for all k , $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \dots, p$. Given this bound, we can place an upper bound, in terms of Δ_k , on the norm of any trial step s_k^i .

LEMMA 3.6. *Given a constant $\mathcal{C} > 0$ such that for all k , $\mathcal{C} > \|c_k^i\|$, for all $i = 1, \dots, p$, there exists a constant $\psi_* > 0$, independent of k , such that for any trial step s_k^i produced by a generalized pattern search method (Algorithm 1) we have*

$$\Delta_k \geq \psi_* \|s_k^i\|.$$

Proof. From (4) we have $s_k^i = \Delta_k B c_k^i$. Then $\|s_k^i\| = \Delta_k \|B c_k^i\| \leq \Delta_k \|B\| \|c_k^i\| \leq \Delta_k C \|\mathcal{B}\|$. Set $\psi_* = \frac{1}{C \|\mathcal{B}\|}$. \square

Note that the columns of $M_k \in \mathbf{M}$ are bounded by the assumption that $|\mathbf{M}| < +\infty$; we use this fact in the proof of Proposition 7.4. The stronger boundedness condition on the columns of $C_k = [M_k - M_k \ L_k]$ is needed to monitor the behavior of L_k .

Second, we must replace the original Hypotheses on Exploratory Moves with a stronger version, as given below. Together, Lemma 3.6 and the Strong Hypotheses on

Strong Hypotheses on Exploratory Moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k [B \Gamma_k \ B L_k]$.
2. If $\min\{f(x_k + y), y \in \Delta_k B \Gamma_k\} < f(x_k)$, then $f(x_k + s_k) \leq \min\{f(x_k + y), y \in \Delta_k B \Gamma_k\}$.

Exploratory Moves allow us to tie decrease in f to the norm of the gradient when the step sizes get small enough. This is the import of Corollary 7.5, which is given in §7.

Third, we require that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. We can use the algorithm for updating Δ_k (Algorithm 2) to ensure that this condition holds. For instance, we can force Δ_k to be nonincreasing by requiring $w_i = 0, i = 1, \dots, L$, which when taken together with Theorem 3.3 guarantees that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. All the algorithms we will consider in §5, except the multidirectional search algorithm, enforce this condition by limiting $\Lambda = \{1\} \equiv \{\tau^0\}$. However, it is not necessary to force the steps to be nonincreasing; we need only require that in the limit the step length control parameter goes to zero, which, in conjunction with Lemmas 3.1 and 3.6, has the effect of ultimately forcing the steps to zero.

THEOREM 3.7. *Assume that $L(x_0)$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. In addition, assume that the columns of the generating matrix are bounded in norm, that the generalized pattern search method (Algorithm 1) enforces the Strong Hypotheses on Exploratory Moves, and that $\lim_{k \rightarrow +\infty} \Delta_k = 0$. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method,*

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

Proof. The proof is by contradiction. Suppose $\limsup_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Let $\varepsilon > 0$ be such that there exists a subsequence $\|\nabla f(x_{m_i})\| \geq \varepsilon$. Since

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0,$$

given any $0 < \eta < \varepsilon$, there exists an associated subsequence l_i such that

$$\|\nabla f(x_k)\| > \eta \text{ for } m_i \leq k < l_i, \quad \|\nabla f(x_{l_i})\| < \eta.$$

Then, since $\Delta_k \rightarrow 0$, we can appeal to Corollary 7.5 to obtain for $m_i \leq k < l_i$, i sufficiently large,

$$f(x_k) - f(x_{k+1}) \geq \sigma \|\nabla f(x_k)\| \|s_k\| \geq \sigma \eta \|s_k\|,$$

where $\sigma > 0$. Then the telescoping sum:

$$(f(x_{m_i}) - f(x_{m_i+1})) + (f(x_{m_i+1}) - f(x_{m_i+2})) + \cdots + (f(x_{l_i-1}) - f(x_{l_i})) \geq \sum_{k=m_i}^{l_i} \sigma \eta \|s_k\|$$

gives us

$$f(x_{m_i}) - f(x_{l_i}) \geq \sum_{k=m_i}^{l_i} \sigma \eta \|s_k\| \geq c' \|x_{m_i} - x_{l_i}\|.$$

Since $f(x_{m_i}) - f(x_{l_i}) \rightarrow 0$ as $i \rightarrow +\infty$, $\|x_{m_i} - x_{l_i}\| \rightarrow 0$ as $i \rightarrow +\infty$. Then, because ∇f is uniformly continuous,

$$\|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| < \eta,$$

for i sufficiently large. However,

$$(11) \quad \|\nabla f(x_{m_i})\| \leq \|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| + \|\nabla f(x_{l_i})\| \leq 2\eta.$$

Since (11) must hold for any η , $0 < \eta < \varepsilon$, we have a contradiction (e.g., try $\eta = \frac{\varepsilon}{4}$). \square

The proof of Theorem 3.7 is almost identical to that of an equivalent result for trust region methods that was first given by Thomas [18] and which is included, in a more general form, in the survey by Moré [11].

One final note: the hypotheses of Theorem 3.7 suggest that in the absence of any explicit higher-order information about the function to be minimized, it makes sense to terminate a generalized pattern search algorithm when Δ_k is less than some reasonably small tolerance. In fact, this is a common stopping condition for algorithms of this sort and the one implemented for the multidirectional search algorithm [21].

4. Parallels with Line Search and Model Trust Region Theory. The global convergence analysis we have just presented for pattern search methods shares similarities with the global convergence analysis for both line search and model trust region methods.

Parallels with the line search theory are perhaps most obvious and are discussed in [20]. The outline for the convergence theory follows the outline for global convergence theorems as detailed by Ortega and Rheinboldt [14] and reviewed in the survey by Nocedal [13]: we consider iterations of the form $x_{k+1} = x_k + s_k$ where $s_k \in \Delta_k P_k$; the columns of P_k determine the search directions and Δ_k serves as a step length parameter. In §7.1 we show that pattern search methods are *descent methods* (as defined in either [8] or [14]) by showing that if x_k is not a stationary point of the function then the generalized pattern search method guarantees decrease in the value of the objective function in a finite number of iterations. We then provide, in §7.2, a measure of the goodness of the search direction by showing that pattern search methods are *gradient-related methods* (as defined in [14]). Finally, in Theorem 3.3 and Proposition 3.4, we consider the length of the step.

Given these major components for the convergence analysis, the one that is most unusual is that involving the step length control. For a standard line search iteration, the

strategy for choosing the step relies on satisfying the Armijo–Goldstein–Wolfe conditions for *sufficient* decrease: the step length Δ_k is chosen to satisfy

$$f(x_k + \Delta_k p_k) \leq f(x_k) + \sigma_1 \Delta_k \nabla f(x_k)^T p_k$$

$$\nabla f(x_k + \Delta_k p_k)^T p_k \geq \sigma_2 \nabla f(x_k)^T p_k,$$

where p_k is the search direction and the constants σ_1 and σ_2 are chosen to satisfy $0 < \sigma_1 < \sigma_2 < 1$. The first condition prevents very small decreases in the function values relative to the lengths of the steps taken. The second condition prevents steps that are too small relative to the initial rate of decrease of f . The standard convergence analysis then proceeds on the assumption that these conditions for sufficient decrease have been satisfied.

The obvious difficulty that arises in any attempt to extend the notion of sufficient decrease to an iteration of any direct search method is that enforcement of the Armijo–Goldstein–Wolfe conditions requires the directional derivative—information that the direct search methods do not otherwise require.¹

The point of both Theorem 3.2 and Theorem 3.3 is that the Armijo–Goldstein–Wolfe conditions need not be enforced to guarantee weak first-order stationary point convergence. The reason is that pattern search methods place strict limitations on the choice of both the search directions and the step lengths. The set of search directions is limited to the columns of P_k . The step length parameter Δ_k must be a multiple of Δ_0 , θ , and $\lambda \in \Lambda$. These restrictions guarantee an underlying lattice structure as stated in Theorem 3.2. Once again we stress that this lattice structure is a purely algebraic fact about the nature of pattern search methods.

