PATTERN SEARCH ALGORITHMS FOR MIXED VARIABLE PROGRAMMING[*]

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

Abstract. Many engineering optimization problems involve a special kind of discrete variable that *can* be represented by a number, but this representation has no significance. Such variables arise when a decision involves some situation like a choice from an unordered list of categories. This has two implications: The standard approach of solving problems with continuous relaxations of discrete variables is not available, and the notion of local optimality must be defined through a user-specified set of neighboring points. We present a class of direct search algorithms to provide limit points that satisfy some appropriate necessary conditions for local optimality for such problems. We give a more expensive version of the algorithm that guarantees additional necessary optimality conditions. A small example illustrates the differences between the two versions. A real thermal insulation system design problem illustrates the efficacy of the user controls for this class of algorithms.

Key words. pattern search algorithm, convergence analysis, bound constrained optimization, mixed variable programming, derivative-free optimization

AMS subject classifications. 49M30, 65K05, 90C11, 90C56

PII. S1052623499352024

1. Introduction. Torczon [12] defined a class of generalized pattern search methods to minimize a function $f : \Re^n \to \Re$ without any knowledge of its derivatives. She shows that the class includes algorithms such as *coordinate search* with fixed step sizes, *evolutionary operation* using factorial design [3], the original *pattern search algorithm* [7], and the *multidirectional search algorithm* [5]. In [12] she gave general convergence results under the assumption of continuous differentiability.

The main result of [12] is that for $f \in \mathcal{C}^1$, the sequence of iterates $\{x_k\}$ of $\Re^n$ generated by any generalized pattern search (GPS) method satisfies

$$(1.1) \qquad\qquad \liminf_{k \to \infty} \|\nabla f(x_k)\| = 0$$

without ever computing or explicitly approximating derivatives. At each iteration, the function is evaluated at trial points on a discrete mesh containing the current iterate in search of one yielding any decrease in the objective function value. Lewis and Torczon [10] use positive basis theory to strengthen the result by roughly cutting in half the worst case number of trial points at each iteration without affecting the convergence result. Lewis and Torczon [9], [11] extend pattern search algorithms and the convergence theory to bound and linearly constrained minimization by adapting the exploration of the domain near the boundary of the feasible region. The optimality

---

[†]Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, GÉRAD, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, H3C 3A7, Canada (charlesa@crt. umontreal.ca). The work of this author was supported by NSERC (Natural Sciences and Engineering Research Council) fellowship PDF-207432-1998 during a postdoctoral stay at Rice University.

[‡]Department of Computational and Applied Mathematics, Rice University, MS 134, 6100 South Main Street, Houston, TX 77005-1892 (dennis@caam.rice.edu).

condition guaranteed by their approach is the existence of a limit point $\hat{x}$ of the sequence of iterates $\{x_k\}$ that satisfies

(1.2) $(x - \hat{x})^T \nabla f(\hat{x}) \geq 0$ for any feasible x.

This condition reduces to (1.1) in the event that $\hat{x}$ is a strictly interior point.

Our purpose here is to further generalize the problem to be solved because many engineering optimization problems contain both continuous and discrete variables. Moreover, the discrete variables are often categorical ones; i.e., they refer to a list or set of categories and thus, the standard mixed integer approach of solving with continuous relaxations through branch and bound is not available. Of course, when branch and bound can be used, it probably should be, but that is not the issue here. Indeed, the context in which our algorithm is to be applied is that the variables are provided by the algorithm as input to a black box simulation. It would be surprising if one could run the simulation code with a continuous variable, where the simulation expects a discrete input—perhaps specifying the state of the physical medium under investigation.

We consider the problem of minimizing the function $f : \Omega \to \Re$, where the domain is partitioned into continuous and discrete variables Ω^c and Ω^d (some or all of the discrete variables may be categorical). The domain of the continuous variables is bound constrained $\Omega^c = [\ell, u]$, where $\ell, u \in \Re^{n^c} \cup \{\pm\infty\}$, $\ell < u$, and n^c is the dimension of the space. The domain of the discrete variable Ω^d has dimension n^d and may be represented by a subset of $\mathcal{Z}^{n^d}$. The continuous and discrete components of the iterates generated by the method will be denoted by $x_k = (x_k^c, x_k^d)$, where $x_k^c \in \Re^{n^c}$ and $x_k^d \in \mathcal{Z}^{n^d}$. We understand that we are abusing notation here, since we certainly mean that $x_k \in \Re^{n^c} \times \mathcal{Z}^{n^d}$. However, the purpose of notation is to explicate, not to be pedantic, and we are sure the reader will forgive us.

The function f is assumed to be continuously differentiable when the discrete variables in Ω^d are fixed. We present a general mixed variable pattern (GMVP) search method that reduces to that of Lewis and Torczon [9] when the dimension n^d is fixed to zero. Thus, like them, we deal with infeasible trial points by setting $f(x)$ to a large value.

A second objective of the paper is to slightly generalize the part of the algorithm that deals with the continuous variables and to revise and shorten the arguments developed in [12] and [10]. We first show how to obtain a limit point $\hat{x}$ of the sequence of iterates that satisfies first-order optimality conditions with respect to the continuous variables. These conditions reduce to (1.2) when there are no discrete variables. We also guarantee that the same limit point $\hat{x}$ satisfies some local optimality conditions with respect to the discrete variables. The notion of local optimality is defined through the user-specified set of neighbors $\mathcal{N}(x) \subset \Omega$ described in section 2.1. We also present a second version of the algorithm that guarantees stronger results.

The paper is structured as follows. First, we present a definition of local optimality for mixed variable programming and the optimality conditions guaranteed by our algorithm. We use a design problem for a thermal insulation system to illustrate categorical variables and our version of local optimality. Then in section 3, we formally describe a general framework for pattern search algorithms with mixed variables. In section 4, we provide the analysis to specify a subsequence of iterates whose limit points satisfy optimality conditions, including a stronger version of the algorithm that uses more function evaluations per iteration to guarantee an additional necessary optimality condition. Section 5 illustrates the difference between

the two versions of the algorithm on a small example, and it reports results for the algorithm applied to the problem in section 2.2. We use that example to illustrate some controls the user has to spend more function evaluations to gain a better local optimum.

2. Mixed variables.

2.1. Local optimality for mixed variables. In the absence of discrete variables, the definition of local optimality is straightforward: $\hat{x} \in [\ell, u]$ is a local minimizer of the bound constrained function f if there exists an $\epsilon > 0$ such that $f(\hat{x}) \leq f(v)$ for all $v \in [\ell, u]$ in a ball $B(\epsilon, \hat{x})$ of radius ϵ around $\hat{x}$.

When the optimization problem contains only discrete variables, a definition of local optimality might be the following: $f(\hat{x}) \leq f(y)$ for all y in $\mathcal{N}(\hat{x})$, where $\mathcal{N}(\hat{x})$ is a finite set of neighbors including and around the discrete variable $\hat{x}$. This specifies the quality of the solution for which one is willing to pay the necessary function values by defining the notion of "local optimality" the algorithm is to achieve with respect to the discrete variables.

An example is the quadratic assignment problem (QAP) in which n facilities must be assigned to n locations: each assignment may be represented using one of the $n!$ permutations of the vector $(1, 2, \ldots, n)$. The key point in the definition is for the user to answer the question, What property must the solution provided by the algorithm satisfy in order for it to be a satisfactory local solution? One might decide, for example, that a QAP solution is interesting if a given assignment x could not be improved by changing x in at most two locations (or, more stringently, in at most three locations). Our approach is completely flexible in this respect; however, the more stringent the conditions of local optimality the user wants to impose, the more expensive the GMVP poll step will be.

Consider, for example, the QAP with three facilities. It may be modeled with three discrete variables ($x^d \in \mathcal{Z}^3$). Not all the points of the integer lattice $\mathcal{Z}^3$ represent feasible assignments, only the permutations of $(1, 2, 3)$. Also, the ordering is not the classical one associated with an inherited metric, since for the set of neighbors $\mathcal{N}(1, 2, 3) = \{(1, 2, 3), (1, 3, 2), (3, 2, 1), (2, 1, 3)\}$ the assignment $(3, 2, 1)$ seems nearer than $(3, 1, 2)$ to $(1, 2, 3)$. Observe that in this example the constraints that define Ω^d (i.e., the set of permutations) are modeled through the definition of the set of neighbors $\mathcal{N}$.

Thus, definition of the set of neighbors $\mathcal{N}$ represents one of the tuning knobs available to the user willing to pay more for a guarantee of a stronger local optimizer. As our thermal example shows, this does not guarantee finding a lower function value, but it does guarantee a wider set of changes that will not produce a better function value. A better way to use this knob is the way we used it in the thermal example—to save evaluating alternatives that are highly unlikely to improve the function value, and thus decrease the cost of the more expensive poll steps in which local exploration is required.

For mixed variable programming, the definition of local optimality must take into account variations of both the continuous and discrete variables. Indeed, in defining $\mathcal{N}(x)$, one would probably need to allow for changes in the continuous as well as the discrete components. That is to say, changing the discrete variables may make no sense without some attendant change in the continuous components as well. We propose the following definition.

