On the Convergence of Pattern Search Algorithms

Virginia Torczon

CRPC-TR93322
June 1993

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Also released as CAM TR93-10, June 1993.

ON THE CONVERGENCE OF PATTERN SEARCH ALGORITHMS[*]

VIRGINIA TORCZON[†]

Abstract. This paper gives a simplified, abstract description of generalized 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. The abstract description of pattern search methods is used to establish a global first-order stationary point convergence theory that neither requires the directional derivative nor enforces a notion of sufficient decrease. The relationship between the convergence theory for pattern search methods and the theory for both line search and model trust region strategies is also discussed.

Key words. unconstrained optimization, convergence analysis, direct search methods, model trust region methods, line search methods, 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 will study methods that do not require the directional derivative to solve the unconstrained minimization problem

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

where $f : \mathbf{R}^n \to \mathbf{R}$. 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 will give a global, first-order stationary point convergence theory for pattern search methods, which, to our knowledge, will provide the first known convergence result for some of these methods, and the first general convergence theory for all of them.

Hooke and Jeeves [10] introduced the term "direct search" to describe methods that work directly with values of the objective function to drive the search. In particular, they noted that:

> Direct search is distinguished from other numerical procedures by having a finite number of states which, without loss of generality, can be indexed by a set of integers.... It does not include methods which possess a continuum of states, such as those (e.g., Newton's method and methods of steepest ascent) which rely on the use of such tools as derivatives and power series approximations. (p. 225)

For Hooke and Jeeves, the notion of a "finite number of states" was no doubt influenced by automata theory:

[*] 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 "strategy" for selecting new trial points is determined by a set of "states" which provide the memory. The number of states is finite. There is an arbitrary initial state S_0, and a final state which stops the search. The other states represent various conditions which arise as a function of the results of the trials made. (p. 213)

Our definition and analysis of pattern search methods will hinge on a different notion of "finite number of states." At iteration k of a pattern search method, the iterate $\vec{x}_{k+1}$ is chosen from a finite set of possibilities that is essentially independent of the function f. A successful iteration is one for which the algorithm chooses from among this finite set of possibilities any iterate that satisfies *simple decrease*, i.e., $f(\vec{x}_{k+1}) < f(\vec{x}_k)$.

A unique feature of the convergence theory we will present is that we are able to guarantee first-order 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. To do so, we specifically exploit the fact that pattern search methods possess a finite number of states. 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.

The important ideas for the convergence theory for pattern search methods come from the convergence theory 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.

We have been aware, for some time, 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, *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, then, was to show that these methods were special cases of a generalized pattern search method.

We flirt with the possibility of confusion by using *pattern search* to describe this class of methods, since Hooke and Jeeves first used the term to describe the particular direct search method they developed. Our reasons for adopting this terminology are several. First, the notion of a pattern that is used to define the search at every inner iteration of these methods aptly captures the common element that allows for the general convergence theory. But just as important is our desire to give credit to Hooke and Jeeves for recognizing the usefulness of direct search methods. These algorithms have remained in favor with users, not because they are particularly efficient when compared to methods that rely on derivatives, but because they are both simple and robust. The convergence theory 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 theory 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 theory 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 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 and thus fall under the domain of the convergence theory we have developed. In §4, we will prove that if the function to be minimized is continuously differentiable, then these methods guarantee first-order stationary point convergence. In §5 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. Finally, in §6, we give some concluding remarks.

2. The Generalized Pattern Search Method.

To make the convergence analysis succinct, we will introduce the following abstraction of pattern search methods. We will defer to the next section demonstrations that the pattern search methods mentioned above fall comfortably within this abstraction. Bear in mind that none of the pattern search methods we will discuss was developed with a convergence theory in mind. The seemingly complicated features found in the abstract specification reflect the original statements of each these algorithms and so were motivated by practical considerations.

The Basis Matrices. We begin with a finite, nonempty set of $n \times n$ nonsingular *basis matrices* $\mathbf{B}$. We choose a particular matrix

(1) $$B_\nu = \left[\vec{b}_\nu^1 \cdots \vec{b}_\nu^n\right] \in \mathbf{B}$$

as a reference and express the columns of any other matrix

$$B_\psi = \left[\vec{b}_\psi^1 \cdots \vec{b}_\psi^n\right] \in \mathbf{B}$$

as a linear combination of the columns of the designated matrix B_ν. Thus,

(2) $$\vec{b}_\psi^j = \sum_{l=1}^{n} \beta_{\psi,l}^j \vec{b}_\nu^l \qquad j = 1, \cdots, n.$$

The Core Pattern. We require a *core pattern* $\Gamma = [\vec{\gamma}^1 \cdots \vec{\gamma}^n] \in \mathbf{R}^{n \times n}$. The core pattern Γ must also be nonsingular. We will refer to the elements γ_m^j of Γ; the superscript refers to the column and the subscript to the row.

The Generating Matrix. At every iteration k we require an $n \times p$ *generating matrix* C_k, where $p > 2n$. We use C_k to generate a pattern of points associated with a given algorithm. We require that the columns of C_k contain the columns of both the core pattern Γ and its negative $-\Gamma$. In addition, we will require C_k to contain a column of all zeros. We will partition C_k as follows:

(3) $$C_k = [\, \underbrace{\Gamma}_{n} \; \underbrace{-\Gamma}_{n} \; \underbrace{A_k}_{p-2n} \,] = [\, \underbrace{F}_{2n} \; \underbrace{A_k}_{p-2n} \,].$$

The block $F = [\Gamma \; -\Gamma]$ is a portion of the generating matrix that is *fixed* across all iterations; A_k contains *additional* columns. The columns of A_k may or may not vary across iterations; however, A_k always contains at least one column—the column of all zeros. We express the columns of

$$C_k = \left[\vec{c}_k^{\,1} \cdots \vec{c}_k^{\,p}\right]$$

as linear combinations of the columns of the core pattern Γ so that

(4) $$\vec{c}_k^{\,i} = \sum_{j=1}^{n} \alpha_{k,j}^{i} \vec{\gamma}^{\,j} \qquad i = 1, \cdots, p.$$

The Constants of Proportionality. To ensure global convergence, we must place a restriction on the scalars $\beta_{\psi,l}^{j}$, γ_m^j, and $\alpha_{k,j}^{i}$; namely, that there exist nonzero scalars $q_1, q_2 \in \mathbf{R}$ such that

(5) $$q_1 \beta_{\psi,l}^{j} \in \mathbf{Z}, \quad \forall\, j = 1, \cdots, n \quad \forall\, l = 1, \cdots, n \quad \forall\, \psi = 1, \cdots, |\mathbf{B}|,$$

where $\mathbf{Z}$ denotes the set of integers and $|\mathbf{B}|$ denotes the cardinality of the set $\mathbf{B}$,

(6) $$q_2 \gamma_m^j \in \mathbf{Z}, \quad \forall\, j = 1, \cdots, n \quad \forall\, m = 1, \cdots, n,$$

and

(7) $$q_2 \alpha_{k,j}^{i} \in \mathbf{Z} \quad \forall\, i = 1, \cdots, p \quad \forall\, j = 1, \cdots, n \quad \forall\, k.$$

The constants of proportionality q_1 and q_2 found in (5), (6), and (7) are present in each of the pattern search methods we will consider.

The Pattern. A *pattern* P_k is defined by the columns of the matrix $P_k = B_k C_k$, where $B_k \in \mathbf{B}$. Because both B_k and C_k have rank n, the columns of P_k span $\mathbf{R}^n$. For convenience, we will use the partition of the generating matrix C_k given in (3) to partition P_k as follows:

(8) $$P_k = B_k C_k = [\, \underbrace{B_k F}_{2n} \; \underbrace{B_k A_k}_{p-2n} \,].$$

We define a *trial step* $\vec{s}_k^{\,i}$ to be any vector of the form

(9) $$\vec{s}_k^{\,i} = \Delta_k B_k \vec{c}_k^{\,i},$$

where $\Delta_k \in \mathbf{R}$. Note that $B_k \vec{c}_k^{\,i}$ 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

$$\vec{x}_k^{\,i} = \vec{x}_k + \vec{s}_k^{\,i},$$

where $\vec{x}_k$ 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 $\vec{x}_k$ in a well-defined deterministic fashion in search of a new iterate $\vec{x}_{k+1} = \vec{x}_k + \vec{s}_k$ 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 shall 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 $\vec{y} \in A$ will mean the vector $\vec{y}$ is contained in the set of columns of A.)

Hypotheses on Exploratory Moves.

1. $\vec{s}_k \in \Delta_k P_k \equiv \Delta_k B_k C_k$.
2. If $\min\{f(\vec{x}_k + \vec{\sigma}),\ \vec{\sigma} \in \Delta_k B_k F\} < f(\vec{x}_k)$, then $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$.

In other words, the choice of exploratory moves must ensure two things:

1. The direction of any step $\vec{s}_k$ accepted at iteration k is defined by the pattern P_k and its length is determined by Δ_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 $\Delta_k B_k F$, then the exploratory moves will produce a step $\vec{s}_k$ that also gives simple decrease on the function value at the current iterate.

   There are two important points worth noting:
   - First, $\vec{s}_k$ may be contained in $\Delta_k B_k A_k$ rather than in $\Delta_k B_k F$.
   - Second, $f(\vec{x}_k + \vec{s}_k)$ need not be $\min\{f(\vec{x}_k + \vec{\sigma}),\ \vec{\sigma} \in \Delta_k B_k F\}$; $\vec{s}_k$ only need satisfy $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$.

These are the properties of the exploratory moves that enable us to prove 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.

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

To define a pattern search method, it is necessary to specify the family of basis matrices $\mathbf{B}$, the core pattern Γ, the generating matrix $C_k = [\Gamma\ -\Gamma\ A_k] = [F\ A_k]$, the

```
Algorithm 1. The Generalized Pattern Search Method.
Let $\vec{x}_0 \in \mathbf{R}^n$ and $\Delta_0 > 0$ be given.
For $k = 0, 1, \cdots$,
    a) Compute $f(\vec{x}_k)$.
    b) Determine a step $\vec{s}_k$ using an exploratory moves algorithm.
    c) Compute $\rho_k = f(\vec{x}_k) - f(\vec{x}_k + \vec{s}_k)$.
    d) If $\rho_k > 0$ then $\vec{x}_{k+1} = \vec{x}_k + \vec{s}_k$. Otherwise $\vec{x}_{k+1} = \vec{x}_k$.
    e) Update $C_k$, $B_k$, and $\Delta_k$.
```

result $\vec{s}_k$ of the exploratory moves, and the algorithms for choosing B_k from $\mathbf{B}$ and 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 $\Delta_k$.
Given $\tau \in \mathbf{Q}$ ($\mathbf{Q}$ denotes the set of rational numbers), $\tau > 1$ and $\{w_0, w_1, \cdots, w_{|\Lambda|}\} \subset \mathbf{Z}$, $|\Lambda| < +\infty$, where $w_0 < 0$ and $w_i \geq 0$, $i = 1, \cdots, |\Lambda|$, let $\theta = \tau^{w_0}$ and $\lambda_k \in \Lambda = \{\tau^{w_1}, \cdots, \tau^{w_{|\Lambda|}}\}$.
    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 theory. Also crucial to the theory is the relationship (overlooked in [20]) between θ and the elements of Λ. These requirements are not particularly restrictive, as we shall see in §3.