What does this algebraic structure mean for the convergence analysis? The answer lies in the simple control mechanisms imposed by the Hypotheses on the Exploratory Moves and the update algorithm for Δ_k (Algorithm 2). The Hypotheses on the Exploratory Moves require that *if* simple decrease can be found for some one of the steps defined by Γ_k , then the exploratory moves must return a step that gives simple decrease. This step can be defined by *any* column of the generating matrix. If we are lucky and guess right, we may only have to consider a single step at any given iteration; however, in the worst case we may have to consider all $2n$ steps defined by Γ_k . The role of the algorithm for updating Δ_k is to ensure that Δ_k is reduced *only* when the exploratory moves algorithm fails to produce a step that gives simple decrease on the value of the objective function.

The combination of these two mechanisms:

- Hypotheses on the Exploratory Moves
- Algorithm 2 for updating Δ_k

introduces backtracking into the generalized pattern search methods, which prevents steps that are too short. This is the import of Lemma 3.1.

¹ The analysis in [22] enforces a notion of sufficient decrease by imposing on the methods an “error-controlling sequence” that does not exist in the original algorithms. No suggestions are provided on how to construct such a sequence in practice.

The backtracking feature of the generalized pattern search methods is also critical because it guarantees that descent will be realized if the current iterate is not a stationary point of the function. The matrix Γ_k guarantees that a descent direction exists, even if we are not sure which of the $2n$ directions it may be. Thus we are assured that the pattern search methods will produce descent once Δ_k becomes small enough. However, the second of the Hypotheses on Exploratory Moves also prevents Δ_k from becoming unnecessarily fine if decrease can be found for one of the trial steps defined by $\Delta_k \Gamma_k$ and thereby ensures that $\{x_k\}$ cannot bog down at points not included in X_* .

The net effect of the algebraic structure of the pattern search methods, when combined with the Hypothesis on the Exploratory Moves and the algorithm for updating Δ_k , is to ensure that the pathologies which might otherwise occur were the Armijo–Goldstein–Wolfe conditions to be ignored (see [8]) cannot happen. These simple, if non-standard, mechanisms prevent the well-known pathologies that can arise if the length of the steps is not monitored.

Parallels with the global convergence theory for model trust region methods are perhaps less obvious. In fact, there is some temptation to call the pattern search methods *sample trust region* methods. This arises from the observation that in the basic version of each of the algorithms covered in §5, all possible steps can be seen to lie on the boundary of a trust region in a weighted l_∞ norm. We weight the norm by using the inverse of the basis matrix B . (Recall that the basis matrix is required to be nonsingular.) Thus, for all the pattern search methods (ignoring the acceleration steps found in the Hooke and Jeeves algorithm and the multidirectional search algorithm, which lie outside the trust region radius) all steps satisfy $\|B^{-1}s_k^i\|_\infty = \Delta_k$. In essence, Δ_k gives the radius of the region within which we trust our sampling to be effective. If the sampling does not produce simple decrease, we reduce the radius of the trust region.

It is worth comparing the global first-order stationary point convergence results for trust region methods and generalized pattern search methods. Roughly speaking, the result due to Powell [15] says that

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

assuming that ∇f is continuous and the Hessian (or its approximation) remains uniformly bounded. Thomas [18] assumes, in addition, that ∇f is uniformly continuous to prove that

$$\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$$

Neither result requires $L(x_0)$ to be compact; instead, f must be bounded below. We have shown comparable results for pattern search methods in Theorems 3.5 and 3.7 under the hypothesis that $L(x_0)$ is compact. The gap between Theorem 3.5 and Theorem 3.7 is due to the fact that while model trust regions link decrease in f to the norm of the gradient, the generalized pattern search methods do not explicitly enforce such a condition. Only when we place stronger conditions on the generating matrix, the

Hypotheses on the Exploratory Moves, and the limiting behavior of Δ_k , which together have the effect of *implicitly* linking decrease in f to the norm of the gradient, can we obtain the stronger convergence result. Nonetheless, it is perhaps surprising that we can prove any such convergence theorem for pattern search methods.

Having noted the unexpected similarities between pattern search methods and line search and trust region methods, we now note one critical difference: the step control mechanism for pattern search methods, however modest, is always active. Line search and model trust region strategies are extensions to Newton or quasi-Newton methods motivated by a need to ensure global convergence. These extensions are designed so that the step length control mechanism introduced by the line search or model trust region strategy “turns off” and the full Newton or quasi-Newton step is accepted once the local quadratic model of the function is deemed reliable. Thus, the fast local convergence properties of these algorithms are assured. However, at best the pattern search methods are relying on implicit information about the gradient; they do not construct local quadratic models of the function. The only natural notion of a full step is that given by Δ_k . As has long been recognized [16], this means that pattern search methods do not enjoy fast local convergence properties. When using these methods one trades speed of convergence for robustness and wider applicability.

One other point worth making is that pattern search methods are well-defined even when the function to be minimized is not differentiable. It is trivial to extend Theorem 3.5 to handle functions that are nondifferentiable; we need only modify the set X_* . This is reassuring in light of the fact that those who rely on these methods typically only assume that the function is continuous. The modifications required are discussed in further detail in [20]. The only qualification to be made is that the resulting theorem is not a general result for the nonsmooth case; rather, it is an extension of the result for the smooth case. Because the uniform continuity of the gradient over

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

is necessary for the proof of Theorem 3.5, the definition of X_* given in (1) must include not only the stationary points of the function, but also the points where the function is nondifferentiable and where the gradient exists but is not continuous.

5. The Particular Pattern Search Methods. In §2 we stated the conditions an algorithm must satisfy to be a pattern search method. We now illustrate these conditions by considering the following specific algorithms:

- coordinate search with fixed step lengths,
- searches based on two-level factorial designs (see [4] and [5]),
- the original pattern search method of Hooke and Jeeves [10], and
- the multidirectional search algorithm of Dennis and Torczon ([9] and [19]).

We will show that these algorithms satisfy the conditions that define pattern search methods and thus are special cases of the generalized pattern search method presented as Algorithm 1. Then we can appeal to Theorem 3.5 to claim weak first-order stationary point convergence for these methods.

There are undoubtedly other algorithms for which the abstraction and accompanying analysis holds—including various modifications to the algorithms presented—but we shall confine our investigation to these, the best known of the pattern search methods, to illustrate the power of our abstract approach to pattern search methods.

5.1. Coordinate Search with Fixed Step Lengths. The method of coordinate search is the simplest and most obvious of all the pattern search methods. Davidon describes it concisely in the opening of his belated preface to Argonne National Laboratory Research and Development Report 5990 [7]:

Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure....

This simple search method enjoys many names, among them *alternating directions*, *alternating variable search*, *axial relaxation*, and *local variation*. We shall refer to it as *coordinate search*.

Perhaps less obvious is that coordinate search is a pattern search method. To see this, we begin by considering all possible outcomes for a single iteration of coordinate search when $n = 2$, as shown in Fig. 1. We mark the current iterate x_k . The x_k^i 's denote trial points considered during the course of the iteration. The next iterate x_{k+1} is marked. Solid circles indicate successful intermediate steps taken during the course of the iteration while open circles indicate points at which the function was evaluated but that did not produce further decrease in the value of the objective function. Thus, in the first scenario shown a step from x_k to x_k^1 resulted in a decrease in the objective function, so the step from x_k^1 to x_{k+1} was tried and led to a further decrease in the objective function value. The iteration was then terminated with a new point x_{k+1} that satisfies the simple decrease condition $f(x_{k+1}) < f(x_k)$. In the worst case, the last scenario shown, $2n$ trial points were evaluated (x_k^1 , $x_k^{1'}$, x_k^2 , and $x_k^{2'}$) without producing decrease in the function value at the current iterate x_k . In this case, $x_{k+1} = x_k$ and the step size must be reduced for the next iteration.

We now show this algorithm is an instance of a generalized pattern search method.

5.1.1. The Matrices. Coordinate search is usually defined so that the basis matrix is the identity matrix; i.e., $B = I$. However, knowledge of the problem may lead to a different choice for the basis matrix. It may make sense to search using a different coordinate system. For instance, if the variables are known to differ by several orders of magnitude, this can be taken into account in the choice of the basis matrix (though, as we will see in §7.2, this may have a significant effect on the behavior of the method).