DEFINITION 2.1. *The solution $\hat{x} = (\hat{x}^c, \hat{x}^d) \in \Omega$ is said to be a local minimizer of*

f with respect to the set of neighbors $\mathcal{N}(\hat{x})$ if there exists an $\epsilon > 0$ such that

$$f(\hat{x}) \leq f(v) \quad for\ any \quad v \in \bigcup_{y \in \mathcal{N}(\hat{x})} \left(([\ell, u] \cap B(\epsilon, y^c)) \times y^d \right),$$

where $\mathcal{N}(\hat{x}) \subset \Omega$ is a finite set of points.

We require a notion of continuity with respect to the set of neighbors: If $\{x_k\}$ is a sequence that converges to $\hat{x}$, then $\mathcal{N}(x_k)$ converges to $\mathcal{N}(\hat{x})$; i.e., for any $\epsilon > 0$ and $\hat{y}$ in the set of neighbors $\mathcal{N}(\hat{x})$, there exists a y_k in $\mathcal{N}(x_k)$ that also belongs to the ball $B(\epsilon, \hat{y})$.

This definition of local optimality requires the user to decide how to define the neighbors, and then we produce a point at which we guarantee that there are no better solutions than $\hat{x}$ in any of the balls (in the continuous space and intersected with the box $[\ell, u]$) around the points in the user-defined set of neighbors. Observe that when there are no discrete variables, or else no continuous ones, this definition reduces to the appropriate one presented above.

Of course, one can generally prove only that an optimization algorithm converges to a point satisfying some necessary conditions for optimality. Thus, we prove that our algorithm produces a limit point $\hat{x}$ that satisfies

$$(2.1) \qquad (x^c - \hat{x}^c)^T \nabla^c f(\hat{x}) \geq 0 \qquad \text{for any feasible } (x^c, \hat{x}^d)$$

(where $\nabla^c f(x) \in \Re^{n^c}$ denotes the gradient of f with respect to the continuous variables x^c while keeping the discrete x^d fixed), and for any $\hat{y} \in \Omega$ in the set of neighbors $\mathcal{N}(\hat{x})$

$$(2.2) \qquad\qquad\qquad\qquad f(\hat{x}) \leq f(\hat{y}).$$

In the cases where $f(\hat{y}) < f(\hat{x}) + \xi$ (for a specified $\xi > 0$), there exists a point $\hat{z} \in \Omega$, whose discrete components $\hat{z}^d$ are identical to $\hat{y}^d$, that satisfies $f(\hat{x}) \leq f(\hat{z}) \leq f(\hat{y})$ and

$$(2.3) \qquad (z^c - \hat{z}^c)^T \nabla^c f(\hat{z}) \geq 0 \qquad \text{for any feasible } (z^c, \hat{z}^d).$$

Furthermore, in the cases where $f(\hat{x}) = f(\hat{y})$ and $\hat{y} \neq \hat{z}$,

$$(2.4) \qquad\qquad\qquad\qquad f(\hat{x}) = f(\bar{y})$$

for an infinite number of intermediate points $\bar{y} \in \Omega$ between $\hat{y}$ and $\hat{z}$ (we show in section 4.2 how to construct these intermediate points). Moreover, we present a stronger version of the algorithm that guarantees

$$(2.5) \qquad (y^c - \bar{y}^c)^T \nabla^c f(\bar{y}) \geq 0 \qquad \text{for any feasible } (y^c, \bar{y}^d)$$

whenever $f(\hat{x}) = f(\hat{y})$.

2.2. An illustrative application. We illustrate our approach on a thermal insulation system. The problem is thoroughly described in [8], where we show that by considering the two categorical variables we obtain a 65% better objective function value than in the earlier work of Hilal and Boom [6], who considered only the continuous variables.

The setting of the problem is as follows. One wishes to control the heat flow from a hot to a cold surface by inserting some shields (heat intercepts) between them.
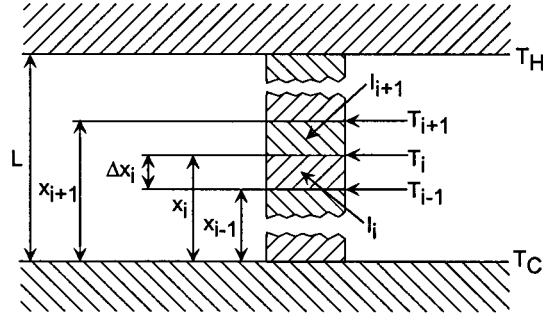
FIG. 2.1. *Schematic of a general thermal insulation system.*

Each shield is kept at a fixed temperature, and the spaces between them are filled by various insulators. The objective is to minimize the power f (with an extra cost for each additional shield) required to keep the shields at their temperatures. This is illustrated in Figure 2.1.

The temperatures of the hot T_H and cold T_C surfaces are given. The decision variables are the number of shields n, their temperatures $T \in \Re^n$, the spacing $\Delta x \in \Re^{n+1}$ between them (or equivalently, the thickness of the insulators), and the types of insulators I between the shields. These last variables are taken from a finite list $\mathcal{I}$ of insulators whose thermal conductivity properties are known. The optimization problem may be formulated as

$$
\begin{aligned}
\min_{\Delta x, T, n, I} \quad & f(\Delta x, T, n, I) \\
\text{subject to} \quad & \Delta x \geq 0, \quad T_C \leq T_i \leq T_H, \quad i = 1, 2, \ldots, n, \\
& n \in \mathcal{Z}^+, \quad I_j \in \mathcal{I}, \qquad j = 1, 2, \ldots, n+1.
\end{aligned}
$$

The continuous variables are Δx and T, and the categorical ones are n and I. An interesting and challenging aspect of this problem is that the number of decision variables depends on a decision variable n. This does not complicate the theory.

Section 5.2 contains numerical results for this problem with local optimality defined through sets of neighbors $\mathcal{N}$ as follows:
- changing the type of one insulator;
- removing one shield and an adjacent insulator;
- adding a shield and an insulator.

3. Pattern search methods. The underlying structure of a pattern search algorithm is as follows. It is an iterative method that generates a sequence of feasible iterates whose objective function value is nonincreasing. At any given iteration, the objective function is evaluated at a finite number of points on a mesh in order to try to find one that yields a decrease in the objective function value.

Any iteration k of a pattern search method is initiated with the *incumbent solution* x_k, as well as with an enumerable subset $\mathcal{M}_k$ of the domain $\Omega \subset \Re^{n^c} \times \mathcal{Z}^{n^d}$. Construction of the mesh $\mathcal{M}_k$ is formally described in section 3.1, and its fineness, or resolution, is parameterized by a positive real number Δ_k. The goal of each iteration is to obtain a new incumbent solution on the current mesh whose objective function value is strictly less (by any amount) than the old incumbent.

Exploration of the mesh is conducted in one or two phases. First, a finite search, free of any other rules imposed by the algorithm, is performed anywhere on the mesh. Any strategy can be used as long as it searches finitely many points (including none). This part of the algorithm has the advantage that the user can put in place any ad hoc search he/she might favor for improving the incumbent with the knowledge that if this fails, the next phase will provide a fail-safe.

If the search does not succeed in improving the incumbent, the second phase is called. A potentially exhaustive (but always finite) search in a local mesh neighborhood around x_k and around promising points in its set of neighbors is performed. The first phase (called the SEARCH step) provides flexibility to the method and determines in practice the global quality of the solution; a user can do a more extensive, and expensive, search in hopes of finding a better local solution. The second phase (called the POLL step) follows stricter rules and guarantees theoretical convergence to a local minimizer of a quality specified by the user. The set of points visited by this phase is referred to as the *poll set*. Rules for constructing this set are detailed in section 3.2.

If a point with a better objective value than x_k is found in either phase, then the iteration is declared *successful*, the better point becomes the new incumbent, and the next iteration is initiated with a (possibly) coarser (and different) mesh around the new incumbent solution.

Otherwise, the iteration is declared *unsuccessful*. The next iteration is initiated at the same incumbent solution, but with a finer mesh on the continuous variables, and a set of neighbors "closer" (if possible) to the incumbent solution. A key property of the mesh exploration is that if an iteration is unsuccessful, then the current objective function value is less than or equal to the objective function values evaluated at all points in the trial set consisting of all points considered in the search and poll set.

In order to properly present the pattern search algorithm, we first detail in the following sections the construction of the mesh and the poll set.

3.1. The mesh. At any given iteration k, the *current mesh* $\mathcal{M}_k$ is a discrete set of points in Ω from which the algorithm selects the next iterate. The mesh is conceptual; it is not actually constructed. The coarseness or fineness of the mesh is dictated by the strictly positive *mesh size parameter* $\Delta_k \in \Re_+$. Both the mesh and mesh size parameter are updated at every iteration.