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 the conditions given in (4) and (7).

Updating B_k requires a way to choose B_{k+1} from the finite family of basis matrices $\mathbf{B}$. Again, this choice will depend on the pattern search method, but will not affect the theory. For theoretical purposes it is sufficient that the matrices contained in $\mathbf{B}$ satisfy the conditions given in (2) and (5).

3. The Particular Pattern Search Methods.

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

- the coordinate search algorithm,
- response surface methodology, in its simplest form, as presented by Box and Wilson in [4],
- 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 methods satisfy the conditions that define pattern search methods and thus are special cases of the generalized pattern search method presented as Algorithm 1. Specifically, we will identify the following for each of the four methods:

1. The finite, nonempty set of $n \times n$ nonsingular basis matrices $\mathbf{B}$.
2. The nonsingular core pattern $\Gamma \in \mathbf{R}^{n \times n}$.
3. The $n \times p$ generating matrix C_k that includes the columns of both Γ and $-\Gamma$, as well as a column of zeros, among its p columns so that C_k can be partitioned as in (3).
4. The constants of proportionality $q_1, q_2 \in \mathbf{R}$ that satisfy (5), (6), and (7).
5. An exploratory moves algorithm that satisfies the conditions given in Hypotheses on Exploratory Moves.
6. An algorithm for choosing B_{k+1} from $\mathbf{B}$.
7. An algorithm for updating C_k.
8. An algorithm for updating Δ_k that conforms with Algorithm 2.

These are the pieces we need to specify in the generalized pattern search method given in Algorithm 1. Once we have established that these conditions hold for a given method, we can appeal to Theorem 4.4 to claim global convergence for the method.

There are undoubtedly other algorithms for which this 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, as illustrative of the power of Theorem 4.4.

3.1. 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....

Not only is this algorithm simple, it conforms with classical notions of good experimental design: vary one factor at a time and observe the effect of that variation on the result of the associated experiment. This 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 and, in fact, could be considered the canonical pattern search algorithm. To see this, we will begin by considering all possible scenarios for a single iteration of coordinate search when $n = 2$. These can be seen in Fig. 1. We mark the current iterate $\vec{x}_k$. The $\vec{x}_k$'s with superscripts mark *trial points* considered during the course of the iteration.

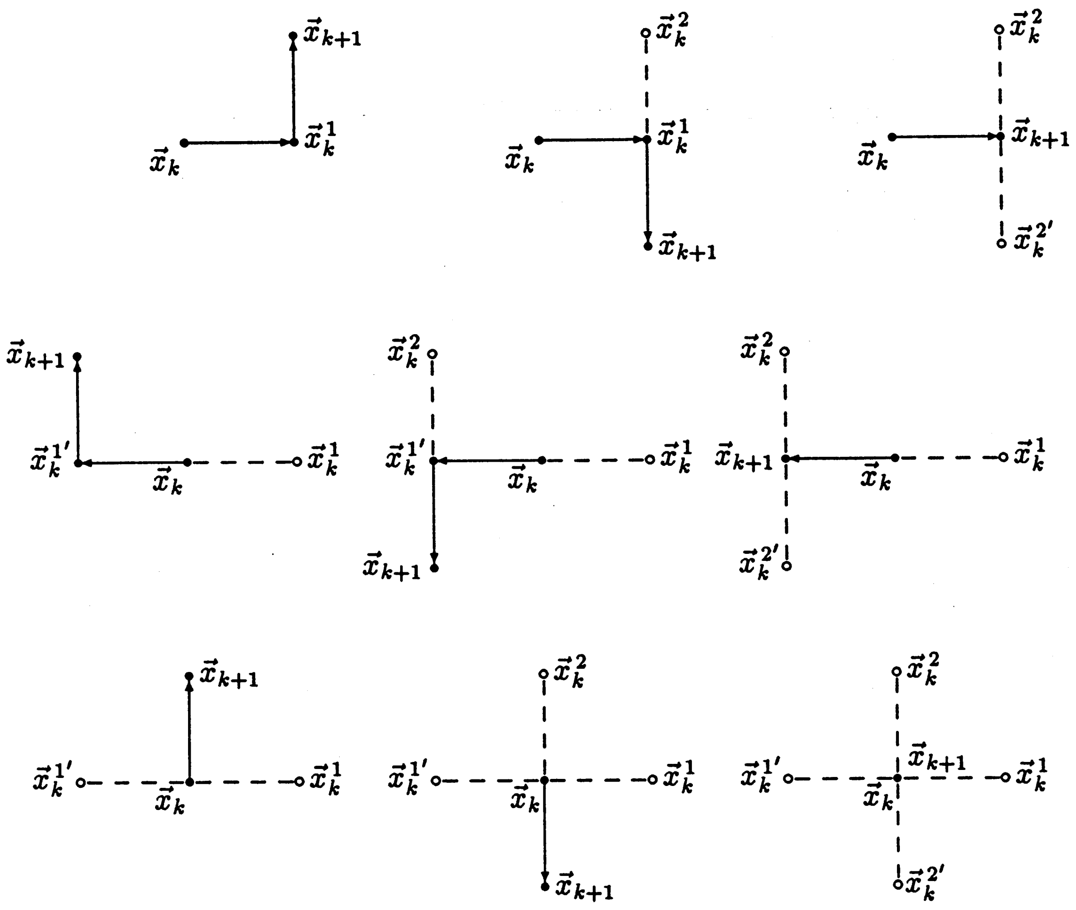

FIG. 1. *All possible subsets of the steps for coordinate search in* $\mathbf{R}^2$.

The next iterate $\vec{x}_{k+1}$ 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 $\vec{x}_k$ to $\vec{x}_k^1$ resulted in a decrease in the objective function, so the step from $\vec{x}_k^1$ to $\vec{x}_{k+1}$ was tried and led to a further decrease in the objective function value. The iteration was then terminated with a new point $\vec{x}_{k+1}$ that satisfies the simple decrease condition $f(\vec{x}_{k+1}) < f(\vec{x}_k)$. In the worst case, the last scenario shown, $2n$ trial points were evaluated ($\vec{x}_k^1$, $\vec{x}_k^{1'}$, $\vec{x}_k^2$, and $\vec{x}_k^{2'}$) without producing decrease in the function value at the current iterate $\vec{x}_k$. In this case, $\vec{x}_{k+1} = \vec{x}_k$ and the step size must be reduced for the next iteration.

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

3.1.1. The Basis Matrix. The family of basis matrices $\mathbf{B}$ consists of a single matrix. Coordinate search is usually defined so that the basis matrix is the identity matrix; i.e., $\mathbf{B} = \{I\}$. 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).

3.1.2. The Core Pattern. For coordinate search, the core pattern is the identity matrix; i.e., $\Gamma = I$.

3.1.3. The Generating Matrix. For coordinate search, the generating matrix C_k contains in its columns all possible combinations of $\{-1, 0, 1\}$. Thus, C_k has $p = 3^n$ columns. In particular, the columns of C_k will contain both I and $-I$, as well as a column of all zeros. Thus $F = [\Gamma \ -\Gamma] = [I \ -I]$, which is consistent with our definition of the core pattern. Note, also, that C_k is fixed across all iterations of the method; A_k consists of the remaining $3^n - 2n$ columns of C_k. For $n = 2$ we have

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

Thus, when $n = 2$, all possible trial points defined by the pattern $P_k = B_k C_k$, 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.

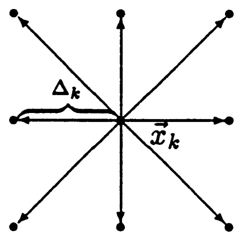

FIG. 2. *The pattern for coordinate search in* $\mathbf{R}^2$ *with a given step length* Δ_k.

3.1.4. The Constants of Proportionality. Since **B** consists of a single matrix, $q_1 = 1$.

Since the generating matrix C_k consists of all possible combinations of $\{-1, 0, 1\}$ (and is fixed across all iterations of the method) and $\Gamma = I$, the second constant of proportionality is simply $q_2 = 1$.

3.1.5. The Exploratory Moves. The exploratory moves for coordinate search are given in Algorithm 3, where the $\vec{e}_i$'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.

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 $\Delta_k P_k$.

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

Algorithm 3. Exploratory Moves Algorithm for Coordinate Search.

```
Given $\vec{x}_k$, $\Delta_k$, $f(\vec{x}_k)$, and $B_k$, set $\vec{s}_k = \vec{0}$, $\rho_k = 0$, and $min = f(\vec{x}_k)$.
For $i = 1, \cdots, n$ do
    a) $\vec{s}_k^{\,i} = \vec{s}_k + \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^{\,i} = \vec{x}_k + \vec{s}_k^{\,i}$. Compute $f(\vec{x}_k^{\,i})$.
    b) If $f(\vec{x}_k^{\,i}) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^{\,i})$, $min = f(\vec{x}_k^{\,i})$, and $\vec{s}_k = \vec{s}_k^{\,i}$.
       Otherwise,
         i) $\vec{s}_k^{\,i} = \vec{s}_k - \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^{\,i} = \vec{x}_k + \vec{s}_k^{\,i}$. Compute $f(\vec{x}_k^{\,i})$.
         ii) If $f(\vec{x}_k^{\,i}) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^{\,i})$, $min = f(\vec{x}_k^{\,i})$, and $\vec{s}_k = \vec{s}_k^{\,i}$.
Return.
```

scenario shown in Fig. 1, decrease in the objective function was realized for the first trial step

$$\vec{s}_k^{\,1} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

so the second trial step

$$\vec{s}_k^{\,2} = \Delta_k I \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

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

$$\vec{s}_k^{\,\prime} = \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which is contained in the core pattern $\Gamma = I$ (the step $\vec{s}_k^{\,2}$ is not contained in the core pattern), but $\vec{s}_k^{\,\prime}$ is not tried when simple decrease is realized by the step $\vec{s}_k^{\,1}$. However, *in the worst case*, as seen in Fig. 1, the algorithm for coordinate search ensures that all $2n$ steps defined by $\Delta_k B_k F = \Delta_k B_k[\Gamma \; -\Gamma] = \Delta_k B_k[I \; -I]$ are tried before returning the step $\vec{s}_k = \vec{0}$. In other words, the exploratory moves given in Algorithm 3 will examine all $2n$ steps defined by $\Delta_k B_k F$ *unless* a step satisfying $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$ is found.

3.1.6. Choosing the Basis Matrix. Since **B** consists of a single matrix, there is no need for a special algorithm to select B_k.

3.1.7. Updating the Generating Matrix. Since C_k is fixed across all iterations of the method, there is no need for an update algorithm.

3.1.8. 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 $w_1 = 0$.