The generating matrix for coordinate search is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combina-

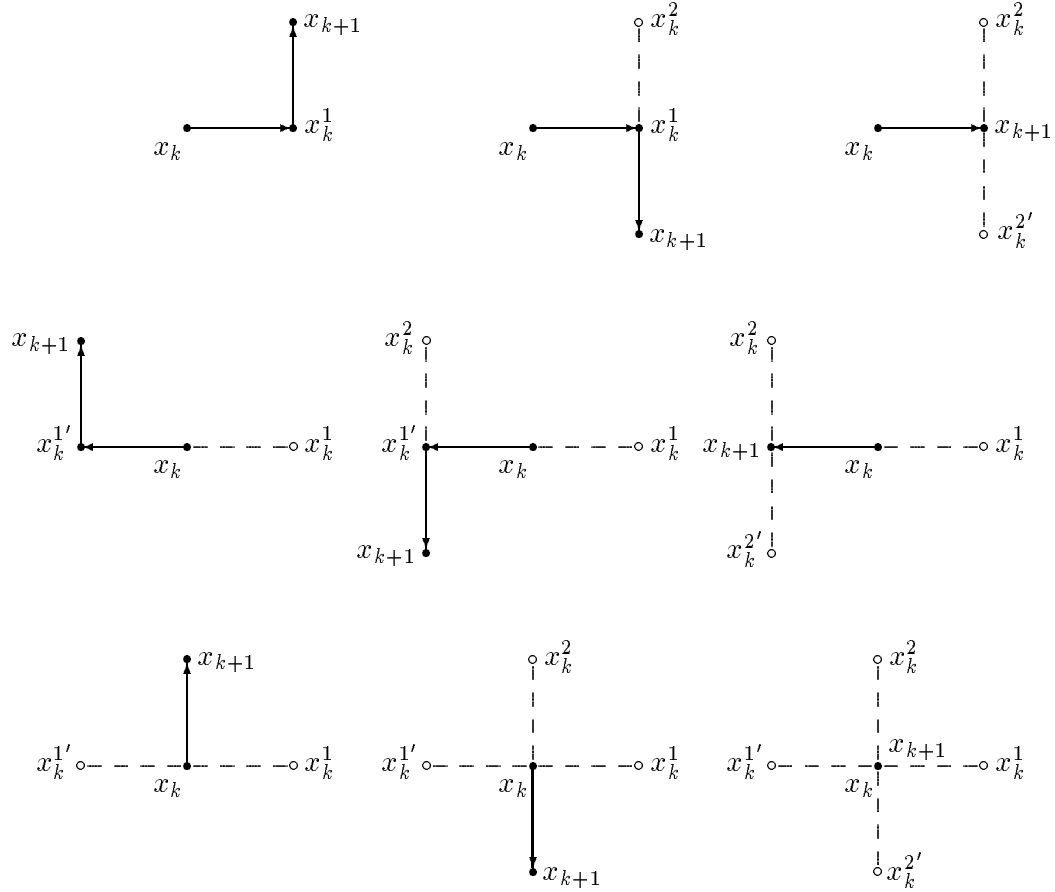


FIG. 1. All possible subsets of the steps for coordinate search in \mathbf{R}^2 .

tions of $\{-1, 0, 1\}$. Thus, C has $p = 3^n$ columns. In particular, the columns of C will contain both I and $-I$, as well as a column of zeros. We define $M = I$; L consists of the remaining $3^n - 2n$ columns of C . Since C is fixed across all iterations of the method, there is no need for an update algorithm.

For $n = 2$ we have

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

Thus, when $n = 2$, all possible trial points defined by the pattern $P = BC$, for a given step length Δ_k , can be seen in Fig. 2. Note that the pattern includes all the possible trial points enumerated in Fig. 1.

5.1.2. The Exploratory Moves. The exploratory moves for coordinate search are given in Algorithm 3, where the e_i 's denote the unit coordinate vectors.

The exploratory moves are executed sequentially in the sense that the selection of the next trial step is based on the success or failure of the previous trial step. Thus, while there are 3^n possible trial steps, we may compute as few as n trial steps, but we compute no more than $2n$ at any given iteration, as we saw in Fig. 1.

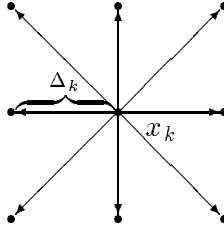


FIG. 2. The pattern for coordinate search in \mathbf{R}^2 with a given step length Δ_k .

Algorithm 3. Exploratory Moves Algorithm for Coordinate Search.

Given x_k , Δ_k , $f(x_k)$, and B , set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, n$ do

- a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.
Otherwise,
 - i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
 - ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

From the perspective of the theory, there are two conditions that need to be met by the exploratory moves algorithm. First, as Figs. 1 and 2 illustrate, all possible trial steps are contained in $\Delta_k P$.

The second condition on the exploratory moves is the more interesting; coordinate search demonstrates the laxity of this second hypothesis. For instance, in the first scenario shown in Fig. 1, decrease in the objective function was realized for the first trial step

$$s_k^1 = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

so the second trial step

$$s_k^2 = \Delta_k I \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

was tried, and accepted. It is certainly possible that greater decrease in the value of the objective function might have been realized for the trial step

$$s_k' = \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which is defined by a column in the matrix M (the step s_k^2 is defined by a column in the matrix L), but s_k' is not tried when simple decrease is realized by the step s_k^1 . However, *in the worst case*, as seen in Fig. 1, the algorithm for coordinate search ensures that all $2n$ steps defined by $\Delta_k B \Gamma = \Delta_k B [M - M] = \Delta_k B [I - I]$ are tried before returning the step $s_k = 0$. In other words, the exploratory moves given in Algorithm 3 will examine all $2n$ steps defined by $\Delta_k B \Gamma$ *unless* a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

5.1.3. Updating the Step Length. The update for Δ_k is exactly as given in Algorithm 2. As noted by Davidon, the usual practice is to continue with steps of the same magnitude until no further decrease in the objective function is realized, at which point the step size is halved. This corresponds to setting $\theta = 1/2$ and $\Lambda = \{1\}$. Thus, $\tau = 2$, $w_0 = -1$, and $w_1 = 0$.

This suffices to verify that coordinate search with fixed step length is a pattern search method. Theorem 3.5 thus holds. The exploratory moves algorithm for coordinate search would need to be modified to satisfy the Strong Hypotheses on Exploratory Moves for the conditions of Theorem 3.7 to be met, though this is a straightforward modification.

5.2. Response Surface Methodology. In 1951, Box and Wilson [4] introduced the notion of “response surface methodology” as a way to investigate an objective function by performing function evaluations at the vertices of some geometric configuration in the space of independent variables. This paper prepared the way for the development of direct search methods, in general, and what we now call pattern search methods, in particular.

In its earliest forms, response surface methodology was based on two-level factorial designs: evaluate the function at the vertices of a hypercube centered about the current iterate. (In fact, Box refers to this as one of a variety of “pattern of variants” [3].) If simple decrease in the value of the objective function is observed at one of the vertices, it becomes the new iterate. Otherwise, the lengths of the edges in the hypercube are halved and the process is repeated. Further discussions of the basic approach can be found in [5] and [17].

5.2.1. The Matrices. As with coordinate search, the usual choice for the basis matrix is $B = I$, though, as with coordinate search, other choices may be made to reflect information known about the problem to be solved.

The generating matrix for response surface methodology is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combinations of $\{-1, 1\}$; to this we append a column of zeros. Thus C has $p = 2^n + 1$ columns.

We take M to be any linearly independent subset of n columns of C ; $-M$ necessarily will be contained in C . Once again, L is fixed and consists of the remaining $(2^n + 1) - 2n$ columns of C .

Since the generating matrix is fixed, there is no need for an algorithm to update C .