The mesh is the direct product of the union of a finite number of lattices in $\Re^{n^c}$ with the integer space $\mathcal{Z}^{n^d}$. Our presentation of the lattices differs from that of Torczon [12], but the sets produced are equivalent. Consider the *basis matrix* $G^B \in \Re^{n^c \times n^c}$, and for j varying from 1 to $j_{max} < \infty$, consider the *generating matrices* $G_j \in \mathcal{Z}^{n^c \times n^c}$; then define the *pattern matrices* $P_j \in \Re^{n^c \times n^c}$ to be the products $G^B G_j$. The continuous variables are chosen from one of the translated (by x_k^c) lattices $\left\{ x_k^c + \Delta_k P_j z : z \in Z^{n^c} \right\}$ for $j = 1, 2, \ldots, j_{max}$. The continuous part x_k^c of the current iterate belongs to each of the j_{max} lattices regardless of the value of the parameter Δ_k. The basis matrix G^B is constant over all iterations. However, in practice the generating matrices G_j (and thus P_j) that define the lattices can be determined as the iteration unfolds as long as only a finite number of them is generated.

Each of these lattices is enumerable, and the minimum distance between two distinct points is proportional to the mesh size parameter Δ_k. When an iteration is successful, the continuous part of the next iterate is chosen in any of these lattices and thus belongs to their union $\mathcal{M}(\Delta_k) = \bigcup_{j=1}^{j_{max}} \left\{ x_k^c + \Delta_k P_j z : z \in Z^{n^c} \right\}$; the discrete part is chosen in the integer lattice Z^{n^d}.

At iteration k, the current mesh is defined to be the direct product of $\mathcal{M}(\Delta_k) \cap \Omega^c$

by Ω^d

$$\mathcal{M}_k = \left(\mathcal{M}(\Delta_k) \times \mathcal{Z}^{n^d} \right) \cap \Omega.$$

The mesh is completely defined by the current iterate x_k and the mesh size parameter Δ_k. Whether or not the iteration is successful, the next iterate x_{k+1} is always selected in the current mesh $\mathcal{M}_k$.

In the case where the SEARCH step in the current mesh is unsuccessful, a second exploration phase must be conducted by the algorithm in the poll set before the iteration is declared *unsuccessful*. The POLL step verifies whether the incumbent solution is a local mesh minimizer, as defined in the next section.

3.2. The poll set. Polling occurs when the SEARCH step was unable to obtain a point on the current mesh that decreased the incumbent value. Polling is conducted as follows in up to three stages (not necessarily in this order):
  – polling with respect to the continuous variables;
  – polling on the current set of neighbors $\mathcal{N}$;
  – extended polling (in the case where $f(y)$ for some y in the set of neighbors is close to the incumbent value).
The first stage is identical to the typical polling in pattern search algorithms for continuous variables only. The second one is the natural generalization to the discrete variables using the set of neighbors. We introduced the last one to explore around some promising points in the set of neighbors and strengthen the optimality conditions achieved by the limit points.

Polling with respect to the continuous variables requires the use of positive bases, or at least positive spanning sets, on $\Re^{n^c}$. A positive basis is a set of nonzero vectors in $\Re^{n^c}$ whose nonnegative linear combinations span $\Re^{n^c}$, but no proper subset does so. Each positive basis contains at least $n^c + 1$ and at most $2n^c$ vectors. These are referred to as minimal and maximal positive bases (see Davis [4] for characterization of positive bases). We use the following key property of positive spanning sets. For any nonzero vector a in $\Re^{n^c}$ and positive spanning set B on $\Re^{n^c}$, there exists a vector b of B such that

$$(3.1) \qquad\qquad\qquad\qquad a^T b < 0.$$

Let $\mathcal{B}$ be a finite set of positive spanning sets on $\Re^{n^c}$ such that every column b of every positive spanning set of $\mathcal{B}$ is of the form $P_j z$ for some $z \in Z^{n^c}$ and $1 \leq j \leq j_{max}$. The pattern matrices P_j are the same ones used to construct the lattices in section 3.1. In a way similar to [9], we assume that at least one positive spanning set of $\mathcal{B}$ is a maximal positive basis whose columns may be partitioned in a way to form two nonsingular diagonal $n^c \times n^c$ matrices. Let $\overline{\mathcal{B}} \subset \mathcal{B}$ be the set of all these bases. Conceptually, the set $\mathcal{B}$ is fixed throughout all iterations, but it may evolve as the solution process proceeds as long as it remains finite.

The poll points with respect to the continuous variables are obtained by scaling a basis $\mathcal{B}$ by the mesh size parameter as follows: At iteration k, for any mesh point x, define $\mathcal{N}^c(x)$, the *mesh neighborhood of the continuous variables around x*, to be

$$(3.2) \qquad\qquad \mathcal{N}^c(x) = \{ x + \Delta_k(b, 0) \in \Omega : b \in B_k(x) \}$$

for some positive spanning set $B_k(x) \in \mathcal{B}$ that depends on both the iteration number k and the point x. Moreover, in order to avoid the infeasibility problem described

in [9], we require that if one of the components of the current iterate x_k^c is within a tolerance parameter $\varepsilon > 0$ of either its lower or upper bound, then the positive spanning set for this iteration must be chosen in $\overline{\mathcal{B}}$.

This definition ensures that the mesh neighborhood $\mathcal{N}^c(x_k)$ is a subset of the current mesh $\mathcal{M}_k$. Moreover, $\mathcal{N}^c(x_k)$ is constructed using a single positive spanning set chosen from a finite set, and thus there are only a finite number of possible ways to define mesh neighborhoods.

The motivation for introducing positive spanning sets for the continuous variables is that if the gradient $\nabla^c f$ of the function f with respect to the continuous variables is nonzero, then at least one vector of the set defines a descent direction. The original work of Torczon [12] uses a maximal positive basis for unconstrained optimization. It was later generalized in Lewis and Torczon [10] to any positive basis, thus reducing the maximum number of points in the polling set from $2n^c$ to $n^c + 1$. However, for bound constrained optimization, they show that taking the maximal positive bases generated by the coordinate directions guarantees finding a feasible descent direction (if there is one) even on the boundary of the feasible region.

The discrete stage of the POLL step depends on the set of neighbors defined by the user (as in section 2.1). In order to allow for varying of the definition of the set of neighbors for a finite number of iterations, we define the *current set of neighbors* $\mathcal{N}_k$ to be such that $\mathcal{N}_k$ differs from $\mathcal{N}(x_k)$ at most at a finite number of iterations k. This flexibility allows finitely many redefinitions of $\mathcal{N}_k$ to allow the user another knob to adjust the cost of a POLL step (see section 3.3) and the likely quality of the limit point.

If none of the above-mentioned polling points (i.e., those in the mesh neighborhood $\mathcal{N}^c(x_k)$ and in the set of neighbors $\mathcal{N}_k$) yields decrease in the objective function value, a third stage might be required before declaring the iteration unsuccessful. This stage is triggered by the last of our user controlled knobs $\xi > 0$ to pay more for a most likely better final function value. An EXTENDED POLL step must be conducted around each point of the set of neighbors $\mathcal{N}_k$ of x_k at which the function value, even though it is larger, is within ξ of $f(x_k)$. Intuitively, ξ represents a tolerance which is such that if a discrete neighbor y in $\mathcal{N}_k$ provides such a near function value, then the user wishes us to poll in the continuous variables around y since this may produce a new best solution. Our convergence analysis is independent of the value of ξ, but intuitively a larger ξ means extended polling will be carried out at more iterations, which may cost more function evaluations, but should give a better local minimizer. Of course, it would be simple to construct examples showing the opposite behavior, but our thermal example shows how this can work.

More precisely, consider any point y in the set of neighbors $\mathcal{N}_k$. (The variable y should be indexed with the iteration number k and with respect to the set of neighbors $\mathcal{N}_k$, but this would obscure the notation.) In the case where $f(y) > f(x_k) + \xi$ or $f(y) \leq f(v)$ for all v in $\mathcal{N}^c(y)$, the POLL step need not be extended and so we set the index J to 0. In all other cases, y^0 is set to y and for $j \geq 1$ we select the feasible point y^j in the mesh neighborhood $\mathcal{N}^c(y^{j-1})$ iteratively so that $f(y^j) < f(y^{j-1})$ until it is no longer possible (or until $f(y^j) < f(x_k)$ in which case iteration k is successful and x_{k+1} is set to y^j). It follows that the last point (whose index is denoted by J) satisfies $f(y^J) \leq f(v)$ for all v in $\mathcal{N}^c(y^J)$. Define z to be the endpoint y^J of the EXTENDED POLL step. Keep in mind that z depends on the iteration number k and on the neighbor y in $\mathcal{N}_k$. These trial points are illustrated in Figure 3.1, where they are indexed with the iteration number.
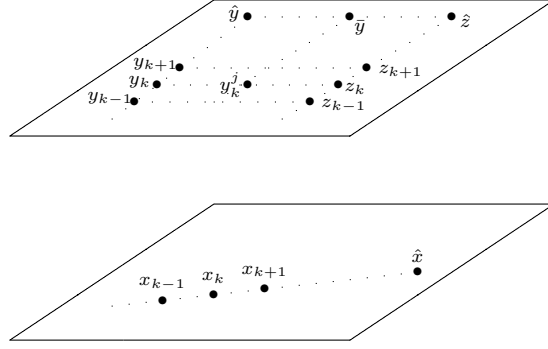
FIG. 3.1. *Limit points of the iterates and the extended poll points.*

With this construction, the function values $f(y) = f(y^0), f(y^1), \ldots, f(y^J) = f(z)$ are monotonically decreasing unless $y = z$. Only at the endpoint z is the function required to be evaluated at every point of its mesh neighborhood $\mathcal{N}^c(z)$. Observe that the index J may be 0, in which case $y = z$. This happens either when $f(y) > f(x_k) + \xi$ or when $f(y) \leq f(v)$ for all v in $\mathcal{N}^c(y)$. The index J is finite since all generated points y^j are distinct and belong to the mesh $\mathcal{M}_k$ intersected with the compact level set $L_\Omega(x_0)$ (see assumption (A1) in section 3.3).