Since we have shown that coordinate search with fixed step length satisfies all the necessary requirements, we can therefore conclude that coordinate search with fixed step length is a pattern search method.

3.2. 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 simplest form, response surface methodology is based on what are known as "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].) An example for $n = 2$ can be seen in Fig. 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].

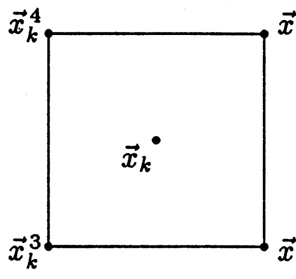

FIG. 3. *The full factorial design in* $\mathbf{R}^2$.

This simple idea is, in some sense, the dual of coordinate search: rather than vary one factor at a time and observe the effect of that single variation on the result of the associated experiment, vary all the factors simultaneously and observe the effect. This more closely conforms with statistical notions of good experimental design since the intent is to better capture interactions between factors. There are many variations on this simple theme, including several in the original paper of Box and Wilson. We will concentrate on this basic version and show that it is a pattern search algorithm. However, many of the proposed variants, such as fractional factorial designs or composite designs, can also be formulated so that they fall within the definition we have given for pattern search methods.

3.2.1. The Basis Matrix. As with coordinate search, the family of basis matrices for response surface methodology consists of a single matrix. The usual choice is $\mathbf{B} = \{I\}$, though, as with coordinate search, other choices may be made to reflect information known about the problem to be solved.

3.2.2. The Core Pattern. For response surface methodology, the core pattern consists of any linearly independent subset of n columns consisting of combinations of $\{-1, 1\}$. This choice will become clearer when we define the generating matrix, but as an illustration we will show a core pattern for the example in Fig. 3:

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

3.2.3. The Generating Matrix. The generating matrix C_k for response surface methodology contains in its columns all possible combinations of $\{-1, 1\}$; to this we append the column of all zeros. Thus C_k has $p = 2^n + 1$ columns. For $n = 2$,

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

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

For $n = 2$ and a given step length Δ_k, all possible trial points defined by the pattern P_k can be seen in Fig. 4.

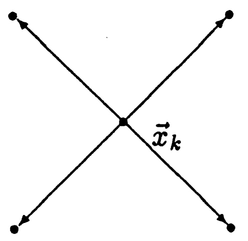

FIG. 4. *The pattern for response surface methodology in* $\mathbf{R}^2$ *with a given step length* Δ_k.

3.2.4. The Constants of Proportionality. Once again, there is a single basis matrix, so $q_1 = 1$.

The columns of A_k consist of integer combinations of the columns of Γ, so $q_2 = 1$.

3.2.5. The Exploratory Moves. The exploratory moves given in Algorithm 4 are simultaneous in the sense that every possible trial step $\vec{s}_k^{\,i} \in \Delta_k P_k = \Delta_k B_k C_k$ is computed at each iteration. It is then the case that every trial step $\vec{s}_k^{\,i}$ is contained in $\Delta_k P_k$. The second observation of note is that since

$$\vec{s}_k = \underset{\vec{s}_k^{\,i} \in \Delta_k P_k}{\arg\min} \{ f(\vec{x}_k + \vec{s}_k^{\,i}) \},$$

then, if $\min\{f(\vec{x}_k + \vec{\sigma}), \vec{\sigma} \in \Delta_k F\} < f(\vec{x}_k)$, we have $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$—regardless of our choice of Γ (and thus, by extension, our choice of F).

3.2.6. Choosing the Basis Matrix. Since **B** consists of a single matrix, there is no need for a special algorithm to choose B_k.

3.2.7. Updating the Generating Matrix. The generating matrix is fixed across all iterations of the method, so there is no need for an algorithm to update C_k.

Algorithm 4. Exploratory Moves Algorithm for Response Surface Methodology.

```
Given $\vec{x}_k$, $\Delta_k$, $f(\vec{x}_k)$, $B_k$, and $C_k = [\vec{c}_k^1 \cdots \vec{c}_k^p]$, set $\vec{s}_k = \vec{0}$, $\rho_k = 0$, and $min = f(\vec{x}_k)$.
For $i = 1, \cdots, 2^n$ do
    a) $\vec{s}_k^i = \Delta_k B_k \vec{c}_k^i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
    b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, and $\vec{s}_k = \vec{s}_k^i$.
Return.
```

3.2.8. 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 $\Lambda = \{1\}$, so that $\tau = 2$, $w_0 = -1$, and $w_1 = 0$.

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.

3.3. Hooke and Jeeves' Pattern Search Algorithm.

In addition to introducing the general notion of a direct search method, Hooke and Jeeves also 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 $\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})$, rather than about the current iterate $\vec{x}_k$. This is the pattern step. The idea is to investigate whether further progress is possible in the general direction $\vec{x}_k - \vec{x}_{k-1}$ (since, if $\vec{x}_k \neq \vec{x}_{k-1}$, then $\vec{x}_k - \vec{x}_{k-1}$ is clearly a promising direction).

To make this a little clearer, we consider the example shown in Fig. 5 for $n = 2$. Given $\vec{x}_{k-1}$ and $\vec{x}_k$ (we assume, for now, that $k > 0$ and that $\vec{x}_k \neq \vec{x}_{k-1}$), the pattern

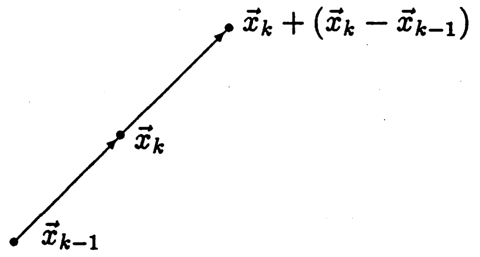

FIG. 5. *The pattern step in* $\mathbf{R}^2$, *given* $\vec{x}_k \neq \vec{x}_{k-1}$, $k > 0$.

search algorithm takes the step $\vec{x}_k - \vec{x}_{k-1}$ from $\vec{x}_k$. The function is evaluated at this trial step and it is accepted, temporarily, *even if* $f(\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})) \geq f(\vec{x}_k)$. The Hooke and Jeeves pattern search algorithm then proceeds to conduct coordinate search about the temporary iterate $\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})$. Thus, in $\mathbf{R}^2$, the exploratory moves are exactly as shown in Fig. 1, but with $\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})$ substituted for $\vec{x}_k$.

If coordinate search about the temporary iterate $\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})$ is successful, then the point returned by coordinate search about the temporary iterate is accepted as the new iterate $\vec{x}_{k+1}$. If not, i.e., $f((\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})) + \vec{s}_k) \geq f(\vec{x}_k)$, then the pattern step is deemed unsuccessful, and the method reduces to coordinate search about $\vec{x}_k$. For our example, then, the exploratory moves would then simply resort to the possibilities shown in Fig. 1.

If the previous iteration was not successful, so $\vec{x}_k = \vec{x}_{k-1}$ and $(\vec{x}_k - \vec{x}_{k-1}) = \vec{0}$, then the iteration is limited to coordinate search about $\vec{x}_k$. 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 will use all the machinery required for coordinate search, but append to the generating matrix another set of 3^n columns to capture the effect of the pattern step. We will make this point clearer in the discussion that follows.

3.3.1. The Basis Matrix. Once again, the family of basis matrices $\mathbf{B}$ consists of a single matrix that is usually chosen to be the identity.

3.3.2. The Core Pattern. As for coordinate search, the core pattern is the identity matrix.

3.3.3. 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 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 $\vec{x}_k$, rather than with respect to the temporary iterate $\vec{x}_k + (\vec{x}_k - \vec{x}_{k-1})$, which is usually how the Hooke and Jeeves pattern search method is described. Thus, for $n = 2$,

$$C_0 = \left[\begin{array}{rrrrrrrrrrrrrrrrrr} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{array}\right]. \tag{10}$$

For notational convenience, we require that the last column of C_0, which we denote as $\vec{c}_0^{\,p}$, be the column of all zeros. In both the algorithm for the exploratory moves (Algorithm 5) and the algorithm for updating C_k (Algorithm 6), we use the column $\vec{c}_k^{\,p}$ to measure the accumulation of a sequence of successful pattern steps. This can be seen, in (11), for our example from Fig. 5. In this example, we have the generating matrix

$$C_k = \left[\begin{array}{rrrrrrrrrrrrrrrrrr} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 2 & 1 & 0 & 1 & 2 & 2 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 & 1 & 2 & 1 & 0 & 2 & 0 & 0 & 2 & 1 \end{array}\right]. \tag{11}$$

The pattern step $(\vec{x}_k - \vec{x}_{k-1})$ 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 (10) and the columns of C_k given in (11) is that $(1\ 1)^T$ has been added to the last 3^2 columns of C_k.

In $\mathbf{R}^2$, with the initial generating matrix C_0, the pattern of points would be identical to that given in Fig. 2 for coordinate search. However, for the example shown in Fig. 5, with the generating matrix given in (11), the pattern would be as shown in Fig. 6.

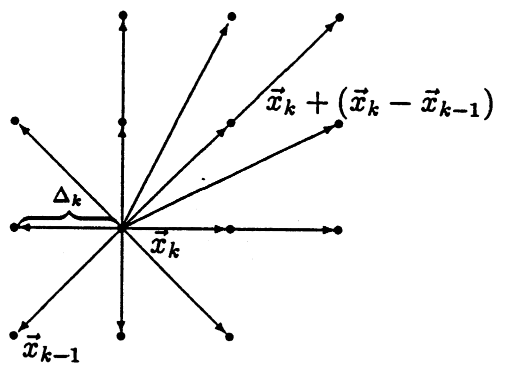

FIG. 6. *The pattern for Hooke and Jeeves in* $\mathbf{R}^2$ *with given step length* Δ_k *and generating matrix* C_k.

3.3.4. The Constants of Proportionality. Since **B** still consists of a single matrix, it is still the case that $q_1 = 1$.

It will also be the case that $q_2 = 1$, as for coordinate search, but that argument can be made only after the discussion in §3.3.7 of the algorithm for updating C_k.