5.2.2. The Exploratory Moves. The exploratory moves given in Algorithm 4 are simultaneous in the sense that every possible trial step $s_k^i \in \Delta_k P = \Delta_k BC$ is computed at each iteration. It is then the case that every trial step s_k^i is contained in $\Delta_k P$. The second observation of note is that since

$$s_k = \arg \min_{s_k^i \in \Delta_k P} \{f(x_k + s_k^i)\},$$

then, if $\min\{f(x_k + y), y \in \Delta_k B\Gamma\} < f(x_k)$, we have $f(x_k + s_k) < f(x_k)$, regardless of our choice of M (and thus, by extension, our choice of Γ). Furthermore, we are guaranteed that the Strong Hypotheses on Exploratory Moves are satisfied.

Algorithm 4. Exploratory Moves Algorithm for Response Surface Methodology.

Given x_k , Δ_k , $f(x_k)$, B , and $C = [c^1 \cdots c^p]$, set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, 2^n$ do

a) $s_k^i = \Delta_k B c^i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

5.2.3. Updating the Step Length. The algorithm for updating Δ_k is exactly as given in Algorithm 2, with θ usually set to $1/2$ and $\Lambda = \{1\}$.

Since we have shown that response surface methodology satisfies all the necessary requirements, we can therefore conclude that it, too, is a pattern search method, so Theorem 3.5 holds. The algorithm, as stated above, also satisfies the conditions of Theorem 3.7.

5.3. Hooke and Jeeves' Pattern Search Algorithm. In addition to introducing the general notion of a “direct search” method, Hooke and Jeeves introduced the pattern search method—a specific kind of search strategy—in their 1961 paper [10]. The pattern search of Hooke and Jeeves is essentially a variant of coordinate search that incorporates a *pattern step* in an attempt to accelerate the progress of the algorithm by exploiting information gained from the search during previous successful iterations.

The Hooke and Jeeves pattern search algorithm is opportunistic. If the previous iteration was successful (i.e., $\rho_{k-1} > 0$), then the current iteration begins by conducting coordinate search about a speculative iterate $x_k + (x_k - x_{k-1})$, rather than about the current iterate x_k . This is the pattern step. The idea is to investigate whether further progress is possible in the general direction $x_k - x_{k-1}$ (since, if $x_k \neq x_{k-1}$, then $x_k - x_{k-1}$ is clearly a promising direction).

To make this a little clearer, we consider the example shown in Fig. 3. Given

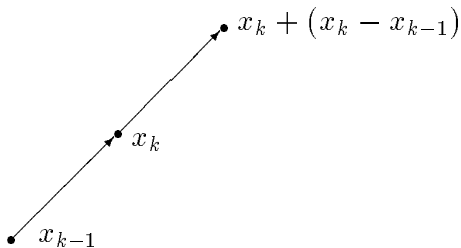


FIG. 3. The pattern step in \mathbf{R}^2 , given $x_k \neq x_{k-1}$, $k > 0$.

x_{k-1} and x_k (we assume, for now, that $k > 0$ and that $x_k \neq x_{k-1}$), the pattern search algorithm takes the step $x_k - x_{k-1}$ from x_k . The function is evaluated at this trial step and it is accepted, temporarily, *even if* $f(x_k + (x_k - x_{k-1})) \geq f(x_k)$. The Hooke and Jeeves pattern search algorithm then proceeds to conduct coordinate search about the

temporary iterate $x_k + (x_k - x_{k-1})$. Thus, in \mathbf{R}^2 , the exploratory moves are exactly as shown in Fig. 1, but with $x_k + (x_k - x_{k-1})$ substituted for x_k .

If coordinate search about the temporary iterate $x_k + (x_k - x_{k-1})$ is successful, then the point returned by coordinate search about the temporary iterate is accepted as the new iterate x_{k+1} . If not, i.e., $f((x_k + (x_k - x_{k-1})) + s_k) \geq f(x_k)$, then the pattern step is deemed unsuccessful, and the method reduces to coordinate search about x_k . For the two dimensional case, then, the exploratory moves would simply resort to the possibilities shown in Fig. 1.

If the previous iteration was not successful, so $x_k = x_{k-1}$ and $(x_k - x_{k-1}) = 0$, then the iteration is limited to coordinate search about x_k . In this instance, though, the updating algorithm for Δ_k will have reduced the size of the step (i.e., $\Delta_k = \theta\Delta_{k-1}$). The algorithm does not execute the pattern step when $k = 0$.

To express the pattern search algorithm within the framework we have developed, we use all the machinery required for coordinate search. Once again, the basis matrix is usually defined to be $B = I$. We append to the generating matrix another set of 3^n columns to capture the effect of the pattern step and we change the exploratory moves algorithm, as detailed below.

5.3.1. The Generating Matrix. Recall that the generating matrix for coordinate search consists of all possible combinations of $\{-1, 0, 1\}$ and is never changed. For the Hooke and Jeeves pattern search method, we allow the generating matrix to change from iteration to iteration to capture the effect of the pattern step. We append another set of 3^n columns, consisting of all possible combinations of $\{-1, 0, 1\}$, to the initial generating matrix for coordinate search. Thus C_k has $p = 2 \cdot 3^n$ columns. The additional 3^n columns allow us to express the effect of the pattern step with respect to x_k , rather than with respect to the temporary iterate $x_k + (x_k - x_{k-1})$, which is how the Hooke and Jeeves pattern search method usually is described. The matrix M is unchanged; $M = I$. Now, however, $L_k \in \mathbf{Z}^{n \times (p-2n)}$ is allowed to vary, though only in the 3^n columns associated with the pattern step. For $n = 2$,

$$(12) \quad C_0 = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

For notational convenience, we require that the last column of C_0 , which we denote as c_0^p , be the column of zeros. In both the algorithm for updating C_k (Algorithm 5) and the algorithm for the exploratory moves (Algorithm 6), we use the column c_k^p to measure the accumulation of a sequence of successful pattern steps. This can be seen, in (13), for our example from Fig. 3. In this example, we have the generating matrix

$$(13) \quad C_k = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 2 & 1 & 0 & 1 & 2 & 2 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 1 & 2 & 1 & 0 & 2 & 0 & 0 & 2 & 1 \end{bmatrix}.$$

The pattern step $(x_k - x_{k-1})$ is represented by the vector $(1 \ 1)^T$, seen in the last column of C_k . Note that the only difference between the columns of C_0 given in (12) and the columns of C_k given in (13) is that $(1 \ 1)^T$ has been added to the last 3^2 columns of C_k .

The algorithm for updating the generating matrix updates the last 3^n columns of C_k ; the first 3^n columns remain unchanged, as in coordinate search. The purpose of the updating algorithm is to incorporate the result of the search at the current iteration into the pattern for the next iteration. This is done using Algorithm 5. Note the distinguished role of c_k^p , the last column of C_k , which represents the pattern step $(x_k - x_{k-1})$.

Algorithm 5. Updating C_k .

For $i = 3^n + 1, \dots, 2 \cdot 3^n$ do

$$c_{k+1}^i = c_k^i + (1/\Delta_k)s_k - c_k^p.$$

Return.

Since $(1/\Delta_k)s_k$ is necessarily a column of C_k , and $C_0 \in \mathbf{Z}^{n \times p}$, an argument by induction shows that the update algorithm for C_k ensures that the columns of C_k always consist of integers.

5.3.2. The Exploratory Moves. In Algorithm 6, the e_i 's denote the unit coordinate vectors and c_k^p denotes the last column of C_k . We set $\rho_{-1} = 0$ so that ρ_{k-1} is defined when $k = 0$.

A useful example for working through the logic of the algorithm can be found in [1], though the presentation and notation differ somewhat from that given here.

Algorithm 6. Exploratory Moves Algorithm for Hooke and Jeeves.

Given x_k , Δ_k , $f(x_k)$, B , and ρ_{k-1} , set $\rho_k = \rho_{k-1}$ and $min = f(x_k)$.

If $\rho_k > 0$ then set $s_k = \Delta_k B c_k^p$, $\rho_k = f(x_k) - f(x_k + s_k)$, and $min = f(x_k + s_k)$.

For $i = 1, \dots, n$ do

- a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Otherwise,

- i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

If $\rho_k \leq 0$ then set $s_k = 0$, $\rho_k = 0$, and $min = f(x_k)$.