The set of all points visited by the POLL step at iteration k is denoted X_k^ξ and may be explicitly written as

$$X_k^\xi = \mathcal{N}^c(x_k) \;\bigcup\; \mathcal{N}_k \bigcup_{\substack{y \in \mathcal{N}_k: \\ f(x_k) \leq f(y) \leq f(x_k) + \xi}} \mathcal{E}(y),$$

where $\mathcal{E}(y)$ is the extended poll set, which contains $\{y^1, y^2, \ldots, y^J\} \cup \mathcal{N}^c(z)$ as well as some points of $\mathcal{N}(y^j)$ for some j's in $\{0, 1, \ldots, J-1\}$.

Figure 3.2 illustrates an instance in which there are two continuous variables and one discrete variable. The set of neighbors of the iterate x_k is assumed to be $\mathcal{N}_k = \mathcal{N}(x_k) = \{x_k, y_1^0, y_2^0\}$, where the subscripts 1 and 2 are added to distinguish the points in the set of neighbors $\mathcal{N}(x_k)$. (Note that the points in $\mathcal{N}_k$ do not have the same values for the continuous variables.) The iterate x_k is a local minimizer of the function f on X_k^ξ if $f(x_k)$ is less than or equal to the function value evaluated at all points in balls around x_k, y_1^0, and y_2^0. The letters a to l in the figure represent mesh neighborhoods of the continuous variables

$$\mathcal{N}^c(y_1^0) = \{d, e, y_1^1\}, \ \mathcal{N}^c(y_1^1) = \{a, b, c\}, \ \mathcal{N}^c(x_k) = \{f, g, h\}, \ \text{and} \ \mathcal{N}^c(y_2^0) = \{i, j, k, l\}.$$

In this example, since $f(x_k) \leq f(y_1^0) < f(x_k) + \xi < f(y_2^0)$ the poll set X_k^ξ contains points in $\mathcal{N}^c(x_k)$ and $\mathcal{N}^c(y_1^0)$ (among others). Assuming that $f(y_1^1) < f(y_1^0)$ but $f(a) \geq f(y_1^1)$, $f(b) \geq f(y_1^1)$ and $f(c) \geq f(y_1^1)$ lead to the poll set $X_k^\xi = \{f, g, h\} \cup \{x_k, y_1^0, y_2^0\} \cup \mathcal{E}(y)$, where $\mathcal{E}(y) = \{y_1^1\} \cup \{a, b, c\}$. Note that, depending on the order in which the function values are evaluated, it is possible that the extended poll set also contains d or e.

Using the above notation, we can now present the GMVP algorithm.

3.3. The GMVP search algorithm. Our presentation of the pattern search algorithm is closer to that of Booker et al. [2] than to that of Torczon [12]. Consider
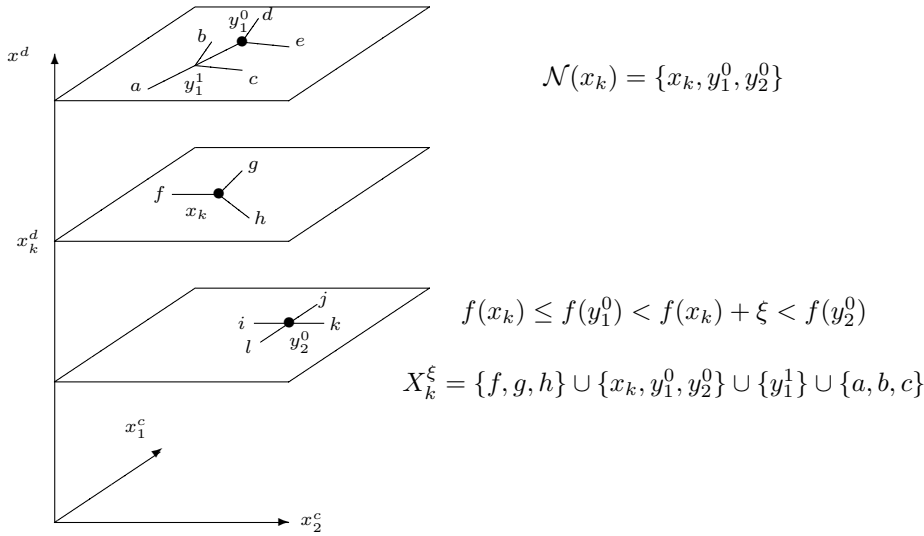
$$\mathcal{N}(x_k) = \{x_k, y_1^0, y_2^0\}$$

$$f(x_k) \leq f(y_1^0) < f(x_k) + \xi < f(y_2^0)$$

$$X_k^\xi = \{f, g, h\} \cup \{x_k, y_1^0, y_2^0\} \cup \{y_1^1\} \cup \{a, b, c\}$$

FIG. 3.2. *Construction of the current mesh neighborhood X_k^ξ around x_k.*

the given initial mesh $M_0 \subset \Omega$ with mesh size parameter Δ_0 and initial point x_0 of M_0. Also, let $\xi > 0$ be the objective function change tolerance used to trigger extended polling in the construction of the poll set. Recall that if $f(x_k) \leq f(y) \leq f(x_k) + \xi$ for some y in the set of neighbors $\mathcal{N}_k$, then the polling step must be extended around y.

Throughout the paper, the following assumptions are made:

(A1) The level set $L_\Omega(x_0) = \{x \in \Omega : f(x) \leq f(x_0) + \xi\}$ is compact.

(A2) f is continuously differentiable over a neighborhood of $L_\Omega(x_0)$ when variables in $\mathcal{Z}^{n^d}$ are fixed, i.e., for any $x^d \in \Omega^d$ the function $x^c \mapsto f(x^c, x^d)$ is continuously differentiable in a neighborhood of $\{x^c : (x^c, x^d) \in L_\Omega(x_0)\}$.

At any iteration $k \geq 0$, the general rules for choosing x_{k+1} in the current mesh $\mathcal{M}_k$ and obtaining the next mesh size parameter Δ_{k+1} are as follows.

GENERALIZED MIXED VARIABLE PATTERN SEARCH ALGORITHM (GMVP).

1. SEARCH **step (in current mesh).** Employ some finite strategy to obtain an $x_{k+1} \in \mathcal{M}_k$ satisfying $f(x_{k+1}) < f(x_k)$. If such an x_{k+1} is found, declare the SEARCH step (as well as the iteration) successful, then expand the mesh at step 3.

2. POLL **step.** This step is reached only if the SEARCH step is unsuccessful. If $f(x_k) \leq f(x)$ for every x in the poll set X_k^ξ, then declare the POLL step (as well as the iteration) unsuccessful and shrink the mesh at step 4. Otherwise, choose $x_{k+1} \in X_k^\xi$ to be a point such that $f(x_{k+1}) < f(x_k)$, declare the POLL step (as well as the iteration) successful, and expand the mesh at step 3.

3. Mesh expansion (at successful iterations). Let $\Delta_{k+1} = \tau^{m_k^+}\Delta_k$ (for $\tau^{m_k^+} \geq 1$ defined below). Increase k, and initiate the next iteration at step 1.

4. Mesh reduction (at unsuccessful iterations). Set x_{k+1} to x_k and let $\Delta_{k+1} = \tau^{m_k^-}\Delta_k$ (for $0 < \tau^{m_k^-} < 1$ defined below). Increase k, and initiate the next iteration at step 1. □

In the SEARCH and POLL steps, the number of candidate points among which the next iterate can be chosen is finite, since they must belong to the intersection of the enumerable current mesh and the compact set $L_\Omega(x_0)$.

The parameters in the last two steps are the rational number $\tau > 1$ and the

integers (whose absolute values are bounded above by a constant $m_{max} \geq 1$) $m_k^+ \geq 0$ and $m_k^- \leq -1$. In [12], the mesh reduction parameter m_k^- was fixed for all $k \geq 0$. This restriction is relaxed here without affecting the convergence results. We plan to exploit this flexibility in subsequent work to increase the practical convergence speed.