3.3.5. The Exploratory Moves. In Algorithm 5, the $\vec{e}_i$'s denote the unit coordinate vectors and $\vec{c}_k^p$ denotes the last column of C_k, as explained in §3.3.3. 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 5. Exploratory Moves Algorithm for Hooke and Jeeves.

```
Given $\vec{x}_k$, $\Delta_k$, $f(\vec{x}_k)$, $B_k$, and $\rho_{k-1}$, set $\rho_k = \rho_{k-1}$ and $min = f(\vec{x}_k)$.
If $\rho_k > 0$ then set $\vec{s}_k = \Delta_k B_k \vec{c}_k^p$, $\rho_k = f(\vec{x}_k) - f(\vec{x}_k + \vec{s}_k)$, and $min = f(\vec{x}_k + \vec{s}_k)$.
   For $i = 1, \cdots, n$ do
      a) $\vec{s}_k^i = \vec{s}_k + \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
      b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, and $\vec{s}_k = \vec{s}_k^i$.
         Otherwise,
            i) $\vec{s}_k^i = \vec{s}_k - \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
            ii) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, and $\vec{s}_k = \vec{s}_k^i$.
If $\rho_k \le 0$ then set $\vec{s}_k = \vec{0}$, $\rho_k = 0$, and $min = f(\vec{x}_k)$.
   For $i = 1, \cdots, n$ do
      a) $\vec{s}_k^i = \vec{s}_k + \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
      b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, and $\vec{s}_k = \vec{s}_k^i$.
         Otherwise,
            i) $\vec{s}_k^i = \vec{s}_k - \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
            ii) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, and $\vec{s}_k = \vec{s}_k^i$.
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 5 will examine all $2n$ steps defined by $\Delta_k B_k F$ *unless* a step satisfying $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$ is found.

3.3.6. Choosing the Basis Matrix. Since $\mathbf{B}$ consists of a single matrix, there is no need for a special algorithm to select B_k.

3.3.7. Updating the Generating Matrix. 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 6. Note the distinguished role of $\vec{c}_k^{\,p}$, the last column of C_k, which represents the pattern step $(\vec{x}_k - \vec{x}_{k-1})$.

Algorithm 6. Updating C_k.
```
For $i = 3^n + 1, \cdots, 2 \cdot 3^n$ do
        $\vec{c}_{k+1}^{\,i} = \vec{c}_k^{\,i} + (1/\Delta_k)\vec{s}_k - \vec{c}_k^{\,p}$.
Return.
```

Since $(1/\Delta_k)\vec{s}_k$ is necessarily a column of C_k, an argument by induction shows that the update for C_k ensures that the columns of C_k always consists of integer combinations of the columns of Γ, thus ensuring that q_2 remains one.

3.3.8. Updating the Step Length. The algorithm for updating Δ_k is exactly as given in Algorithm 2. Again, θ is usually set to $1/2$ and $\Lambda = \{1\}$ so that $\tau = 2$, $w_0 = -1$, and $w_1 = 0$.

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.

3.4. 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 general 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 global first-order stationary point convergence result 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].

A simple example (with $B_k = I$) showing all possible trial steps as well as the underlying simplices for a problem in $\mathbf{R}^2$ is given in Fig. 7.

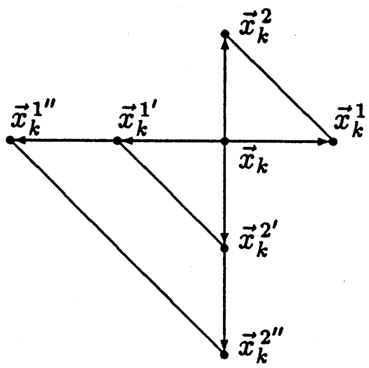

FIG. 7. *All possible trial steps for multidirectional search in* $\mathbf{R}^2$.

The multidirectional search algorithm is described in some 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 employ the general convergence theory given in §4 and makes clear the similarities between this and other pattern search methods.

3.4.1. The Basis Matrices. One of the most interesting features of the multidirectional search algorithm lies in the use of multiple basis matrices. The family of basis matrices $\mathbf{B}$ consists of all matrices representing the edges adjacent to each vertex in a nondegenerate n-dimensional simplex, a simplex that the user is allowed to specify. We add the matrices representing all possible permutations of the columns of the $(n+1)$ matrices we have just constructed. We then add the product of these $(n+1)!$ basis matrices with $-I$. Thus $|\mathbf{B}| = 2(n+1)!$. For the example shown in Fig. 7, the matrices representing the edges adjacent to each vertex in the original nondegenerate 2-dimensional simplex would be:

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

We then add all possible permutations of the columns of these three matrices:

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

Finally, we add the product of these six basis matrices with $-I$:

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

Fortunately, there is no need to actually construct and "store" this unwieldy number of basis matrices to initialize the method. Instead, as we will show in §3.4.6, given the outcome of the exploratory moves, Algorithm 8 can be used to update the basis matrix after each iteration by reconstructing the new basis matrix from the trial points considered during the course of the exploratory moves.

We will also defer to §3.4.6 a discussion of the fact that the choice of an initial simplex defines the family of basis matrices across *all* iterations of the multidirectional search algorithm.

3.4.2. The Core Pattern. Once again, the core pattern is the identity matrix, so $\Gamma = I$.

3.4.3. The Generating Matrix. The generating matrix for the multidirectional search algorithm is $C_k = \left[I \ -I \ -\mu I \ \vec{0}\right]$. Thus, C_k contains $p = 3n + 1$ columns.

We will define μ in §3.4.4 (μ is equal to τ^{w_2}, for $w_2 \in \mathbf{N}$ and thus must be greater than one; μ is typically 2). For $n = 2$ we have

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

We set $F = [I \ -I]$.

For the example shown in Fig. 7, the pattern would then be as shown in Fig. 8

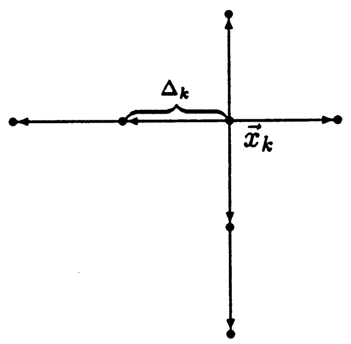

FIG. 8. *The pattern for multidirectional search in* $\mathbf{R}^2$ *with given step length* Δ_k *and basis matrix* $B_k = I$.

3.4.4. The Constants of Proportionality. A convenient feature of using the edges of a simplex to form the set of basis matrices is that the $\beta^j_{\psi,l}$'s from (5) belong to the set $\{-1, 0, 1\}$ for any choice of reference matrix B_ν. Thus $q_1 = 1$.

We must also satisfy the proportionality property found in (6) and (7). Since $\mu = \tau^{w_2}$ for some $w_2 \in \mathbf{N}$, and τ is rational, we can express μ as a fraction μ_n/μ_d where $\mu_n, \mu_d \in \mathbf{N}$ and μ_n, μ_d are relatively prime. Since $\Gamma = I$, $q_2 = \mu_d$. The multidirectional search algorithm usually is specified so that $\tau = 2$ and $w_2 = 1$, so that $\mu = \tau^{w_2} = 2$. Thus, typically, $q_2 = 1$.

Algorithm 7. Exploratory Moves Algorithm for Multidirectional Search.

```
Given $\vec{x}_k$, $\Delta_k$, $f(\vec{x}_k)$, $B_k$, and $\mu > 1$, set $\vec{s}_k = \vec{0}$, $\rho_k = 0$, $min = f(\vec{x}_k)$,
$\lambda_k = 1$, $scale = 1/\Delta_k$, $best = 0$, and $\vec{x}_k^0 = \vec{x}_k$.
For $i = 1, \cdots, n$ do
    a) $\vec{s}_k^i = \Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
    b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, $\vec{s}_k = \vec{s}_k^i$, and $best = i$.
If $\rho_k \leq 0$ then
    For $i = 1, \cdots, n$ do
        a) $\vec{s}_k^i = -\Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k^i$. Compute $f(\vec{x}_k^i)$.
        b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, $\vec{s}_k = \vec{s}_k^i$, and $best = i$.
    If $\rho_k > 0$ then set $scale = 1/\mu\Delta_k$.
        For $i = 1, \cdots, n$ do
            a) $\vec{s}_k^i = -\mu\Delta_k B_k \vec{e}_i$ and $\vec{x}_k^i = \vec{x}_k + \vec{s}_k$. Compute $f(\vec{x}_k^i)$.
            b) If $f(\vec{x}_k^i) < min$ then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k^i)$, $min = f(\vec{x}_k^i)$, $\vec{s}_k = \vec{s}_k^i$, $best = i$,
               and $\lambda_k = \mu$.
Return.
```

3.4.5. The Exploratory Moves. The exploratory moves for the multidirectional search method are given in Algorithm 7; the $\vec{e}_i$'s denote the unit coordinate vectors.

Clearly, $\vec{s}_k \in \Delta_k P_k$. Since the exploratory moves algorithm will consider all steps of the form $\Delta_k B_k F = \Delta_k B_k [I \; -I]$, *unless* simple decrease is found after examining only the steps defined by $\Delta_k B_k \Gamma = \Delta_k B_k I$, this guarantees we satisfy the condition that if $min\{f(\vec{x}_k + \vec{\sigma}), \vec{\sigma} \in \Delta_k B_k F\} < f(\vec{x}_k)$, then $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$.

3.4.6. Updating the Basis Matrix. We can update the basis matrix after each iteration k of the generalized pattern search method by reconstructing the new basis matrix B_{k+1}, given the outcome of the exploratory moves, from the trial points $\vec{x}_k^i$, $i = 1, \cdots, n$, considered during the course of the exploratory moves. This procedure is given in Algorithm 8.

Algorithm 8. Updating B_k.