For $i = 1, \dots, n$ do

- a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Otherwise,

- i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- ii) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, and $s_k = s_k^i$.

Return.

All possible steps are contained in $\Delta_k P_k$ since C_k contains columns that represent the ‘‘pattern steps’’ tried at the beginning of the iteration. And, once again, the exploratory moves given in Algorithm 6 will examine all $2n$ steps defined by $\Delta_k B \Gamma$ unless a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

Since we have shown that the pattern search algorithm of Hooke and Jeeves satisfies all the necessary requirements, we can therefore conclude that it, too, is a special case of the *generalized* pattern search method and Theorem 3.5 holds.

5.4. Multidirectional Search. The multidirectional search algorithm was introduced by Dennis and Torczon in 1989 [19] as a first step towards a general purpose optimization algorithm with promising properties for parallel computation. While subsequent work led to a class of algorithms, based on the multidirectional search algorithm, that allows for more flexible computation ([9] and [21]) one of the unanticipated results of the original research was a weak first-order stationary point convergence theorem for the multidirectional search algorithm [20].

The multidirectional search algorithm is a simplex-based algorithm. The pattern of points can be expressed as a simplex (i.e., $n + 1$ points, or vertices) based at the current iterate; as such, multidirectional search owes much in its conception to its predecessors, the simplex design algorithm of Spendley, Hext, and Himsworth [16] and the simplex algorithm of Nelder and Mead [12]. However, multidirectional search is a different algorithm—particularly from a theoretical standpoint. Convergence for the Spendley, Hext and Himsworth algorithm can be shown only with some modification of the original algorithm, and then only under the additional assumption that the function f is convex. There are numerical examples to demonstrate that the Nelder–Mead simplex algorithm may fail to converge to a stationary point of the function because the uniform linear independence property (discussed in §7.2), which plays a key role in the convergence analysis, cannot be guaranteed to hold [19].

The multidirectional search algorithm is described in detail in both [9] and [20]. The formulation given here is different and, in fact, introduces some redundancy that can be eliminated when actually implementing the algorithm. However, the way of expressing the algorithm that we will use here allows us to make clear the similarities between this and other pattern search methods.

5.4.1. The Matrices. It is most natural to express multidirectional search in terms of *multiple* basis matrices B_k and a *fixed* generating matrix C , which is at odds with our definition for generalized pattern search methods. As we shall see, however, it is possible to convert the more natural specification to one that conforms to our requirements for a pattern search method.

The multidirectional search algorithm centers around a family of basis matrices \mathbf{B} that consists of all matrices representing the edges adjacent to each vertex in a nondegenerate n -dimensional simplex that the user is allowed to specify. Since the ordering of the columns is not unique, and typically not preserved in the implementation of the method, we consider all possible representations of the columns of the matrices associated with the edges adjacent to the $(n + 1)$ vertices of the simplex. We then add the negatives of these $(n + 1)!$ basis matrices to account for the effect of the *reflection* step allowed by the multidirectional search algorithm. Thus the cardinality of the set \mathbf{B} is $|\mathbf{B}| = 2(n + 1)!$.

Fortunately, there is no need to actually construct and “store” this unwieldy number

of basis matrices to initialize the method. We can update the basis matrix after each iteration k by reconstructing the new basis matrix B_{k+1} , given the outcome of the exploratory moves, from the trial points x_k^i , $i = 1, \dots, n$, considered during the course of the exploratory moves. This procedure is given in Algorithm 8. The scalar *scale* is chosen during the course of the exploratory moves (see Algorithm 7) to ensure that $B_{k+1} \in \mathbf{B}$ by factoring out any change in the size of the simplex introduced by a change in Δ_k . This has the further effect of preserving the role of Δ_k as a step length parameter.

Algorithm 8. Updating B_k .

Given B_k , *scale*, *best*, and x_k^i for $i = 0, \dots, n$, denote $B_k = [b_k^i]$, $i = 1, \dots, n$.

If $\rho_k > 0$ then

For $i = 0, \dots, (\text{best} - 1)$ do

$$b_{k+1}^{i+1} = \text{scale} * (x_k^i - x_k^{\text{best}}).$$

For $i = (\text{best} + 1), \dots, n$ do

$$b_{k+1}^i = \text{scale} * (x_k^i - x_k^{\text{best}}).$$

Otherwise

For $i = 1, \dots, n$ do

$$b_{k+1}^i = b_k^i.$$

Return.

Given this use of a family of basis matrices to help define the multidirectional search algorithm, the generating matrix is then the fixed matrix $C = [I \ -I \ -\mu I \ 0]$. Thus, C contains $p = 3n + 1$ columns, with $M = I$. To ensure that $C \in \mathbf{Z}^{n \times p}$, we require $\mu \in \mathbf{Z}$. Furthermore, to ensure that the role of Δ_k as a step length parameter is not lost with the introduction of the *expansion* step represented by $-\mu I$, we require $\mu \in \Lambda$. The algorithm is defined so that $\Lambda = \{\tau^{w_1}, \tau^{w_2}\}$, with $\mu = \tau^{w_2}$. This requires the further restriction that $\tau \in \mathbf{N}$. Again, this is not an onerous restriction. Multidirectional search usually is specified so that $\tau = 2$, $w_2 = 1$, and thus $\mu = 2$.

Now, to bring this notation into conformity with our definition for a generalized pattern search method, observe that we can represent all possible basis matrices $B_\nu \in \mathbf{B}$ in terms of a single reference matrix $B \in \mathbf{B}$ so that

$$B_\nu = B \hat{B}_\nu, \quad \nu = 1, \dots, |\mathbf{B}|.$$

A convenient feature of using the edges of a simplex to form the set of basis matrices is that the matrices \hat{B}_ν consist only of elements from the set $\{-1, 0, 1\}$. The matrices \hat{B}_ν are necessarily nonsingular because of the nondegeneracy of the simplex. We use $\hat{\mathbf{B}}$ to represent the set of matrices \hat{B}_ν and observe that since \mathbf{B} is a finite set, the set $\hat{\mathbf{B}}$ is also finite.

We then observe that

$$P_k = B_k C = B_k [I \ -I \ -\mu I \ 0] \equiv B [\hat{B}_k \ -\hat{B}_k \ -\mu \hat{B}_k \ 0] = B C_k.$$

Thus we can define the pattern in terms of the single reference matrix B and the redefined generating matrix

$$C_k = [\hat{B}_k \ -\hat{B}_k \ -\mu \hat{B}_k \ 0],$$

with $M_k = \hat{B}_k$ and $\mathbf{M} \equiv \hat{\mathbf{B}}$. Since $\mu \in \mathbf{Z}$, $L_k \in \mathbf{Z}^{n \times (n+1)}$ and the column of zeros has been added explicitly.

5.4.2. The Exploratory Moves. The exploratory moves for the multidirectional search method are given in Algorithm 7; the e_i 's denote the unit coordinate vectors. We use the notion of $B_k \in \mathbf{B}$ for consistency with the update algorithm given in Algorithm 6, but we could just as easily substitute $B\hat{B}_k$ for B_k in the algorithm given below.

Algorithm 7. Exploratory Moves Algorithm for Multidirectional Search.

Given x_k , Δ_k , $f(x_k)$, B_k , and $\mu = \tau^{w_2} \in \mathbf{N}$, set $s_k = 0$, $\rho_k = 0$, $min = f(x_k)$, $\lambda_k = 1$, $scale = 1/\Delta_k$, $best = 0$, and $x_k^0 = x_k$.

For $i = 1, \dots, n$ do

a) $s_k^i = \Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, and $best = i$.

If $\rho_k \leq 0$ then

For $i = 1, \dots, n$ do

a) $s_k^i = -\Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, and $best = i$.

If $\rho_k > 0$ then set $scale = 1/\mu\Delta_k$.

For $i = 1, \dots, n$ do

a) $s_k^i = -\mu\Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.

b) If $f(x_k^i) < min$ then $\rho_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$, $s_k = s_k^i$, $best = i$, and $\lambda_k = \mu$.

Return.