The conditions on these parameters imply the *simple decrease property* used throughout the growing literature on GPS methods: iteration k is successful if and only if $f(x_{k+1}) < f(x_k)$, if and only if $\Delta_{k+1} \geq \Delta_k$, and if and only if $x_{k+1} \neq x_k$. Another important implication of the parameters' definition is that if the iteration k is unsuccessful, then $f(x_k) \leq f(v)$ for all $v \in X_k^{\xi}$ and thus $f(x_k) \leq f(y)$ for all $y \in \mathcal{N}^c(x_k)$ and, whenever $f(y) \leq f(x_k) + \xi$ for some $y \in \mathcal{N}_k$, then $f(z) \leq f(v)$ for all $v \in \mathcal{N}^c(z)$, where z is the end point of the extended poll step initiated at y. Moreover, Δ_{k+1} is obtained by multiplying Δ_k by a finite positive or negative integer power of τ. Therefore, for any $k \geq 0$, we can write

$$(3.3) \qquad\qquad \Delta_k = \Delta_0 \tau^{r_k}$$

for some r_k belonging to $\mathcal{Z}$.

Notice that the cost of the POLL step is expected to depend on both ξ and the definition of the set of neighbors $\mathcal{N}$. Thus, the user can pay more function evaluations for a stronger local solution by defining a larger ξ or a larger neighborhood $\mathcal{N}$. Another way that the user can likely improve the quality of the solution is through the SEARCH step. In that step, knowledge of the problem and/or favorite heuristics can be used to improve the solution. The user can also try to evaluate the function at various places in the variable space and design interpolary models or use surrogate functions (as discussed in Booker et al. [2]). The search strategy, which aims at finding the best solution on the current mesh, can be as sophisticated as one wants, but may increase the number of function evaluations.

4. Proof of convergence. This section contains the convergence proof for the GMVP algorithm. We start by studying the behavior of the mesh size parameter Δ_k. The first important result, due to Torczon for the continuous case, is $\lim\inf_{k \to +\infty} \Delta_k = 0$ which implies the existence of a subsequence of mesh size parameters that converges to zero. A key to our simpler proof is to conclude from this that there is an infinite number of unsuccessful iterations. We analyze converging subsequences of unsuccessful iterates whose mesh size parameters converge to zero. We show that any limit point of such a subsequence satisfies the optimality conditions (2.1)–(2.4). By focusing on unsuccessful iterations, the result for the continuous variables is shown using a much shorter proof than in [12] and [9]. We also present a stronger version of the algorithm that yields a stronger result, i.e., the optimality condition (2.5).

Thus, when we consider the analysis of this class of algorithms, it is the sequences of unsuccessful iterates that we show converge. The terminology *successful/unsuccessful* that made perfect sense in explaining the algorithm suddenly jars us because of the pejorative connotation of the word "unsuccessful." In fact, an iteration is unsuccessful because the corresponding iterate is a local mesh minimizer, and so the discrete resolution of the domain must be refined by reducing Δ_k before we can expect to proceed downhill. Likewise, an iteration is successful because it moves us toward finding a local mesh minimizer. Thus, in a sense, *successful/unsuccessful* could be replaced by *inner/outer* or *minor/major* as labels for the two types of iterations. We hesitate to suggest such a change to well-established terminology too quickly, but we hope this short warning discussion will alleviate the confusion readers have expressed to us.

4.1. Behavior of the mesh size. To show that there is a subsequence of mesh size parameters Δ_k that converges to zero, we first show that these parameters are bounded above by a constant, independent of the iteration number k.

LEMMA 4.1. *There exists a positive integer r^u such that $\Delta_k \leq \Delta_0 \tau^{r^u}$ for any $k \geq 0$.*

Proof. Let Δ be a mesh size parameter large enough so that the union of lattices $\mathcal{M}(\Delta)$ intersects the compact feasible level set $\{x^c : x \in L_\Omega(x_0)\}$ only at the translation parameter x_k^c; i.e., for any $1 \leq j \leq j_{max}, z \in \mathcal{Z}^{n^c}$, and x in $L_\Omega(x_0)$ the solution $x^c + \Delta P_j z$ does not belong to the projection of $L_\Omega(x_0)$ on the continuous variables space unless it equals x_k^c. Therefore, if at iteration k the mesh size parameter Δ_k is greater than or equal to Δ, then

$$\mathcal{M}_k \cap L_\Omega(x_0) \subset \{x_k^c\} \times \Omega^d.$$

Moreover, only a finite number of iterations will follow before the mesh size parameter drops below Δ. Indeed, the continuous part of all these iterates will necessarily be equal to x_k^c, and the discrete part of these iterates can take only a finite number of values because the set $L_\Omega(x_0)$ is bounded. Let d_{max} be the total number of distinct values that the discrete variables may take in $L_\Omega(x_0)$. Therefore, there will be no more than d_{max} successful iterations before the mesh size parameter goes below Δ.

Recall that the expansion mesh size control parameter is bounded above by $\tau^{m_{max}}$. Let r^u be a large enough integer so that $\Delta_0 \tau^{r^u} \geq \Delta(\tau^{m_{max}})^{d_{max}}$. It follows that the mesh size parameter at any iteration will never exceed $\Delta_0 \tau^{r^u}$. □

We now study the convergence behavior of the mesh size parameter. The proof of this result is essentially identical to that of Torczon [12] despite the presence of discrete variables.

THEOREM 4.2. *The mesh size parameters satisfy $\liminf_{k \to +\infty} \Delta_k = 0$.*

Proof. Suppose by way of contradiction that there exists a negative integer r^ℓ such that $0 < \Delta_0 \tau^{r^\ell} \leq \Delta_k$ for all $k \geq 0$. Equation (3.3) states that for every $k \geq 0$ there is an $r_k \in \mathcal{Z}$ such that $\Delta_k = \tau^{r_k} \Delta_0$. Combining this with Lemma 4.1 implies that for any $k \geq 0$, r_k takes its value among the integers of the bounded interval $[r^\ell, r^u]$. Therefore, r_k and Δ_k can take only a finite number of values for all $k \geq 0$.

For any k, the continuous part of the next iterate x_{k+1}^c belongs to a lattice and can be written $x_k^c + \Delta_k P_{j_k} z_k$ for some $z_k \in \mathcal{Z}^{n^c}$ and $1 \leq j_k \leq j_{max}$. By substituting $\Delta_k = \Delta_0 \tau^{r_k}$ and $P_j = G^B G_{j_k}$, it follows that for any integer N

$$x_N^c = x_0^c + \sum_{k=1}^{N-1} \Delta_k P_{j_k} z_k$$

$$= x_0^c + \Delta_0 G^B \sum_{k=1}^{N-1} \tau^{r_k} G_{j_k} z_k \;=\; x_0^c + \frac{p^{r^\ell}}{q^{r^u}} \Delta_0 G^B \sum_{k=1}^{N-1} p^{r_k - r^\ell} q^{r^u - r_k} G_{j_k} z_k,$$

where p and q are relatively prime integers satisfying $\tau = \frac{p}{q}$.

Since for any k the term $p^{r_k - r^\ell} q^{r^u - r_k} G_{j_k} z_k$ appearing in this last sum is an integer, it follows that the continuous part of all iterates lies on the translated integer lattice generated by x_0^c and the columns of $\frac{p^{r^\ell}}{q^{r^u}} \Delta_0 G^B$. Moreover, the discrete part of all iterates also lies on the integer lattice $\mathcal{Z}^{n^d}$.

Therefore, since all iterates belong to the compact set $L_\Omega(x_0)$, it follows that there is only a finite number of different iterates, and thus one of them must be

visited infinitely many times. Simple decrease ensures that the mesh size parameters converge to zero, which is a contradiction. □

4.2. The main results. Lewis and Torczon [9] show that condition (1.2) holds; i.e., there exists a limit point $\hat{x}$ of the sequence of iterates for which $s^T \nabla f(\hat{x}) \geq 0$ for any feasible direction s. Through a shorter proof, we show a stronger result. We show the existence of a limit point $\hat{x}$ of the sequence of *unsuccessful* iterates that satisfies (2.1) and is a local optimizer with respect to the set of neighbors $\mathcal{N}(\hat{x})$ in the sense of conditions (2.2), (2.3), and (2.4). Recall that iteration k is unsuccessful if and only if $x_{k+1} = x_k$, which is equivalent to $\Delta_{k+1} < \Delta_k$. Thus, the number of unsuccessful iterations is infinite since $\liminf_{k \to +\infty} \Delta_k = 0$ by Theorem 4.2.

Consider the indices of the unsuccessful iterations whose corresponding mesh size parameters go to zero. For any limit point of such a subsequence, there is an iterate x_k arbitrarily close to it for which no trial point of the poll set X_k^ξ yields a decrease in the objective function value. The following result details properties of a limit point $\hat{x}$ of the subsequence of unsuccessful iterations whose mesh size parameters converge to zero (Figure 3.1 depicts this result).