```
Given $B_k$, $scale$, $best$, and $\vec{x}_k^i$ for $i = 0, \cdots, n$, denote $B_k = [\vec{b}_k^i]$, $i = 1, \cdots, n$.
If $\rho_k > 0$ then
    For $i = 0, \cdots, (best - 1)$ do
        $\vec{b}_{k+1}^{i+1} = scale * (\vec{x}_k^i - \vec{x}_k^{best})$.
    For $i = (best + 1), \cdots, n$ do
        $\vec{b}_{k+1}^i = scale * (\vec{x}_k^i - \vec{x}_k^{best})$.
Otherwise
    For $i = 1, \cdots, n$ do
        $\vec{b}_{k+1}^i = \vec{b}_k^i$.
Return.
```

At iteration k, the basis matrix B_k represents the edges adjacent to the current iterate $\vec{x}_k$ (the "best" vertex). To see that the basis B_{k+1} constructed by Algorithm 8 is contained in **B** as required, we note that while the multidirectional search algorithm may reflect or rescale the simplex, it does not change the basic shape of the simplex. The rescaling of the simplex is handled by the choice of Δ_k; we eliminate this from the construction of the basis for the next iteration using the variable *scale*. Since the reflection is effected by applying the orthogonal transformation $-I$, the Euclidean inner product is preserved. Thus the reflection does not affect the angles in the simplex used to start the current iteration (as can be seen, for example, in Fig. 7) and we have included this alternative in our definition of **B**. Thus, an argument by induction shows that the B_{k+1} constructed by the update algorithm will be contained in **B**. (A further discussion can be found in [20].)

3.4.7. Updating the Generating Matrix. The generating matrix is fixed across all iterations, so there is no need for an algorithm to update C_k.

3.4.8. 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.

4. The Convergence Theory.

Having set up the machinery to define pattern search methods, and demonstrated that several of the better known direct search methods fall naturally within this framework, we are now ready to analyze these methods. This analysis will produce two interesting theorems. The first, developed in §4.1, demonstrates an algebraic fact about the nature of pattern search methods that requires *no* assumption on the function f. The second theorem, developed in §4.2, states the global first-order stationary point convergence result. The first theorem is critical to the proof of the second for it shows that we only need require simple decrease in f to ensure global convergence.

Before proceeding, we note that all norms will be Euclidean vector norms or the subordinate operator norm. We also define the level set of f at $\vec{x}_0$ to be

$$L(\vec{x}_0) = \{\vec{x} : f(\vec{x}) \leq f(\vec{x}_0)\}.$$

4.1. The Algebraic Structure of the Iterates. As we noted in the introduction, Hooke and Jeeves observed that direct search methods are distinguished from other search techniques by having a finite number of states that can be indexed by a set of integers. We will give mathematical rigor to this observation. 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 4.1. *There exists a constant $c_* > 0$, independent of k, such that for any trial step $\vec{s}_k^{\,i} \neq \vec{0}$ produced by a generalized pattern search method (Algorithm 1) we have*

$$c_*\Delta_k \leq \| \vec{s}_k^{\,i} \|.$$

Proof. Suppose $\vec{s}_k^{\,i} \neq \vec{0}$. From (4) and (9) we have

$$\begin{aligned}\vec{s}_k^{\,i} &= \Delta_k B_k \vec{c}_k^{\,i} \\ &= \Delta_k q_2^{-2} B_k \sum_{j=1}^{n} q_2^2 \alpha_{k,j}^i \vec{\gamma}^j,\end{aligned}$$

where q_2 is as defined in (6) and (7). Let

$$\vec{z}_k^{\,i} = \sum_{j=1}^{n} q_2^2 \alpha_{k,j}^i \vec{\gamma}^j;$$

from (6) and (7) we know that the components of $\vec{z}_k^{\,i}$ are integers.

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

$$\begin{aligned}\| \vec{s}_k^{\,i} \| &= \Delta_k q_2^{-2} \| B_k \vec{z}_k^{\,i} \| \\ &\geq \Delta_k q_2^{-2} \sigma_n(B_k) \| \vec{z}_k^{\,i} \| \\ &\geq \Delta_k q_2^{-2} \sigma_n(B_k).\end{aligned}$$

The last inequality holds because at least one of the components of $\vec{z}_k^{\,i}$ is a nonzero integer, and hence $\| \vec{z}_k^{\,i} \| \geq 1$.

Since $|\mathbf{B}|$ is finite, we can define

$$\sigma_n(\mathbf{B}) = \min_{B_\psi \in \mathbf{B}} \sigma_n(B_\psi),$$

and

$$\| \vec{s}_k^{\,i} \| \geq \Delta_k q_2^{-2} \sigma_n(\mathbf{B}).$$

□

From Lemma 4.1 we can conclude that the role of Δ_k as a step length parameter is to regulate backtracking and thus prevent excessively short steps.

There is one key feature, shared by all the pattern search methods, that makes clear the similarities between these methods. That is the fact that pattern search methods could be considered *adaptive grid search* algorithms.

The pattern search methods are "grid search" algorithms in the sense that there is an underlying grid structure which each of these methods manifests throughout the course of the search. The grid structure is preserved by the strict adherence to steps defined by the pattern P_k and by the careful restriction of the way in which the step length parameter Δ_k can be updated. Theorem 4.2 clarifies this aspect of pattern search methods.

The pattern search methods are "adaptive" grid search algorithms in two senses. First, none of the pattern search methods consider all possible grid points; subsets of these points are explored, based both on the pattern defined by the particular choice of pattern search method and on the success of the exploratory moves taken during previous iterations. Second, the update rules for Δ_k stated in Algorithm 2 give a clear indication of when, and how, to refine the grid to advance the search—which further clarifies the import of Lemma 4.1.

THEOREM 4.2. *Any iterate $\vec{x}_N$ produced by a generalized pattern search method (Algorithm 1) can be expressed in the following form:*

$$\vec{x}_N = \vec{x}_0 + \sum_{l=1}^{n} \left(\sum_{k=0}^{N-1} \tilde{N}_{k,l} \right) \left(\tau_{num}^{r_{LB}} \tau_{den}^{-r_{UB}} \Delta_0 q_1^{-1} q_2^{-2} \right) \vec{b}_\nu^l,$$

where

- *$\tau = \tau_{num}/\tau_{den}$ is the τ found in the algorithm for updating Δ_k (Algorithm 2),*
- *r_{LB} and r_{UB} are bounds on the exponents of τ* that depend on *N,*
- *Δ_0 is the initial choice for the step length control parameter,*
- *q_1 and q_2 are the constants of proportionality found in (5), (6) and (7),*
- *$\vec{b}_\nu^l$, $l = 1, \cdots, n$, are the columns of the reference matrix B_ν found in (1), and*
- *$\tilde{N}_{k,l}$, $l = 1, \cdots, n$, $k = 0, \cdots, N$ are* integers *that depend on the $\beta_{\psi,l}^j$ found in (2), the core pattern Γ, the $\alpha_{k,j}^i$ found in (4), the constants of proportionality q_1 and q_2, τ, r_{LB}, and r_{UB}.*

Proof. The generalized pattern search algorithm, as stated in Algorithm 1, guarantees that any iterate $\vec{x}_N$ is of the form

$$\vec{x}_N = \vec{x}_0 + \sum_{k=0}^{N-1} \vec{s}_k. \tag{12}$$

(We adopt the convention that $\vec{s}_k = \vec{0}$ if iteration k is unsuccessful.) We also know that the step $\vec{s}_k$ must come from the set of trial steps $\vec{s}_k^i$, $i = 1, \cdots, p$. The trial steps are of the form

$$\vec{s}_k^i = \Delta_k B_k \vec{c}_k^i = \Delta_k \sum_{j=1}^{n} c_{k,j}^i \vec{b}_k^j.$$

From (4) we have

$$c_{k,j}^i = \sum_{m=1}^{n} \alpha_{k,m}^i \gamma_j^m.$$

From (2) we have

$$\vec{b}_k^j = \sum_{l=1}^{n} \beta_{k,l}^j \vec{b}_\nu^l.$$

We can then express the trial steps in terms of the columns of the reference matrix B_ν:

$$\begin{aligned}\vec{s}_k^{\,i} &= \Delta_k \sum_{j=1}^{n} \left(\sum_{m=1}^{n} \alpha_{k,m}^i \gamma_j^m \right) \left(\sum_{l=1}^{n} \beta_{k,l}^j \vec{b}_\nu^{\,l} \right) \\ &= \Delta_k \sum_{j=1}^{n} \sum_{m=1}^{n} \sum_{l=1}^{n} \alpha_{k,m}^i \gamma_j^m \beta_{k,l}^j \vec{b}_\nu^{\,l} \\ &= \Delta_k \sum_{l=1}^{n} \left(\sum_{j=1}^{n} \sum_{m=1}^{n} \alpha_{k,m}^i \gamma_j^m \beta_{k,l}^j \right) \vec{b}_\nu^{\,l}.\end{aligned}$$

Thus,

$$\sum_{l=1}^{n} \left(\sum_{j=1}^{n} \sum_{m=1}^{n} \alpha_{k,m}^i \gamma_j^m \beta_{k,l}^j \right) \vec{b}_\nu^{\,l} \tag{13}$$

defines the search direction while Δ_k controls the length of the step. We now examine each of these two components of the step.

The search direction given in (13) can be expressed as a scaled sum of the columns of B_ν so that

$$\sum_{l=1}^{n} \left(\sum_{j=1}^{n} \sum_{m=1}^{n} \alpha_{k,m}^i \gamma_j^m \beta_{k,l}^j \right) \vec{b}_\nu^{\,l} = \sum_{l=1}^{n} \phi_{k,l}^i \vec{b}_\nu^{\,l}.$$

We have specified the pattern search methods so that

$$q_1 \beta_{k,l}^j \in \mathbf{Z}, \quad \forall\, j = 1, \cdots, n \quad \forall\, l = 1, \cdots, n \quad \forall\, k$$

and

$$q_2^2 \alpha_{k,m}^i \gamma_j^m \in \mathbf{Z}, \quad \forall\, i = 1, \cdots, p \quad \forall\, j = 1, \cdots, n \quad \forall\, m = 1, \cdots, n \quad \forall\, k.$$

Thus, we have

$$\vec{s}_k^{\,i} = \Delta_k \sum_{l=1}^{n} N_{k,l}^i \left(q_1^{-1} q_2^{-2} \right) \vec{b}_\nu^{\,l}$$

where $N_{k,l}^i \in \mathbf{Z}$ depends on q_1, q_2, $\alpha_{k,m}^i$, γ_j^m, and $\beta_{k,l}^j$.

Now consider the second component of the step, 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

$$\Delta_k = \theta^{p_k^0} \lambda_1^{p_k^1} \lambda_2^{p_k^2} \cdots \lambda_{|\Lambda|}^{p_k^{|\Lambda|}} \Delta_0, \tag{14}$$

where $p_k^i \in \mathbf{Z}$ and $p_k^i \geq 0$. 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 $\{w_0, w_1, \cdots, w_{|\Lambda|}\} \subset \mathbf{Z}$,

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

and

$$\lambda_i = \tau^{w_i}, \qquad w_i \geq 0, \qquad i = 1, \cdots, |\Lambda|.$$

We can thus rewrite (14) as:

$$\begin{aligned}\Delta_k &= (\tau^{w_0})^{p_k^0} (\tau^{w_1})^{p_k^1} (\tau^{w_2})^{p_k^2} \cdots (\tau^{w_{|\Lambda|}})^{p_k^{|\Lambda|}} \Delta_0 \\ &= \tau^{r_k} \Delta_0,\end{aligned}$$

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

$$\begin{aligned} r_{LB} &= \min_{1 \leq k \leq N} \{r_k\} \\ r_{UB} &= \max_{1 \leq k \leq N} \{r_k\}. \end{aligned}$$

Returning to (12) we have

$$\begin{aligned}\vec{x}_N &= \vec{x}_0 + \sum_{k=0}^{N-1} \Delta_k \sum_{l=1}^{n} \phi_{k,l} \vec{b}_\nu^l \\ &= \vec{x}_0 + \sum_{k=0}^{N-1} \tau^{r_k} \Delta_0 \sum_{l=1}^{n} N_{k,l} \left(q_1^{-1} q_2^{-2}\right) \vec{b}_\nu^l \\ &= \vec{x}_0 + \sum_{l=1}^{n} \left(\sum_{k=0}^{N-1} \tau^{r_k} N_{k,l} \right) \left(\Delta_0 q_1^{-1} q_2^{-2}\right) \vec{b}_\nu^l.\end{aligned}$$