Clearly, $s_k \in \Delta_k P_k$. Since the exploratory moves algorithm will consider all steps of the form $\Delta_k B\Gamma_k$, *unless* simple decrease is found after examining only the steps defined by $\Delta_k B M_k$, this guarantees we satisfy the condition that if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

5.4.3. Updating the Step Length. The algorithm for updating Δ_k is that given in Algorithm 2. In this case, while θ usually is set to $1/2$ so that $\tau = 2$, $w_0 = -1$, and $w_1 = 0$, we also include an expansion factor $\mu = \tau^{w_2}$, where w_2 usually equals one. Thus $\Lambda = \{1, \mu\}$, where μ is usually 2. The choice of $\lambda_k \in \Lambda$ is made during the execution of the exploratory moves.

Since we have shown that the multidirectional search algorithm satisfies all the necessary requirements, we conclude that it is also a pattern search method and thus Theorem 3.5 applies. Note that since we allow $\mu > 1$, which is a useful algorithmic feature, we cannot guarantee that $\lim_{k \rightarrow +\infty} \Delta_k = 0$ and so Theorem 3.7 does not automatically apply.

6. Conclusions. We have presented a framework in which one can analyze pattern search methods. This framework abstracts and quantifies the similarities of the classical pattern search methods and enables us to prove weak first-order stationary

point convergence for this class of algorithms. We also specify the conditions under which first-order stationary point convergence can be shown to hold.

These convergence results are perhaps surprising, given the simplicity of pattern search methods, but derive from the algebraic rigidity imposed on the iterates produced by pattern search methods. This is gratifying, since while this rigidity originally was introduced as a heuristic for directing the local search, it turns out to be the key to proving convergence as well. This analysis also highlights just how weak the conditions on the acceptance of the step can be and yet still allow a global convergence analysis, an observation that may prove useful in the analysis of other classes of optimization methods.

7. Appendix. We deferred the proof of Proposition 3.4 for several reasons. First, many of the results in this section are generalizations of similar results to be found in [20]. The abstraction in §2 leads to more succinct proofs. Second, the proof of Proposition 3.4 is closely related to that of several other results presented in this section and requires us to introduce several additional notions before tackling its proof.

We return to our definition of the pattern as $P_k = BC_k$ to show that the pattern contains at least one direction of descent whenever $\nabla f(x_k) \neq 0$.

Recall that we require the columns of C_k to contain both M_k and $-M_k$. Thus, P_k can be partitioned as follows:

$$P_k = BC_k = B [M_k \quad -M_k \quad L_k] = B [\Gamma_k \quad L_k].$$

We now elaborate on these requirements. Since M_k is an $n \times n$ nonsingular matrix, and B is nonsingular, we are guaranteed that BM_k forms a basis for \mathbf{R}^n . Thus at any iteration k , if $\nabla f(x_k) \neq 0$ we are guaranteed that Bc_k^i will be a direction of descent for at least one column c_k^i contained in the block Γ_k .

7.1. Descent Methods. Of course, the existence of a trial step in a descent direction is not sufficient to guarantee that decrease in the value of the objective function will be realized. To guarantee that a pattern search method is a descent method, we need to guarantee that in a finite number of iterations the method will produce a positive step size Δ_k that achieves decrease on the objective function at the current iterate. We now show that this is the case.

LEMMA 7.1. *Suppose that f is continuously differentiable on $L(x_0)$. If $\nabla f(x_k) \neq 0$, then there exists $p \in \mathbf{Z}$, $p \geq 0$, such that $\rho_{k+p} > 0$ (i.e., the $(k+p)^{\text{th}}$ iteration is successful).*

Proof. A key hypothesis placed on the exploratory moves is that if descent can be found for some one of the trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves will return a step that produces descent.

Because BC_k has rank n , if $\nabla f(x_k) \neq 0$, then there exists at least one trial direction $d_k^i = Bc_k^i$, where $c_k^i \in \Gamma_k$, such that $\nabla f(x_k)^T d_k^i \neq 0$. But, since $-c_k^i \in \Gamma_k$, without loss of generality, $\nabla f(x_k)^T d_k^i < 0$. Thus, there exists an $h_k > 0$ such that for $0 < h \leq h_k$, $f(x_k + hd_k^i) < f(x_k)$.

If at iteration k , $\Delta_k > h_k$, then the iteration may be unsuccessful; that is, $\rho_k = f(x_k) - f(x_k + s_k) \leq 0$. When the iteration is unsuccessful, the generalized pattern

search method sets $x_{k+1} = x_k$ and the updating algorithm sets $\Delta_{k+1} = \theta\Delta_k$. Since θ is strictly less than one, there exists $p \in \mathbf{Z}$, $p \geq 0$, such that $\theta^p\Delta_k \leq h_k$. Thus we are guaranteed descent, i.e., a successful iteration, in at most p iterations. \square

7.2. Uniform Linear Independence. The pattern P_k guarantees the existence of at least one direction of descent whenever $\nabla f(x_k) \neq 0$. We now want to guarantee the existence of a bound on the angle between the direction of descent contained in $B\Gamma_k$ and the negative gradient at x_k (whenever $\nabla f(x_k) \neq 0$). We will show, in fact, that this bound is uniform across all iterations of the pattern search algorithm. To do so, we use the notion of *uniform linear independence* [14].

LEMMA 7.2. *For a pattern search algorithm, there exists a constant $\xi > 0$ such that for all $k \geq 0$ and $x \neq 0$,*

$$(14) \quad \max \left\{ \frac{|x^T(x_k^i - x_k)|}{\|x\|\|x_k^i - x_k\|}, i = 1, \dots, p \right\} \geq \xi.$$

Proof. To demonstrate the existence of ξ , we first consider the simplest possible case, $B = I$ and $C = [M \ -M \ 0] = [I \ -I \ 0]$, and use this to derive a bound for any choice of B and C_k that satisfies the conditions we have imposed.

LEMMA 7.3. *Suppose $\|y\| = 1$. Let*

$$|\cos \theta(y)| = \max_{1 \leq j \leq n} \{ |y^T e_j| \},$$

where the e_j 's are the unit coordinate vectors.

If $B = I$ and $C = [I \ -I \ 0]$, then

$$\min_{y \in \mathbf{R}^n} |\cos \theta(y)| = \frac{1}{\sqrt{n}}.$$

Proof. We have $|y^T e_j| = |y_j|$, where $y = (y_1, \dots, y_n)^T$. Since $\sum_{j=1}^n |y_j|^2 = 1$, we are guaranteed that $|y_j| \geq 1/\sqrt{n}$ for some j , so $|y^T e_j| \geq 1/\sqrt{n}$ for some j . Thus $|\cos \theta(y)| \geq 1/\sqrt{n}$.

Now note that $\cos \theta(y)$ attains this lower bound for any $y = \alpha_1 e_1 + \alpha_2 e_2 + \dots + \alpha_n e_n$, where $\alpha_j = \pm 1/\sqrt{n}$. \square

Thus, if the pattern search is restricted to the coordinate directions defined by $P = [I \ -I \ 0]$, $\xi = 1/\sqrt{n}$ gives the lower bound on the absolute value of the cosine of the angle between the gradient and a guaranteed direction of descent. We now use the bound for this particular case to derive a bound for the general case.