PROPOSITION 4.3. *There is a point $\hat{x} \in L_\Omega(x_0)$ and a subset of indices of unsuccessful iterates $K \subset \{k : x_{k+1} = x_k\}$ such that*

$$\lim_{k \in K} \Delta_k = 0, \qquad \lim_{k \in K} x_k = \hat{x}, \quad and \quad \mathcal{N}_k = \mathcal{N}(x_k) \ \forall k \in K.$$

Moreover, if $\hat{y}$ belongs to the set of neighbors $\mathcal{N}(\hat{x})$, then there exists a $\hat{z} = (\hat{z}^c, \hat{y}^d) \in \Omega$ such that

$$\lim_{k \in K} y_k = \hat{y} \quad and \quad \lim_{k \in K} z_k = \hat{z},$$

where $z_k \in \Omega$ is the endpoint of the EXTENDED POLL *step initiated at $y_k \in \mathcal{N}(x_k)$ at iteration $k \in K$.*

Proof. Theorem 4.2 guarantees that $\liminf_{k \to +\infty} \Delta_k = 0$; thus there is an infinite subset of indices of unsuccessful iterations $K' \subset \{k : x_{k+1} = x_k\} = \{k : \Delta_{k+1} < \Delta_k\}$ such that the subsequence $\{\Delta_k\}_{k \in K'}$ converges to zero.

Since all iterates x_k lie in the compact set $L_\Omega(x_0)$, we can extract an infinite subset of indices $K'' \subset K'$ such that the subsequence $\{x_k\}_{k \in K''}$ converges. Let $\hat{x}$ in $L_\Omega(x_0)$ be the limit point of such a subsequence. Moreover, since $\mathcal{N}_k$ differs from $\mathcal{N}(x_k)$ at most at a finite number of iterates, we may assume, without any loss of generality, that $x_k^d = \hat{x}^d$ for all $x_k \in K''$.

Let $\hat{y} \in \Omega$ be a point of the set of neighbors $\mathcal{N}(\hat{x})$. Recall that we assumed in section 2.1 a notion of continuity of the sets of neighbors. Therefore, $\hat{y}$ is a limit point of a subsequence $y_k \in \mathcal{N}_k$. Let $\hat{z} \in \Omega$ be a limit point of the sequence $z_k \in \Omega$ of endpoints of the EXTENDED POLL step initiated at y_k. By definition, the endpoint z_k is equal to y_k in the case that the EXTENDED POLL step is not required.

Choose $K \subset K''$ to be such that both $\{y_k\}_{k \in K}$ converges to $\hat{y}$ and $\{z_k\}_{k \in K}$ is convergent (let $\hat{z}$ denote the limit point). □

Torczon [12] observes that setting the mesh size increase parameter m_k^+ to zero (in the mesh expansion step of the GPS algorithm) ensures that $\lim_{k \to \infty} \Delta_k = 0$. Thus the mesh is never expanded: at unsuccessful iterations, the mesh size parameter Δ_{k+1} is set to be equal to Δ_k. The same observation holds for our algorithm. It follows that in this case, all the convergence results below hold for *every* limit point of the sequence of unsuccessful iterates.

For the rest of this section, we assume that $\hat{x}$ and K satisfy the conditions of Proposition 4.3. The main results can now be proved. We first show that $\hat{x}$ is a local optimal solution with respect to the set of neighbors $\mathcal{N}(\hat{x}) \subset \Omega$.

THEOREM 4.4. *The limit point $\hat{x}$ satisfies $f(\hat{x}) \leq f(\hat{y})$ for all $\hat{y} \in \mathcal{N}(\hat{x})$.*

Proof. Suppose by way of contradiction that there is a $\hat{y} \in \mathcal{N}(\hat{x})$ such that $f(\hat{x}) > f(\hat{y})$. Continuity of the function f with respect to the continuous variables guarantees the existence of an $\epsilon > 0$ such that if v belongs to the ball $B(\epsilon, \hat{y})$ of radius ϵ centered at $\hat{y}$, then $f(v) < f(\hat{x})$.

Proposition 4.3 guarantees that the subsequences $\{x_k\}_{k \in K}$ and $\{y_k\}_{k \in K}$ (where $y_k \in \mathcal{N}(x_k)$), respectively, converge to $\hat{x}$ and $\hat{y}$. We required in section 2.1 that the set $\mathcal{N}(x_k)$ converges for $k \in K$ to $\mathcal{N}(\hat{x})$ in the sense that if $k \in K$ is large enough, then there exists a $y_k \in \mathcal{N}(x_k)$ such that $y_k \in B(\epsilon, \hat{y})$.

Therefore, there exists an iteration $k \in K$ such that y_k belongs to $\mathcal{N}_k \cap B(\epsilon, \hat{y})$ and satisfies $f(y_k) < f(\hat{x}) \leq f(x_k)$. It follows that the iteration is successful, contradicting the fact that k belongs to $K \subset \{k : x_{k+1} = x_k\}$. □

In the case where the inequality in Theorem 4.4 is strict, i.e., $f(\hat{x}) < f(\hat{y})$, then the notion of local optimality for mixed integer programming presented in section 2.1 is verified: There exists an $\epsilon > 0$ such that $f(\hat{x}) \leq f(v)$ for any v in a ball of radius ϵ around $\hat{y}$. This follows from the continuity of the function f with respect to the continuous variables.

Next, we study the gradient of the function f with respect to the continuous variables at the limit point $\hat{x}$. The proof of Theorem 4.7 for the continuous case is much shorter than the original one of Torczon [12]. Its proof, as well as that for the EXTENDED POLL step (Theorem 4.8), relies on the following two lemmas.

The first lemma shows that the gradient is zero in the strict interior of the boundary of the feasible region. These results concern points around which polling is unsuccessful (i.e., p_k in the lemmas will take the value of x_k or z_k).

LEMMA 4.5. *Let $\{p_k\}_{k \in K}$ be a subsequence of unsuccessful poll points and let $\hat{p}$ be a limit point. If the continuous part of the limit point $\hat{p}$ is in the strict interior of the feasible region $[\ell, u]$, then $\nabla^c f(\hat{p}) = 0$.*

Proof. Since $k \in K$ and $\hat{p}$ is strictly feasible, it follows that $\{\Delta_k\}_{k \in K}$ goes to zero and $\{p_k + \Delta_k(b, 0) : b \in B_k(p_k)\}$ is contained in Ω for k large enough. Equation (3.2) and the mean value theorem imply that

$$
\begin{aligned}
f(p_k) &\leq \min_{v \in \mathcal{N}^c(p_k)} f(v) = \min_{v \in \{p_k + \Delta_k(b,0) : b \in B_k(p_k)\}} f(v) \\
&= \min_{b \in B_k(p_k)} f(p_k + \Delta_k(b, 0)) \\
&= \min_{b \in B_k(p_k)} f(p_k) + \Delta_k b^T \nabla^c f(p_k + \alpha_k^b \Delta_k(b, 0)) \\
&= f(p_k) + \Delta_k \min_{b \in B_k(p_k)} b^T \nabla^c f(p_k + \alpha_k^b \Delta_k(b, 0))
\end{aligned}
$$

for some $\alpha_k^b \in [0, 1]$ that depends on both the positive basis vector b and iteration number k. Therefore

$$
0 \leq \min_{b \in B_k(p_k)} b^T \nabla^c f(p_k + \alpha_k^b \Delta_k(b, 0)).
$$

Taking the limit for $k \to \infty$ yields $0 \leq \min_{b \in B} b^T \nabla^c f(\hat{p})$ for at least one positive spanning set B of the finite set $\mathcal{B}$ since f is assumed to be continuously differentiable. The positive spanning set property (3.1) guarantees that $\nabla^c f(\hat{p}) = 0$. □

The second lemma shows that there are no descent directions for points on the boundary of the feasible region.

LEMMA 4.6. *Let $\{p_k\}_{k \in K}$ be a subsequence of unsuccessful poll points and let $\hat{p}$ be a limit point. If the continuous part of the limit point $\hat{p}$ is on the boundary of the feasible region $[\ell, u]$, then $(p^c - \hat{p}^c)^T \nabla^c f(\hat{p}) \geq 0$ for any feasible $(p^c, \hat{p}^d)$.*

Proof. If the continuous part of p_k is within ε of the boundary of the feasible region $[\ell, u]$, then $B_k(p_k)$ is a maximal positive basis belonging to $\overline{\mathcal{B}}$ constructed from two diagonal matrices (see the discussion preceding the definition of the mesh neighborhood (3.2)).

As in Lemma 4.5, but only for the feasible positive basis directions b of $B_k(p_k)$, we have that

$$0 \ \leq \ b^T \nabla^c f(p_k + \alpha_k^b \Delta_k(b, 0)).$$

The result follows since any feasible direction $(p^c - \hat{p}^c)$ at $\hat{p}$ is a convex combination of some feasible positive basis directions.   □

We can now state our first main result.

THEOREM 4.7. *The limit point $\hat{x}$ satisfies $(x^c - \hat{x}^c)^T \nabla^c f(\hat{x}) \geq 0$ for any feasible $(x^c, \hat{x}^d)$.*

Proof. The result follows directly from Lemmas 4.5 and 4.6, where x_k plays the role of p_k, and from the results on the sequence $\{x_k\}_{k \in K}$ of Proposition 4.3.   □

Audet [1] shows through a small example containing two continuous variables and no discrete ones that, in the unconstrained case, this result cannot be strengthened to $\lim_{k \to \infty} \|\nabla^c f(x_k)\| = 0$ since there may be a limit point whose gradient is nonzero. It is also shown there that no second-order optimality conditions can be guaranteed, which is as it should be for an algorithm that uses only function values and no derivatives.