Since τ is rational, we can express τ as

$$\tau = \frac{\tau_{num}}{\tau_{den}}$$

where $\tau_{num}, \tau_{den} \in \mathbf{N}$. Then

$$\begin{aligned}\vec{x}_N &= \vec{x}_0 + \sum_{l=1}^{n} \left(\sum_{k=0}^{N-1} \tau_{num}^{r_k - r_{LB}} \tau_{den}^{r_{UB} - r_k} N_{k,l} \right) \left(\tau_{num}^{r_{LB}} \tau_{den}^{-r_{UB}} \Delta_0 q_1^{-1} q_2^{-2} \right) \vec{b}_\nu^l \\ &= \vec{x}_0 + \sum_{l=1}^{n} \left(\sum_{k=0}^{N-1} \tilde{N}_{k,l} \right) \left(\tau_{num}^{r_{LB}} \tau_{den}^{-r_{UB}} \Delta_0 q_1^{-1} q_2^{-2} \right) \vec{b}_\nu^l,\end{aligned}$$

where the $\tilde{N}_{k,l}$ are integers. □

Theorem 4.2 synthesizes the requirements we have placed on the choice of the basis matrices, the core pattern, the generating matrix, the constants of proportionality, the definition of the trial steps, and the algorithm for updating Δ_k (Algorithm 2). One other consequence of this structure, though, is to give us a way to define a nested sequence of grids across which the exploratory moves for a pattern search method must be conducted. The following corollary to Theorem 4.2, which is still a strictly algebraic fact about the nature of pattern search methods, gives us a first step towards the proof of global convergence for these methods.

COROLLARY 4.3. *Suppose that for a generalized pattern search method, there exist* $0 < \Delta_{LB}$ *and* $\Delta_{UB} < +\infty$ *such that* $\Delta_{LB} < \Delta_k < \Delta_{UB}$ *for all* k. *Then the sequence of iterates* $\vec{x}_k$ *produced by a generalized pattern search method (Algorithm 1) has no accumulation points.*

Proof. From Theorem 4.2 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 $\Delta_{LB} < \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. The hypothesis that $\Delta_k < \Delta_{UB}$ for all k 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

$$r_{LB} = \min_k\{r_k\}$$
$$r_{UB} = \max_k\{r_k\}.$$

From these bounds on the sequence $\{r_k\}$, together with Theorem 4.2, we see that the sequence of iterates $\{\vec{x}_k\}$ has no accumulation points. □

4.2. First-Order Stationary Point Convergence. We are now ready to state the global convergence theorem.

THEOREM 4.4. *Assume that* $L(\vec{x}_0)$ *is compact and that* $f : \mathbf{R}^n \to \mathbf{R}$ *is continuously differentiable on* $L(\vec{x}_0)$. *Then for the sequence of iterates* $\{\vec{x}_k\}$ *produced by the generalized pattern search method (Algorithm 1)*

$$\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| = 0\,.$$

We assume that f is continuously differentiable on $L(\vec{x}_0)$; however, this assumption can be weakened, as we shall discuss further in §5.

Before proving Theorem 4.4 we need the following results. We will defer the proofs of Lemma 4.5 and Proposition 4.6 to §7 in part because we wish to discuss there several other issues that are interesting but tangential to the proof of Theorem 4.4. It is also the case that the proofs of these two results are similar in style to those given for the equivalent results found in [20], though they are restated in terms of the machinery we developed in §2.

Lemma 4.5 establishes that pattern search methods are *descent methods.*

LEMMA 4.5. *Suppose that* f *is continuously differentiable on* $L(\vec{x}_0)$. *If* $\nabla f(\vec{x}_k) \neq 0$, *then there exists* $p \in \mathbf{Z}$, $p \geq 0$, *such that* $\rho_{k+p} > 0$ *(i.e., the* $k + p$*'th iteration is* successful*).*

The proof of Theorem 4.4 is by contradiction. We will posit the following *null* hypothesis: $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \neq 0$, or, equivalently, for all but finitely many k the sequence of iterates $\{\vec{x}_k\}$ stays bounded away from the set of stationary points

$$X_* = \{\vec{x} : \nabla f(\vec{x}) = 0\}. \tag{15}$$

The result in Proposition 4.6 spells out a consequence of the null hypothesis.

PROPOSITION 4.6. *Assume that* $L(\vec{x}_0)$ *is compact and* f *is continuously differentiable on* $L(\vec{x}_0)$. *Suppose that* $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \neq 0$. *Then there exists a constant* $\Delta_{LB} > 0$ *such that for all* k,

$$\Delta_{LB} < \Delta_k.$$

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

We are now ready to prove Theorem 4.4.

Proof. (Theorem 4.4.) The proof is by contradiction.

Suppose that $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \neq 0$. Then Proposition 4.6 holds.

Meanwhile, we also know that the sequence $\{\Delta_k\}$ is bounded above because all the iterates $\vec{x}_k$ must lie inside the level set $L(\vec{x}_0)$ and the latter set is compact; Lemma 4.1 then guarantees the upper bound on $\{\Delta_k\}$.

Corollary 4.3 says that the sequence of iterates $\vec{x}_k$ has no accumulation points. The simple decrease condition ensures that $\{\vec{x}_k\} \subset L(\vec{x}_0)$ as well. Since $L(\vec{x}_0)$ is compact, it follows from the Bolzano-Weierstrass theorem that $\{\vec{x}_k\}$ must be a finite set. Thus, under the supposition that $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \neq 0$, a pattern search method can visit only a finite number of points.

However, this contradicts the fact, stated in Lemma 4.5, that if $\nabla f(\vec{x}_k) \neq \vec{0}$, then pattern search methods guarantee simple decrease in a finite number of iterations. Therefore

$$\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| = 0$$

as required. □

5. Parallels with Line Search and Model Trust Region Theory.

The global convergence theory we have just presented for the pattern search methods shares similarities with the global convergence theory 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 recently reviewed in the survey by Nocedal [13]: we consider iterations of the form $\vec{x}_{k+1} = \vec{x}_k + \vec{s}_k$ where $\vec{s}_k \in \Delta_k P_k$; 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 $\vec{x}_k$ 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 Corollary 4.3 and Proposition 4.6, we consider the length of the step.

Given these major components for the convergence theory, the one that is most unusual, and certainly the most unexpected, 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

$$f(\vec{x}_k + \Delta_k \vec{p}_k) \leq f(\vec{x}_k) + \sigma_1 \Delta_k \nabla f(\vec{x}_k)^T \vec{p}_k$$

$$\nabla f(\vec{x}_k + \Delta_k \vec{p}_k)^T \vec{p}_k \geq \sigma_2 \nabla f(\vec{x}_k)^T \vec{p}_k,$$

where $\vec{p}_k$ 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 Corollary 4.3 is that the Armijo–Goldstein–Wolfe conditions need not be enforced to guarantee that pattern search methods will converge to a stationary point of the function. 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 grid structure that is at the heart of the proof of Theorem 4.2. Once again we stress that this grid structure is a purely algebraic fact about the nature of pattern search methods.

What does this algebraic structure mean for the convergence theory? 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 the core pattern (and its negative), 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 the core matrix and its negative. 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

[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 on how to construct such a sequence in practice are provided.

introduces backtracking into the generalized pattern search methods, which prevents steps that are too short. This is the import of Lemma 4.1.

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 core pattern 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.

The algebraic structure also prevents steps that are too long. The rigidity of the algebraic structure prevents arbitrary directions and arbitrary step lengths.

The net effect of the algebraic rigidity 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, but exist nonetheless. So much so, in fact, that 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 §3, 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 current basis matrix. Recall that the basis matrices are required to be nonsingular and that all but the multidirectional search method typically use the identity matrix as a basis matrix. Thus, for all the pattern search methods (ignoring the acceleration steps found in the Hooke and Jeeves pattern search algorithm and the multidirectional search algorithm, which lie outside the trust region radius) all steps satisfy $\|B_k^{-1}\vec{s}_k^{\,i}\|_\infty = \Delta_k$. To see this, note all possible trial points for problems in $\mathbf{R}^2$ that could be generated by coordinate search ($\vec{x}_k^{\alpha}$, $\vec{x}_k^{\beta}$, $\vec{x}_k^{\gamma}$, $\vec{x}_k^{\delta}$, $\vec{x}_k^{\epsilon}$, $\vec{x}_k^{\zeta}$, $\vec{x}_k^{\eta}$, $\vec{x}_k^{\theta}$), the Hooke and Jeeves pattern search (the same possibilities as for coordinate search), the full factorial design of Box and Wilson ($\vec{x}_k^{\beta}$, $\vec{x}_k^{\delta}$, $\vec{x}_k^{\zeta}$, $\vec{x}_k^{\theta}$), and multidirectional search ($\vec{x}_k^{\alpha}$, $\vec{x}_k^{\gamma}$, $\vec{x}_k^{\epsilon}$, $\vec{x}_k^{\eta}$), as seen in Fig. 9.

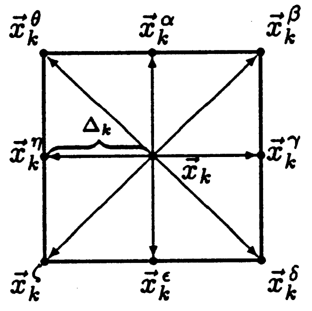

FIG. 9. *The sample trust region in* $\mathbf{R}^2$.

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. If the sampling did produce simple decrease, we may choose to *increase* the size of the trust region radius—an option that traditionally has not been considered for pattern search methods, but one that the global convergence theory clearly allows. Changing the radius of the trust region may affect more than simply the length of the step taken. While the set of search directions is fixed by the pattern, the exploratory moves algorithm may return any step that provides simple decrease as long as the step is defined by the pattern. Thus, changing the radius of the trust region before accepting a step may have the effect of changing the search direction selected by the exploratory move algorithm—a feature not found in the usual line search methods but shared with model trust region methods.