Assume a general basis matrix B and a general matrix $M_k \in \mathbf{M}$, where $|\mathbf{M}| < +\infty$. We adopt the notation $BM_k = [y_k^1 \ \dots \ y_k^n]$. Then for any $x \neq 0$ we have the following:

$$|\cos \theta| = \frac{|x^T y_k^j|}{\|x\|\|y_k^j\|} = \frac{|x^T BM_k e_j|}{\|x\|\|BM_k e_j\|} = \frac{|((BM_k)^T x)^T e_j|}{\|x\|\|BM_k e_j\|}.$$

If we set $w = (BM_k)^T x$, so that $x = (BM_k)^{-T} w$, we have

$$\begin{aligned} |\cos \theta| &= \frac{|w^T e_j|}{\|(BM_k)^{-T} w\| \|BM_k e_j\|} \geq \frac{|w^T e_j|}{\|(BM_k)^{-T}\| \|w\| \|BM_k\| \|e_j\|} \\ &= \frac{1}{\|(BM_k)^{-T}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) = \frac{1}{\|(BM_k)^{-1}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) \\ &\geq \frac{1}{\kappa(BM_k)} \frac{1}{\sqrt{n}}, \end{aligned}$$

where $\kappa(BM_k)$ is the condition number of the matrix BM_k . Thus, we have

$$|\cos \theta| \geq \frac{1}{\kappa(BM_k) \sqrt{n}} > 0.$$

To ensure a bound ξ that is independent of the choice of any particular matrix $M \in \mathbf{M}$, we simply observe that the set \mathbf{M} is required to be finite. Thus, ξ is taken to be

$$(15) \quad \xi = \min_{M \in \mathbf{M}} \left\{ \frac{1}{\kappa(BM) \sqrt{n}} \right\}.$$

□

The bound given in (15) points to two features that explain much about the behavior of pattern search methods. Since we never explicitly calculate—or approximate—the gradient, we are dependent on the fact that in the worst case at least one of our search directions is not orthogonal to the gradient; ξ gives us a bound on how far away we can be. Thus, as either the condition number of the product BM_k increases, or the dimension of the problem increases, our bound on the angle between the search direction and the gradient deteriorates. This suggests two things. First, we should be very careful in our choice of B and \mathbf{M} for any particular pattern search method. Second, we should not be surprised that these methods become less effective as the dimension of the problem increases.

Nevertheless, even though pattern search methods neither require nor explicitly approximate the gradient of the function, the uniform linear independence condition demonstrates that the pattern search methods are, in fact, *gradient-related methods*, as defined by Ortega and Rheinboldt [14], which is one reason why we can establish global first-order stationary point convergence.

7.3. The Descent Condition. Having introduced the notion of uniform linear independence with the bound ξ , we are now ready to show that pattern search methods reduce Δ_k only when necessary to find descent. This enables us to prove Proposition 3.4.

PROPOSITION 7.4. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on $L(x_0)$. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exists $\delta > 0$, independent of k , such that if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta$, then the k th iteration of a generalized pattern search method (Algorithm 1) will be successful (i.e., $\rho_k = f(x_k) - f(x_k + s_k) > 0$) and thus $\Delta_{k+1} \geq \Delta_k$.

Proof. We restrict our attention to the steps defined by the columns of $\Delta_k B\Gamma_k$. This is sufficient since the Hypotheses on Exploratory Moves ensure that a step s_k satisfying the simple decrease condition $\rho_k > 0$ must be returned if a trial step defined by a column of $\Delta_k B\Gamma_k$ satisfies the simple decrease condition.

We first need some measure of the relative lengths of the steps defined by M_k and its negative $-M_k$. We begin by defining

$$e^k = \min_{i=1, \dots, 2n} \|x_k^i - x_k\| \equiv \min_{i=1, \dots, 2n} \|s_k^i\| \equiv \min_{j=1, \dots, n} \Delta_k \|Bc_k^j\|$$

and

$$E^k = \max_{i=1, \dots, 2n} \|x_k^i - x_k\| \equiv \max_{i=1, \dots, 2n} \|s_k^i\| \equiv \max_{j=1, \dots, n} \Delta_k \|Bc_k^j\|,$$

where c_k^i is restricted to be a column of $M_k \in \mathbf{M}$. (We assume that P_k is partitioned as in (3) so that the first $2n$ columns of P_k contain the columns of $[B\Gamma_k] = [BM_k - BM_k]$.) Since $|\mathbf{M}| < +\infty$, we can define η as

$$\eta = \min_{M \in \mathbf{M}} \frac{\min_{j=1, \dots, n} \|Bc^j\|}{\max_{j=1, \dots, n} \|Bc^j\|},$$

where c^j is a column of $M \in \mathbf{M}$. Observe that $0 < \eta \leq 1$. Then, for any $1 \leq i, j \leq 2n$,

$$\|x_k^i - x_k\| = \|s_k^i\| \leq E^k \leq \frac{1}{\eta} e^k \leq \frac{1}{\eta} \|s_k^j\| = \frac{1}{\eta} \|x_k^j - x_k\|.$$

We define the *contour* $C(x_0)$ to be $C(x_0) = \{x : f(x) = f(x_0)\}$. Since $x_0 \in \Omega_\epsilon$, Lemma 7.1 allows us to define $N = \min\{k : x_k \neq x_0\}$. We then define d to be $d = \text{dist}(L(x_N), C(x_0))$. Because $L(x_N)$ and $C(x_0)$ are compact and disjoint, we know that $d > 0$. We now make the following claim:

CLAIM. *Suppose $k \geq N$. If for some $j = 1, \dots, 2n$, $\|x_k^j - x_k\| \leq \eta d$, then $x_k^i \in L(x_0)$ for all $i = 1, \dots, 2n$.*

The proof of this claim is by straightforward application of the triangle inequality [20].

Let

$$\alpha = \min_{x \in \Omega_\epsilon} \|\nabla f(x)\|.$$

By design, $\alpha > 0$. Since ∇f is continuous on $L(x_0)$, ∇f is uniformly continuous on $L(x_0)$. Thus, there exists a constant $r > 0$, depending only on α and the ξ from (14), such that

$$\|\nabla f(x) - \nabla f(x_k)\| \leq \frac{\xi\alpha}{2} \text{ whenever } \|x - x_k\| \leq r \text{ (and } x \in L(x_0)\text{)}.$$

We define

$$(16) \quad \delta' = \min\{\eta d, r\}.$$

Now, assume $k \geq N$ and $x_k \in \Omega_\epsilon$. Choose a trial point x_k^i , $i = 1, \dots, 2n$, that satisfies both $\nabla f(x_k)^T(x_k^i - x_k) < 0$ and

$$\frac{|\nabla f(x_k)^T(x_k^i - x_k)|}{\|\nabla f(x_k)\| \|x_k^i - x_k\|} \geq \xi.$$

The definitions of Ω_ϵ and the pattern P_k , together with Lemma 7.2, guarantee the existence of at least one such x_k^i .

The Mean Value Theorem tells us that $f(x_k^i) - f(x_k) = \nabla f(\omega)^T(x_k^i - x_k)$ for some $\omega \in (x_k, x_k^i)$, whence

$$(17) \quad f(x_k^i) - f(x_k) = \nabla f(x_k)^T(x_k^i - x_k) + (\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k).$$

Consider the first term on the right-hand side of (17). Our choice of x_k^i gives us

$$|\nabla f(x_k)^T(x_k^i - x_k)| \geq \xi \|\nabla f(x_k)\| \|x_k^i - x_k\|.$$

Furthermore, since $\nabla f(x_k)^T(x_k^i - x_k) < 0$, we have

$$(18) \quad \nabla f(x_k)^T(x_k^i - x_k) \leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\|.$$

Now consider the second term on the right-hand side of (17). The Cauchy–Schwarz inequality gives us

$$(19) \quad |(\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k)| \leq \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\|.$$

Combine (18) and (19) to rewrite (17) as

$$\begin{aligned} f(x_k^i) - f(x_k) &\leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\| + \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\| \\ &= (-\xi \|\nabla f(x_k)\| + \|\nabla f(\omega) - \nabla f(x_k)\|) \|x_k^i - x_k\|. \end{aligned}$$

Since $\omega \in (x_k, x_k^i)$, once $\|x_k^i - x_k\| \leq \delta'$,

$$(20) \quad f(x_k^i) - f(x_k) \leq (-\xi \|\nabla f(x_k)\| + \frac{\xi}{2} \|\nabla f(x_k)\|) \|x_k^i - x_k\| < 0.$$

Thus, when $\|x_k^i - x_k\| \leq \delta'$, $f(x_k^i) < f(x_k)$. The Hypotheses on Exploratory Moves guarantee that if $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$. Thus, $\rho_k = f(x_k) - f(x_k + s_k) > 0$ and the algorithms for updating Δ_k (Algorithm 2) ensures that $\Delta_{k+1} \geq \Delta_k$.

We invoke Lemma 3.1 to obtain

$$\delta' \geq \|x_k^i - x_k\| \equiv \|s_k^i\| \geq \zeta_* \Delta_k.$$

Thus, at any iteration $k \geq N$, if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta'/\zeta_*$, then $\Delta_{k+1} \geq \Delta_k$.