The following result shows that the gradient norm at the endpoints of the EX-TENDED POLL converges to zero for $k \in K$.

THEOREM 4.8. *The limit point $\hat{x}$, and any point $\hat{y}$ in the set of neighbors $\mathcal{N}(\hat{x})$ satisfying $f(\hat{y}) < f(\hat{x}) + \xi$, are such that $(z^c - \hat{z}^c)^T \nabla^c f(\hat{z}) \ \geq \ 0$ for any feasible $(z^c, \hat{z}^d)$, where $\hat{z}$ is any limit point of the EXTENDED POLL endpoints.*

Proof. The result follows directly from Lemmas 4.5 and 4.6, where z_k plays the role of p_k, and from the results on the sequence $\{z_k\}_{k \in K}$ of Proposition 4.3.   □

The next result shows that the function is constant at an infinite number of intermediate points between $\hat{y}$ and the endpoint $\hat{z}$ whenever $f(\hat{y}) = f(\hat{x})$. In order to show this result, we add the index k here to avoid confusion. The extended poll points at iteration k initiated at y_k are denoted $y_k^0 = y_k, y_k^1, \ldots, y_k^J = z_k$, where the index J depends on both k and y_k. Again, this is illustrated in Figure 3.1.

PROPOSITION 4.9. *The limit point $\hat{x}$, and any $\hat{y} \in \mathcal{N}(\hat{x})$ satisfying $f(\hat{y}) = f(\hat{x})$, are such that any limit point $\bar{y}$ of the sequence of EXTENDED POLL points $\{y_k^j\}$ satisfies $f(\bar{y}) = f(\hat{x})$. Moreover, if $\hat{y} \neq \hat{z}$, then there are infinitely many of these limit points.*

Proof. Let $\hat{y}$ in $\mathcal{N}(\hat{x})$ be such that $f(\hat{y}) = f(\hat{x})$. Let $\bar{y}$ be a limit point distinct from $\hat{y}$ and $\hat{z}$ of the sequence of EXTENDED POLL points $\{y_k^j\}$.

Since $f(\hat{x}) \leq f(y_k^{j+1}) < f(y_k^j)$ for $j = 0, 1, \ldots, J$ and since the subsequence $\{f(y_k^0)\}_{k \in K}$ converges to $f(\hat{x})$, we conclude that $f(\hat{x}) = f(\bar{y})$.

To show the second part of the result, we first let $d = \|\hat{y} - \hat{z}\|$ be the nonzero distance between $\hat{y}$ and $\hat{z}$. This makes sense because both share the same discrete components. Second, for any scalar p in the open interval $]0, d[$, we define the set

$$Y_p = \left\{ y_k^j : \ k \in K, \ j \in \{0, 1, \ldots, J\}, \ \|y_k^j - \hat{y}\| \leq p, \ \|y_k^{j+1} - \hat{y}\| > p\| \right\}.$$

Since $y_k^0 = y_k \to \hat{y}$ and $y_k^J = z_k \to \hat{z}$, it follows that the set Y_p contains infinitely many points for any p in $]0,1[$. Any limit point $\bar{y}_p$ of Y_p satisfies $\|\bar{y}_p - \hat{y}\| = p$ since Δ_k converges to 0 (in K) and y_k^{j+1} is equal to $y_k^j + \Delta_k b_k^j$ for some vector b_k^j of the basis $B_k(y_k^j)$ of the finite set $\mathcal{B}$. Therefore if $p \neq q$, then $\bar{y}_p \neq \bar{y}_q$ and the result follows.   □

4.3. Stronger results. Theorem 4.8 may be strengthened under the following (more expensive) version of extended polling.

STRONG EXTENDED POLL STEP:

- $y_k^{j+1} \in \arg\min_{y \in \mathcal{N}^c(y_k^j)} f(y)$ for a given y_k^0 and $j = 0, 1, \ldots, J$ at iteration k;
- the same positive basis in $\overline{\mathcal{B}}$ must be used throughout the strong extended poll step.

This requires performing a complete EXTENDED POLL step; i.e., y_k^{j+1} is chosen only after evaluating the function value at all feasible points of the continuous mesh neighborhood around y_k^j and retaining the one that yields the smallest value (ties are broken arbitrarily).

This also means that the matrix $B_k(y_k^j)$ in $\mathcal{N}^c(y_k^j) = \{y_k^j + \Delta_k(b, 0) \in \Omega : b \in B_k(y_k^j)\}$ does not depend on the index j; it can, however, vary with the iteration number k. This positive basis is maximal and constructed from diagonal matrices. This is to make sure that the basis directions are correctly chosen in the event that the extended poll iterates approach the boundary of the feasible region.

The following result bounds the decrease in the objective function value under precise conditions. We will denote by b_k^j the vector of the positive basis used by the EXTENDED POLL step at the point y_k^j for some $j < J$. The next point is therefore $y_k^{j+1} = y_k^j + \Delta_k b_k^j$.

LEMMA 4.10. *Let $\hat{y} \in \mathcal{N}(\hat{x})$. For any $\eta < 0$, there exist $\delta > 0$ and $\sigma > 0$, both independent of the iteration number k, such that all extended poll iterates y_k^j for which $j < J$, $\Delta_k < \delta$, $y_k^{jd} = \hat{y}^d$ and for which $(b_k^j)^T \nabla^c f(y_k^j) \leq \frac{\eta}{3}$ also satisfy $f(y_k^j) - f(y_k^{j+1}) > \sigma \|y_k^j - y_k^{j+1}\|$.*

Proof. Let $\eta < 0$ be given. Continuous differentiability of the function f with respect to the continuous variables over a neighborhood of the compact set $L_\Omega(x_0)$ implies the existence of $\delta > 0$ such that any $y \in L_\Omega(x_0)$ and $w \in \Omega$ that satisfy $\|w - y\| < \delta \times \max\{\|b\| : b \in B \in \mathcal{B}\}$ also satisfy $\|b^T(\nabla^c f(w) - \nabla^c f(y))\| < \frac{|\eta|}{6}$ for each feasible direction $b \in B \in \mathcal{B}$ at y and, in particular, $b^T \nabla^c f(w) < b^T \nabla^c f(y) - \frac{\eta}{6}$.

Let $\hat{y} \in \mathcal{N}(\hat{x})$ and consider the extended poll iterate $y_k^{j+1} = y_k^j + \Delta_k b_k^j$, where $y_k^{jd} = \hat{y}^d$. Applying the mean value theorem yields

$$(4.1) \qquad f(y_k^{j+1}) = f(y_k^j) + \Delta_k (b_k^j)^T \nabla^c f(w_k^j)$$

for some $w_k^j = y_k^j + \alpha_k^j \Delta_k b_k^j$, where α_k^j is a real number in the interval $[0, 1]$.

Assume that y_k^j satisfies $(b_k^j)^T \nabla^c f(y_k^j) \leq \frac{\eta}{3}$; if no such point exists, then the result is trivial. Observe that

$$(4.2) \qquad \Delta_k = \frac{\|y_k^j - y_k^{j+1}\|}{\|b_k^j\|} \geq \frac{\|y_k^j - y_k^{j+1}\|}{\max\{\|b\| : b \in B \in \mathcal{B}\}}.$$

Moreover, if $\Delta_k < \delta$, then $w_k^j \in \Omega$ is within $\delta \times \max\{\|b\| : b \in B \in \mathcal{B}\}$ of $y_k^j \in L_\Omega(x_0)$ since

$$\|w_k^j - y_k^j\| = \alpha_k^j \Delta_k \|b_k^j\| < 1 \times \delta \|b_k^j\| \leq \delta \times \max\{\|b\| : b \in B \in \mathcal{B}\}.$$

Therefore, $(b_k^j)^T \nabla^c f(w_k^j) < (b_k^j)^T \nabla^c f(y_k^j) - \frac{\eta}{6} \leq \frac{\eta}{3} - \frac{\eta}{6} = \frac{\eta}{6}$, and it follows by (4.1) and (4.2) that

$$f(y_k^{j+1}) - f(y_k^j) < \Delta_k \frac{\eta}{6} \leq \frac{\eta \|y_k^j - y_k^{j+1}\|}{6 \max\{\|b\| : b \in B \in \mathcal{B}\}}.$$

Setting $\sigma = \frac{|\eta|}{6 \max\{\|b\| : b \in B \in \mathcal{B}\}} > 0$ concludes the proof. $\square$

This second lemma relates $(b_k^j)^T \nabla^c f(y_k^j)$ to $(b_k^l)^T \nabla^c f(y_k^l)$ when Δ_k is small.