It is worth comparing the global first-order stationary point convergence results for model trust region methods and generalized pattern search methods. Under the hypotheses of Theorem 4.4, the results due to Powell [15] and Thomas [18] (see [11]) give

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

for the sequence of iterates generated by a model trust region method. We have shown that

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

for the sequence of iterates generated by a generalized pattern search method. The gap between the two results is due to the fact that the model trust region methods link the size of the step to the norm of the gradient through the notion of fraction of Cauchy decrease. The generalized pattern search methods do not explicitly enforce such a condition and hence lead to a weaker result.

A further connection with trust region algorithms is that we can allow the trust region radius Δ_k to increase if the algorithm is making good progress. This is a marked departure from other attempts to prove convergence for methods of this sort (e.g., [6] and [22]) which force the steps to be monotonically decreasing in length. The multidirectional search algorithm is the only one of the pattern search methods that explicitly increases the trust region radius Δ_k (the "pattern steps" of Hooke and Jeeves are another attempt to allow longer steps even though Δ_k is not allowed to increase), but it could just as easily be incorporated into any of the classical pattern search methods. Like the model trust region theory, the controlled expansion of the trust region radius plays no significant role in the convergence theory we have presented, but it often makes good algorithmic sense.

One other point worth making is that pattern search methods are well-defined even when the function to be minimized is not differentiable. In fact, it is trivial to extend Theorem 4.4 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 modification to the theory is 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

$$\Omega_\epsilon = \{\vec{x} \in L(\vec{x}_0) : \text{dist}(\vec{x}, X_*) \geq \epsilon\}.$$

is necessary for the proof of Theorem 4.4, the definition of X_* given in (15) 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.

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 a first-order stationary point convergence result for this class of algorithms. This convergence result is perhaps surprising, given the simplicity of pattern search methods, but derives from the algebraic rigidity with which pattern search methods conduct their local searches. This is gratifying, since while this rigidity originally was introduced as a heuristic for directing the local search, it turned out to be the key to proving convergence as well.

7. Appendix.

We deferred the proofs of Lemma 4.5 and Proposition 4.6 both because their proofs are closely related and because there are several additional notions that we must introduce before tackling the proof of Proposition 4.6.

To do so, we return to our definition of the pattern as $P_k = B_k C_k$ to show that the pattern contains at least one direction of descent whenever $\nabla f(\vec{x}_k) \neq 0$.

We require that the columns of C_k contain both the core pattern Γ and its negative $-\Gamma$ as subsets. Thus, P_k can be partitioned as follows:

$$P_k = B_k C_k = B_k \,[\,\Gamma \;\; -\Gamma \;\; A_k\,] = B_k \,[\,F \;\; A_k\,].$$

We shall now elaborate on these requirements. Since Γ is an $n \times n$ nonsingular matrix, and B_k comes from a family of $n \times n$ nonsingular matrices, we are guaranteed that $B_k\Gamma$ forms a basis for $\mathbf{R}^n$ (as does $-B_k\Gamma$). Thus at any iteration k, if $\nabla f(\vec{x}_k) \neq 0$ we are guaranteed that $B_k \vec{c}_k^{\,i}$ will be a direction of descent for at least one column $\vec{c}_k^{\,i}$ contained in the partition F of the columns of the generating matrix C_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 show that this is the case in the following proof of Lemma 4.5.

Proof. (Lemma 4.5.) A key hypothesis placed on the exploratory moves (see Hypotheses on Exploratory Moves) is that if descent can be found for some one of the trial steps defined by $\Delta_k B_k F$, then the exploratory moves will return a step that produces descent.

Because $B_k C_k$ has rank n, if $\nabla f(\vec{x}_k) \neq 0$, then there exists at least one trial direction $\vec{d}_k^{\,i} = B_k \vec{c}_k^{\,i}$, where $\vec{c}_k^{\,i} \in \Gamma$, such that $\nabla f(\vec{x}_k)^T \vec{d}_k^{\,i} \neq 0$. But, since $-\vec{c}_k^{\,i} \in F$, without loss of generality,

$$\nabla f(\vec{x}_k)^T \vec{d}_k^{\,i} < 0.$$

Thus, there exists an $h_k > 0$ such that for $0 < h \leq h_k$,

$$f(\vec{x}_k + h\vec{d}_k^i) < f(\vec{x}_k).$$

If at iteration k, $\Delta_k > h_k$, then the iteration may be unsuccessful; that is, $\rho_k = f(\vec{x}_k) - f(\vec{x}_k + \vec{s}_k) \leq 0$. When the iteration is unsuccessful, the generalized pattern search method sets $\vec{x}_{k+1} = \vec{x}_k$ 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 further descent, i.e., a successful iteration, in at most p iterations. □

7.2. Uniform Linear Independence. The pattern P_k guarantees the existence of at least one direction of descent whenever $\nabla f(\vec{x}_k) \neq 0$. We now want to guarantee the existence of a bound on the angle between the direction of descent contained in $B_k F$ and the negative gradient at $\vec{x}_k$ (whenever $\nabla f(\vec{x}_k) \neq 0$). 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* (as introduced in [14]).

LEMMA 7.1. *For a pattern search algorithm, there exists a constant $\xi > 0$ such that for all $k \geq 0$ and $\vec{x} \neq 0$,*

$$\max\left\{\frac{|\vec{x}^T(\vec{x}_k^i - \vec{x}_k)|}{\|\vec{x}\|\|\vec{x}_k^i - \vec{x}_k\|}, i = 1, \cdots p\right\} \geq \xi. \tag{16}$$

Proof. To demonstrate the existence of ξ, we will first consider the simplest possible case, $\mathbf{B} = \{I\}$ and $C_k = \begin{bmatrix}\Gamma & -\Gamma & \vec{0}\end{bmatrix} = \begin{bmatrix}I & -I & \vec{0}\end{bmatrix}$, and use this to derive a bound for any choice of $\mathbf{B}$ and Γ that satisfies the conditions we have imposed.

LEMMA 7.2. *Assume $\nabla f(\vec{x}_k) \neq 0$. Let*

$$\vec{y} = \frac{\nabla f(\vec{x}_k)}{\|\nabla f(\vec{x}_k)\|}$$

and

$$\cos\theta(\vec{y}) = \max_{1 \leq j \leq n}\left\{|\vec{y}^T\vec{e}_j|\right\},$$

where the $\vec{e}_j$'s are the unit coordinate vectors.

If $\mathbf{B} = \{I\}$ and $C_k = \begin{bmatrix}I & -I & \vec{0}\end{bmatrix}$, then

$$\min\cos\theta(\vec{y}) = \frac{1}{\sqrt{n}}.$$

Proof. We have $|\vec{y}^T\vec{e}_j| = |y_j|$, where $\vec{y} = (y_1, \cdots, y_n)^T$. Since

$$\sum_{j=1}^{n} |y_j|^2 = 1,$$

we are guaranteed that $|y_j| \geq 1/\sqrt{n}$ for some j, so $|\vec{y}^T\vec{e}_j| \geq 1/\sqrt{n}$ for some j. Thus $\cos\theta(\vec{y}) \geq 1/\sqrt{n}$.

Now note that $\cos\theta(\vec{y})$ attains this lower bound for any

$$\vec{y} = \alpha_1\vec{e}_1 + \alpha_2\vec{e}_2 + \cdots + \alpha_n\vec{e}_n,$$

where $\alpha_j \in \{\pm 1/\sqrt{n}\}$. □

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

Assume a general basis matrix B_k, where $B_k \in \mathbf{B}$, and a general core matrix Γ. We adopt the notation $B_k\Gamma = [\vec{y}_k^1 \cdots \vec{y}_k^n]$. Then for any $\vec{x} \neq 0$ we have the following:

$$\cos\theta = \frac{\left|\vec{x}^T\vec{y}_k^j\right|}{\|\vec{x}\|\|\vec{y}_k^j\|} = \frac{\left|\vec{x}^T B_k\Gamma\vec{e}_j\right|}{\|\vec{x}\|\|B_k\Gamma\vec{e}_j\|} = \frac{\left|\left((B_k\Gamma)^T\vec{x}\right)^T\vec{e}_j\right|}{\|\vec{x}\|\|B_k\Gamma\vec{e}_j\|}.$$

If we set $\vec{w} = (B_k\Gamma)^T\vec{x}$, so that $\vec{x} = (B_k\Gamma)^{-T}\vec{w}$, we have

$$\begin{aligned}\cos\theta &= \frac{|\vec{w}^T\vec{e}_j|}{\|(B_k\Gamma)^{-T}\vec{w}\|\|B_k\Gamma\vec{e}_j\|} \geq \frac{|\vec{w}^T\vec{e}_j|}{\|(B_k\Gamma)^{-T}\|\|\vec{w}\|\|B_k\Gamma\|\|\vec{e}_j\|} \\ &= \frac{1}{\|(B_k\Gamma)^{-T}\|\|B_k\Gamma\|}\left(\frac{|\vec{w}^T\vec{e}_j|}{\|\vec{w}\|\|\vec{e}_j\|}\right) = \frac{1}{\|(B_k\Gamma)^{-1}\|\|B_k\Gamma\|}\left(\frac{|\vec{w}^T\vec{e}_j|}{\|\vec{w}\|\|\vec{e}_j\|}\right) \geq \frac{1}{\kappa(B_k\Gamma)}\frac{1}{\sqrt{n}},\end{aligned}$$

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

$$\cos\theta \geq \frac{1}{\kappa(B_k\Gamma)\sqrt{n}} > 0\,.$$

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

$$\xi = \min_{B_\psi \in \mathbf{B}} \left\{\frac{1}{\kappa(B_\psi\Gamma)\sqrt{n}}\right\}. \tag{17}$$

□

The bound given in (17) 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 $B_k\Gamma$ 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 $\mathbf{B}$ and Γ 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.

7.2.1. Uniform Linear Independence, Coordinate Search and Response Surface Methodology. Before proceeding, there is one other point worth noting. In §3.2 we observed that in some sense response surface methodology is dual to coordinate search. The notion of the uniform linear independence of the set of search directions allows us to further clarify this remark.

Without loss of generality, assume that the trial steps generated by the search pattern around the current iterate lie on the l_∞ unit sphere. For coordinate search, in the worst case, we consider the $2n$ unit vectors $\pm\vec{e}_j$, which correspond to the $2n$ vertices of the l_1 unit sphere inscribed inside the l_∞ unit sphere, as can be seen for $\mathbf{R}^2$ in Fig. 10. The worst case for the bound on the angle between the search directions

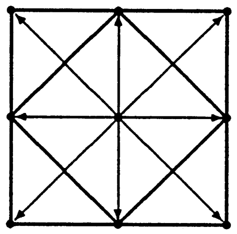

FIG. 10. *Coordinate search versus factorial design in* $\mathbf{R}^2$.

and the gradient at the current iterate occurs when the gradient is directed towards one of the 2^n vertices on the l_∞ unit sphere. The result is the bound $\xi = 1/\sqrt{n}$—the case demonstrated in Lemma 7.2.