We close by noting that since iteration $k = N - 1$ is defined to be the first successful iteration, if we define

$$\delta = \min \left\{ \Delta_{N-1}, \frac{\delta'}{\zeta_*} \right\},$$

we are guaranteed that at any iteration k , if $x_k \in \Omega_\epsilon$ and $\delta \geq \Delta_k$, then $\Delta_{k+1} \geq \Delta_k$. \square

Proposition 7.4 guarantees that if Δ_k is small enough, a generalized pattern search method will realize simple decrease because there exists at least one step among the $2n$ steps defined by $\Delta_k B\Gamma_k$ that gives decrease as a function of the norm of the gradient at the current iterate, as shown in (20); the Hypotheses on Exploratory Moves then ensure that the exploratory moves algorithm must return a step that satisfies at least simple decrease. However, there are no guarantees that the step returned by an exploratory moves algorithm satisfies more than the simple decrease condition.

To tie the amount of actual decrease to the norm of the gradient, we must place much stronger conditions on the generalized pattern search method, as discussed in §3.3.2. Once we have done so, Corollary 7.5 follows more or less immediately from Proposition 7.4.

COROLLARY 7.5. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on $L(x_0)$. Suppose that the columns of the generating matrix are bounded in norm and that the generalized pattern search method (Algorithm 1) enforces the Strong Hypotheses on Exploratory Moves. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exist $\delta' > 0$ and $\sigma > 0$, independent of k , such that for all but finitely many k , if $x_k \in \Omega_\epsilon$ and $\Delta_k \leq \delta'$, then

$$f(x_{k+1}) \leq f(x_k) - \sigma \|\nabla f(x_k)\| \|s_k\| < f(x_k).$$

Proof. From Proposition 7.4, (20) says that for $k \geq N = \min\{k : x_k \neq x_0\}$ (Lemma 7.1 guarantees that $N < +\infty$), there exists at least one trial step $s_k^i \in \Delta_k B\Gamma_k$ such that once $\|s_k^i\| < \delta'$, where δ' is as defined in (16), we have

$$f(x_k^i) \leq f(x_k) - \frac{\xi}{2} \|\nabla f(x_k)\| \|s_k^i\| < f(x_k).$$

The Strong Hypotheses on the Exploratory Moves give us

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \|\nabla f(x_k)\| \|s_k^i\| < f(x_k).$$

Lemma 3.1 ensures that

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \zeta_* \Delta_k \|\nabla f(x_k)\| < f(x_k).$$

Lemma 3.6, which holds only when the columns of the generating matrix are bounded in norm, gives us

$$f(x_{k+1}) \leq f(x_k) - \frac{\xi}{2} \zeta_* \psi_* \|\nabla f(x_k)\| \|s_k\| < f(x_k).$$

We define $\sigma = \frac{\xi}{2} \zeta_* \psi_*$ to complete the proof. \square

We now prove Proposition 3.4.

Proof. (Proposition 3.4.) By assumption, $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$. Then we can find N_1 and $\epsilon > 0$ such that for all $k \geq N_1$, $x_k \in \Omega_\epsilon = \{x \in L(x_0) : \text{dist}(x, X_*) \geq \epsilon\}$. Lemma 7.1 guarantees the existence of $N_2 = \min\{k : x_k \neq x_0\}$. Let $N = \max(N_1, N_2)$.

From Proposition 7.4 we are assured of $\delta > 0$ such that if $\Delta_k \leq \delta$, then the iteration will be successful. Given Δ_0 , there exists a constant $p \in \mathbf{Z}$, $p \geq 0$, such that $\theta^p \Delta_0 \leq \delta$. Thus, for $k \geq N$, $\theta^{p+1} \Delta_0 < \Delta_k$.

Set $\Delta_{LB} = \theta \min(\theta^p \Delta_0, \Delta_1, \dots, \Delta_{N-1})$. Then for all k , $\Delta_{LB} < \Delta_k$. \square

Acknowledgments. This paper benefited from conversations with J. E. Dennis, Michael Trosset, Luís Vicente, and especially Robert Michael Lewis. In particular, discussions with Michael Lewis were critical in the distillation of the abstraction for pattern search methods found in §2 and in the development of the analytic arguments for the algebraic structure of the iterates found in §3.1.

The review and comments made by two anonymous referees and Jorge Moré, the editor, are gratefully acknowledged. In particular, the observation by one of the referees that the pattern contains only vectors in a fixed lattice led to a more general result and a much more elegant presentation.

REFERENCES

- [1] M. AVRIEL, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [2] G. E. P. BOX, *The exploration and exploitation of response surfaces: Some general considerations and examples*, *Biometrics*, 10 (1954), pp. 16–60.
- [3] G. E. P. BOX, *Evolutionary operation: A method for increasing industrial productivity*, *Appl. Statist.*, 6 (1957), pp. 81–101.
- [4] G. E. P. BOX AND K. B. WILSON, *On the experimental attainment of optimum conditions*, *Journal of the Royal Statistical Society, Series B*, XIII (1951), pp. 1–45.
- [5] M. J. BOX, D. DAVIES, AND W. H. SWANN, *Non-Linear Optimization Techniques*, ICI Monograph No. 5, Oliver & Boyd, Edinburgh, 1969.
- [6] J. CÉA, *Optimisation : théorie et algorithmes*, Dunod, Paris, 1971.
- [7] W. C. DAVIDON, *Variable metric method for minimization*, *SIAM J. Optimization*, 1 (1991), pp. 1–17. The belated preface was received by the editors June 4, 1990; accepted for publication August 10, 1990. The rest of the article was originally published as Argonne National Laboratory Research and Development Report 5990, May 1959 (revised November 1959).
- [8] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [9] J. E. DENNIS, JR. AND V. TORCZON, *Direct search methods on parallel machines*, *SIAM J. Optimization*, 1 (1991), pp. 448–474.
- [10] R. HOOKE AND T. A. JEEVES, “Direct search” solution of numerical and statistical problems, *J. Assoc. Comput. Mach.*, 8 (1961), pp. 212–229.
- [11] J. J. MORÉ, *Recent developments in algorithms and software for trust region methods*, in *Mathematical Programming, The State of the Art*, A. Bachem, M. Grottschel, and G. Korte, eds., Springer-Verlag, 1983, pp. 256–287.
- [12] J. A. NELDER AND R. MEAD, *A simplex method for function minimization*, *Comput. J.*, 7 (1965), pp. 308–313.
- [13] J. NOCEDAL, *Theory of algorithms for unconstrained optimization*, *Acta Numerica*, 1 (1992), pp. 199–242.
- [14] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several*

- Variables*, Academic Press, New York, 1970.
- [15] M. J. D. POWELL, *Convergence properties of a class of minimization algorithms*, in *Nonlinear Programming 2*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, 1975, pp. 1–27.
 - [16] W. SPENDLEY, G. R. HEXT, AND F. R. HIMSWORTH, *Sequential application of simplex designs in optimisation and evolutionary operation*, *Technometrics*, 4 (1962), pp. 441–461.
 - [17] W. H. SWANN, *Direct search methods*, in *Numerical Methods for Unconstrained Optimizations*, W. Murray, ed., Academic Press, London and New York, 1972, pp. 13–28.
 - [18] S. W. THOMAS, *Sequential estimation techniques for quasi-Newton algorithms*, Ph.D. thesis, Cornell University, Ithaca, NY, 1975.
 - [19] V. TORCZON, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1989; also available as Tech. Report 90–7, Department of Mathematical Sciences, Rice University, Houston, TX 77251–1892.
 - [20] ———, *On the convergence of the multidirectional search algorithm*, *SIAM J. Optimization*, 1 (1991), pp. 123–145.
 - [21] ———, *PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines*, Tech. Report 92–9, Department of Mathematical Sciences, Rice University, Houston, TX 77251–1892, 1992.
 - [22] Y. WEN-CI, *Positive basis and a class of direct search techniques*, *Scientia Sinica, Special Issue of Mathematics*, 1 (1979), pp. 53–67.