In the basic variant of response surface methodology, we always consider the 2^n vertices of the l_∞ unit sphere at every iteration of the algorithm. Now the worst case for the bound on the angle between the search directions and the gradient at the current iterate corresponds to taking any one of the $2n$ vertices on the l_1 unit sphere as the (appropriately scaled) gradient—which once again gives the bound $\xi = 1/\sqrt{n}$.

This illustrates that while we may sample significantly more trial points for a single iteration of response surface methodology with a full factorial design ($O(2^n)$) than we do for a single iteration of coordinate search ($O(n)$), we do not gain any further improvement on the bound ξ.[2]

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 4.6.

[2] There may be, however, other reasons for generating the additional information. One important component of response surface methodology is the sensitivity analysis that can be done once a potential solution has been identified. For instance, the additional trial points can be used to obtain a least squares estimate of the gradient and the Hessian.

PROPOSITION 7.3. *Suppose that $L(\vec{x}_0)$ is compact and f is continuously differentiable on $L(\vec{x}_0)$. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{\vec{x} \in L(\vec{x}_0) : dist(\vec{x}, X_*) \geq \epsilon\}.$$

Suppose also that $\vec{x}_0 \in \Omega_\epsilon$. Then there exists $\delta > 0$, depending only on f, ξ from (16), and ϵ, such that if $\vec{x}_k \in \Omega_\epsilon$ and

$$\Delta_k \leq \delta,$$

then $\rho_k = f(\vec{x}_k) - f(\vec{x}_k + \vec{s}_k) > 0$; i.e., the iteration will be successful.

Proof. We restrict our attention to the fixed steps defined by the core pattern Γ and its negative, i.e., steps of the form $\Delta_k B_k F$. This is sufficient since the Hypotheses on Exploratory Moves ensure that a step $\vec{s}_k$ satisfying the simple decrease condition must be returned if a trial step satisfying the simple decrease condition is satisfied for a step defined by $\Delta_k B_k F$.

We first need some measure of the relative lengths of the steps defined by the core pattern Γ and its negative $-\Gamma$. We begin by defining

$$e^k = \min_{i=1,\cdots,2n} \|\vec{x}_k^i - \vec{x}_k\| = \min_{i=1,\cdots,2n} \|\vec{s}_k^i\| = \min_{j=1,\cdots,n} \|\Delta_k B_k \vec{\gamma}^j\|$$

and

$$E^k = \max_{i=1,\cdots,2n} \|\vec{x}_k^i - \vec{x}_k\| = \max_{i=1,\cdots,2n} \|\vec{s}_k^i\| = \max_{j=1,\cdots,n} \|\Delta_k B_k \vec{\gamma}^j\|.$$

We assume here that P_k is partitioned as in (8) so that the first $2n$ columns of P_k contain the columns of $[B_k F] = [B_k\Gamma \; -B_k\Gamma]$.

Since $|\mathbf{B}|$ is finite, we can define η as follows:

$$\eta = \min_{B_\psi \in \mathbf{B}} \frac{\min_{j=1,\cdots,n} \|B_\psi \vec{\gamma}^j\|}{\max_{j=1,\cdots,n} \|B_\psi \vec{\gamma}^j\|}.$$

Thus, for any $1 \leq i, j \leq 2n$,

$$\|\vec{x}_k^i - \vec{x}_k\| = \|\vec{s}_k^i\| \leq E^k \leq \frac{1}{\eta} e^k \leq \frac{1}{\eta}\|\vec{s}_k^j\| = \frac{1}{\eta}\|\vec{x}_k^j - \vec{x}_k\|. \tag{18}$$

Note that η is independent of k since the columns of the core matrix Γ are fixed across all iterations of a pattern search method. Observe also that $0 < \eta \leq 1$.

We define the *contour* $C(\vec{x}_0)$ to be

$$C(\vec{x}_0) = \{\vec{x} : f(\vec{x}) = f(\vec{x}_0)\}.$$

Since $\vec{x}_0 \in \Omega_\epsilon$, Lemma 4.5 allows us to define

$$N = \min\{k : \vec{x}_k \neq \vec{x}_0\}.$$

We define d to be the distance between the contour defined by $\vec{x}_0$ and the level set defined by $\vec{x}_N$

$$d = \text{dist}\,(L(\vec{x}_N), C(\vec{x}_0))\,.$$

Because both sets are compact and disjoint, we know that $d > 0$.

We now make the following two claims.

CLAIM 1. *Suppose $k \geq N$. If, for some $j = 1, \cdots, 2n$, $\|\vec{x}_k^j - \vec{x}_k\| \leq \eta d$, then $\vec{x}_k^i \in L(\vec{x}_0)$ for all $i = 1, \cdots, 2n$.*

Proof. The triangle inequality gives us

$$\begin{aligned} \text{dist}\left(\vec{x}_k^i, L(\vec{x}_N)\right) &\leq \text{dist}\left(\vec{x}_k^i, L(\vec{x}_k)\right) + \text{dist}\,(L(\vec{x}_k), L(\vec{x}_N)) \\ &= \text{dist}\left(\vec{x}_k^i, L(\vec{x}_k)\right) \\ &\leq \|\vec{x}_k^i - \vec{x}_k\|. \end{aligned}$$

From (18) we have

$$\|\vec{x}_k^i - \vec{x}_k\| \leq \frac{1}{\eta}\|\vec{x}_k^j - \vec{x}_k\|,$$

so that

$$\text{dist}\left(\vec{x}_k^i, L(\vec{x}_N)\right) \leq d,$$

for *any* choice of $i = 1, \cdots, 2n$. □

CLAIM 2. *If for some $i = 1, \cdots, 2n$, $\|\vec{x}_k^i - \vec{x}_k\| \leq \eta\frac{\epsilon}{3}$, then*

$$\vec{x}_k^j \in \Omega_{\epsilon/2} = \{\vec{x} \in L(\vec{x}_0) : \mathit{dist}(\vec{x}, X_*) \geq \epsilon/2\}$$

for any $j = 1, \cdots, 2n$.

Proof. Suppose $\vec{x}_* \in X_*$. We appeal to the "reverse" form of the triangle inequality to obtain

$$\|\vec{x}_k^j - \vec{x}_*\| \geq |\underbrace{\|\vec{x}_k^j - \vec{x}_k\|}_{\leq\frac{\epsilon}{3}} - \underbrace{\|\vec{x}_k - \vec{x}_*\|}_{\geq\epsilon}| > \frac{\epsilon}{2}.$$

□

Finally, let

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

By design, $\alpha > 0$. Since ∇f is continuous on $L(\vec{x}_0)$, ∇f is uniformly continuous on $L(\vec{x}_0)$. Thus, there exists $r > 0$, depending only on the ξ from (16) and α, such that

$$\|\nabla f(\vec{x}) - \nabla f(\vec{x}_k)\| \leq \tfrac{\xi\alpha}{2} \text{ whenever } \|\vec{x} - \vec{x}_k\| \leq r \text{ (and } \vec{x} \in L(\vec{x}_0)).$$

We define δ as follows:

$$\delta = \eta \min \{d, \; \epsilon/3, \; r, \; \Delta_{N-1}\}.$$

If $k < N$, then because $\delta \le \Delta_{N-1}$, if Δ_k were smaller than δ the iteration necessarily would be successful. In fact, this happens exactly when $k = N - 1$.

On the other hand, if $k \ge N$, we are now assured that if

$$E^k \le \delta, \tag{19}$$

then for $i = 1, \cdots, 2n$, $\vec{s}_k^{\,i}$ is contained in $L(\vec{x}_0)$ and for some $i = 1, \cdots, 2n$ satisfying (16), $\|\nabla f(\vec{x}_k^{\,i}) - \nabla f(\vec{x}_k)\| < \xi\alpha/2$. We are ready to argue that if, at any iteration k, (19) is satisfied, then an iteration of a pattern search method will be successful.

Choose a trial point $\vec{x}_k^{\,i}$, $i = 1, \cdots, 2n$, that satisfies both $\nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k) < 0$ and

$$\frac{|\nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k)|}{\|\nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\|} \ge \xi.$$

The definition of the pattern P_k and Lemma 7.1 guarantee the existence of at least one such $\vec{x}_k^{\,i}$.

The Mean Value Theorem tells us that

$$f(\vec{x}_k^{\,i}) - f(\vec{x}_k) = \nabla f(\vec{\omega})^T(\vec{x}_k^{\,i} - \vec{x}_k)$$

for some $\vec{\omega}$ contained in the line segment from ($\vec{x}_k^{\,i}$ to $\vec{x}_k$), whence

$$f(\vec{x}_k^{\,i}) - f(\vec{x}_k) = \nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k) + (\nabla f(\vec{\omega}) - \nabla f(\vec{x}_k))^T(\vec{x}_k^{\,i} - \vec{x}_k). \tag{20}$$

Consider the first term on the right-hand side of (20). Our choice of $\vec{x}_k^{\,i}$ gives us

$$\left|\nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k)\right| \ge \xi\|\nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\|.$$

Furthermore, since $\nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k) < 0$, we have

$$\nabla f(\vec{x}_k)^T(\vec{x}_k^{\,i} - \vec{x}_k) \le -\xi\|\nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\|. \tag{21}$$

Now consider the second term on the right-hand side of (20). The Cauchy–Schwarz inequality gives us

$$\left|(\nabla f(\vec{\omega}) - \nabla f(\vec{x}_k))^T(\vec{x}_k^{\,i} - \vec{x}_k)\right| \le \|\nabla f(\vec{\omega}) - \nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\|. \tag{22}$$

Combine (21) and (22) to rewrite (20) as

$$\begin{aligned} f(\vec{x}_k^{\,i}) - f(\vec{x}_k) &\le -\xi\|\nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\| + \|\nabla f(\vec{\omega}) - \nabla f(\vec{x}_k)\|\|\vec{x}_k^{\,i} - \vec{x}_k\| \\ &= (-\xi\|\nabla f(\vec{x}_k)\| + \|\nabla f(\vec{\omega}) - \nabla f(\vec{x}_k)\|)\|\vec{x}_k^{\,i} - \vec{x}_k\|. \end{aligned}$$

Since $\vec{\omega}$ lies on the line segment between $\vec{x}_k^{\,i}$ and $\vec{x}_k$, once $\|\vec{x}_k^{\,i} - \vec{x}_k\| \le \delta$,

$$f(\vec{x}_k^{\,i}) - f(\vec{x}_k) \le (-\xi\|\nabla f(\vec{x}_k)\| + \tfrac{\xi}{2}\|\nabla f(\vec{x}_k)\|)\|\vec{x}_k^{\,i} - \vec{x}_k\| < 0.$$

Thus, when $\|\vec{x}_k^i - \vec{x}_k\| \le \delta$,

$$f(\vec{x}_k) - f(\vec{x}_k^i) > 0.$$

The condition on the exploratory moves guarantee that if $\min\{f(\vec{x}_k+\vec{\sigma}), \vec{\sigma} \in \Delta_k B_k F\} < f(\vec{x}_k)$, then $f(\vec{x}_k + \vec{s}_k) < f(\vec{x}_k)$. Thus, $\rho_k = f(\vec{x}_k) - f(\vec{x}_k + \vec{s}_k) > 0$. ◻

We now will prove Proposition 4.6.

Proof. (Proposition 4.6.) Suppose that $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \ne 0$. Then we can find N_1 and $\epsilon > 0$ such that for all $k \ge N_1$,

$$\vec{x}_k \in \Omega_\epsilon = \{\vec{x} \in L(\vec{x}_0) : \operatorname{dist}(\vec{x}, X_*) \ge \epsilon\}.$$

We also know that there exists N_2 such that $\vec{x}_{N_2} \ne \vec{x}_0$; if $\vec{x}_k = \vec{x}_0$ for all k, this would mean that $\nabla f(\vec{x}_0) = \vec{0}$, contradicting the assumption that $\liminf_{k\to+\infty} \|\nabla f(\vec{x}_k)\| \ne 0$.

Let $N = \max(N_1, N_2)$. From Proposition 7.3 we are assured of $\delta > 0$ such that if $k \ge N$ and $\Delta_k < \delta$, then the iteration will be successful. Given Δ_0, there exists a constant $p \in \mathbf{Z}$, $p \ge 0$, such that

$$\theta^p \Delta_0 \le \delta.$$

Thus, for $k \ge N$, $\theta^{p+1}\Delta_0 < \Delta_k$.

Now set

$$\Delta_{LB} = \theta \min(\theta^p \Delta_0,\ \Delta_1,\ \cdots,\ \Delta_{N-1}).$$

Then for all k, $\Delta_{LB} < \Delta_k$. ◻

Acknowledgments. This paper benefitted greatly from my conversations with J. E. Dennis, Jr., Michael Trosset, and especially Robert Michael Lewis. In particular, my many 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 §4.1.

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. Grotschel, 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.