For the remainder of the section, we assume that the STRONG EXTENDED POLL step is used and that $\bar{y}$ is a limit point of the subsequence $\{y_k^{j(k)}\}_{k \in K}$ for which the continuous part $\bar{y}^c$ is in the strict interior of the feasible region $[\ell, u]$, where $j(k)$ is an index between 1 and J (where J depends on k). Moreover, we assume, without any loss in generality, that all iterates of this subsequence use the same positive basis that we denote by B. Note that the results of Proposition 4.3 concerning the other limit points still hold.

LEMMA 4.11. *For any $\eta < 0$ there exists an $\epsilon > 0$ and a $\delta' > 0$ such that if $\|y_k^j - \bar{y}\| \leq \epsilon$ and $\|y_k^l - y_k^j\| \leq \epsilon$, and $\Delta_k < \delta'$ for some k, j, and l, then $(b_k^j)^T \nabla^c f(y_k^j) - (b_k^l)^T \nabla^c f(y_k^l) \geq \frac{\eta}{3}$.*

Proof. Let $\eta < 0$ be given. Define $\epsilon > 0$ such that if $\|y_k^j - \bar{y}\| \leq \epsilon$ and $\|y_k^l - y_k^j\| \leq \epsilon$, then

$$(4.3) \qquad \|b^T (\nabla^c f(y_k^l) - \nabla^c f(y_k^j))\| \leq \frac{|\eta|}{12}$$

for all $b \in B$.

Using the mean value theorem, define $w_k^l = y_k^l + \alpha_k^l \Delta_k b_k^l$ and $w_k^j = y_k^j + \alpha_k^j \Delta_k b_k^j$ (where both α_k^l and α_k^j are in $[0, 1]$) such that $f(y_k^l + \Delta_k b_k^l) = f(y_k^l) + \Delta_k (b_k^l)^T \nabla^c f(w_k^l)$ and $f(y_k^l + \Delta_k b_k^j) = f(y_k^j) + \Delta_k (b_k^j)^T \nabla^c f(w_k^j)$.

Let $\delta' > 0$ be such that if $\Delta_k < \delta'$ for some $k \in K$, then $y_k^l + \Delta_k b \in \Omega$ for all positive bases directions $b \in B$. This is possible since $\bar{y}^c$ belongs to the strict interior of $[\ell, u]$.

$$(4.4) \quad \|(b_k^l)^T (\nabla^c f(w_k^l) - \nabla^c f(y_k^l))\| \leq \frac{|\eta|}{6} \text{ and } \|(b_k^j)^T (\nabla^c f(w_k^j) - \nabla^c f(y_k^l))\| \leq \frac{|\eta|}{12} .$$

Combining (4.3) (using $b = b_k^j$) with the second inequality of (4.4) yields

$$(4.5) \qquad \|(b_k^j)^T (\nabla^c f(w_k^j) - \nabla^c f(y_k^j))\| \leq \frac{|\eta|}{6}.$$

In summary, we have shown in the first inequality of (4.4) and in (4.5) that

$$(4.6) \quad (b_k^l)^T \nabla^c f(w_k^l) = (b_k^l)^T \nabla^c f(y_k^l) + \mu^l \text{ and } (b_k^j)^T \nabla^c f(w_k^j) = (b_k^j)^T \nabla^c f(y_k^j) + \mu^j,$$

where $|\mu^l| \leq \frac{|\eta|}{6}$ and $|\mu^j| \leq \frac{|\eta|}{6}$.

Moreover, since b_k^l is obtained through the STRONG EXTENDED POLL steps, and since $y_k^l + \Delta_k b_k^j$ is feasible, then it follows that $f(y_k^l + \Delta_k b_k^l) \leq f(y_k^l + \Delta_k b_k^j)$, and therefore $(b_k^l)^T \nabla^c f(w_k^l) \leq (b_k^j)^T \nabla^c f(w_k^j)$. Using the two equalities of (4.6) we get $(b_k^l)^T \nabla^c f(y_k^l) - (b_k^j)^T \nabla^c f(y_k^j) = u^j - u^l \leq \frac{|\eta|}{3}$. $\square$

The following result strengthens Theorem 4.8 by showing that there are no feasible descent directions at the limit points $\bar{y}$ of Proposition 4.9.

THEOREM 4.12. *If the limit points $\hat{x}$ and $\hat{y} \in \mathcal{N}(\hat{x})$ obtained under the STRONG EXTENDED POLL step satisfy $f(\hat{y}) = f(\hat{x})$, then $\nabla^c f(\bar{y}) = 0$.*

Proof. Suppose by way of contradiction that the limit point $\bar{y} \neq \hat{z}$ satisfies $\nabla^c f(\bar{y}) \neq 0$. Set $\eta = \min_{b \in B} b^T \nabla^c f(\bar{y}) < 0$ let δ and σ be from Lemma 4.10, and ϵ and δ' be from Lemma 4.11. Let k be an index in K such that the following six conditions hold:

(i) $\Delta_k \leq \delta$,   (iv) $b^T \nabla^c f(z_k) > \frac{\eta}{3}$ for all feasible directions $b \in B$,

(ii) $\Delta_k \leq \delta'$,   (v) $\|y_k^{j(k)} - \bar{y}\| < \epsilon$,

(iii) $(b_k^{j(k)})^T \nabla^c f(y_k^{j(k)}) < \frac{2\eta}{3}$,   (vi) $f(y_k^{j(k)}) - f(y_k^l) < \sigma\epsilon$ for any $l > j(k)$.

Conditions (i) and (ii) hold since $\Delta_k \to 0$. Condition (iii) holds since $b_k^{j(k)}$ is chosen with the STRONG EXTENDED POLL step. Theorem 4.8 implies condition (iv). Condition (v) follows since $y_k^{j(k)} \to \bar{y}$. Proposition 4.9 guarantees condition (vi) since $f(y_k^{j(k)}) \to f(\bar{\gamma}) = f(\hat{x})$.

Define the index $l(k) = \min \left\{ l \geq j(k) : (b_k^l)^T \nabla^c f(y_k^l) > \frac{\eta}{3} \right\}$ (condition (iv) guarantees that $l(k) \leq J$). Therefore, condition (i) and Lemma 4.10 ensure that $f(y_k^j) - f(y_k^{j+1}) > \sigma\|y_k^j - y_k^{j+1}\|$ when $j(k) \leq j < l(k)$. Writing out the telescopic sum leads to

$$f(y_k^{j(k)}) - f(y_k^{l(k)}) = \sum_{j=j(k)}^{l(k)-1} \left(f(y_k^j) - f(y_k^{j+1}) \right)$$

$$> \sigma \sum_{j=j(k)}^{l(k)-1} \|y_k^j - y_k^{j+1}\| \geq \sigma\|y_k^{j(k)} - y_k^{l(k)}\|.$$

Together with condition (vi) this gives $\|y_k^{j(k)} - y_k^{l(k)}\| \leq \epsilon$. Combining this with conditions (ii)–(v) and with Lemma 4.11 leads to

$$\frac{2\eta}{3} > (b_k^{j(k)})^T \nabla^c f(y_k^{j(k)})$$

$$= (b_k^{l(k)})^T \nabla^c f(y_k^{l(k)}) + \left((b_k^{j(k)})^T \nabla^c f(y_k^{j(k)}) - (b_k^{l(k)})^T \nabla^c f(y_k^{l(k)}) \right)$$

$$> \frac{\eta}{3} + \frac{\eta}{3} = \frac{2\eta}{3},$$

which is a contradiction.   ☐

In the next section, we illustrate the behavior of the algorithm through two examples.

5. Examples. The first example shows the value of the STRONG EXTENDED POLL versus the cheaper EXTENDED POLL step. This illustrates the difference between Proposition 4.9 and Theorem 4.12. The second example shows how the algorithm behaves on the larger problem presented in section 2.2.

5.1. Illustration of the stronger version of the algorithm. Consider the following example in which there are two continuous variables and a single binary one. In order to simplify notation, the continuous variables x^c are written $x^c = (a, b)$. The objective function is

$$f(x) = f(a, b, x^d) = g(a, b)(1 - x^d) + h(a, b)x^d,$$

where $g(a, b) = a^2 + b^2$ and $h(a, b) = a^2 v + a(1 - b)$. Both variables are constrained to be in the interval $[-2, 2]$, but these bounds are never approached by the trial points.

The pattern search algorithm we apply here does not have a SEARCH step; we use only a POLL and an EXTENDED POLL step triggered by $\xi = 1$. The current mesh neighborhood at iteration k is defined to be

$$\mathcal{N}^c(x) = \{x + \Delta_k(0, 1, 0),\ x + \Delta_k(0, -1, 0),\ x + \Delta_k(5, 0, 0),\ x + \Delta_k(-7, 0, 0)\}$$

for any $x = (a, b, x^d)$ except when the iterate may be written $x = (2\Delta_k, 1 - \Delta_k, 1)$, in which case it is

$$\mathcal{N}^c(x) = \{x + \Delta_k(0, -1, 0),\ x + \Delta_k(5, 1, 0),\ x + \Delta_k(-7, 1, 0)\}.$$

The set of neighbors of $x = (a, b, x^d)$ is $\mathcal{N}(x) = \{(a, b, 1 - x^d), (a, b, x^d)\}$. This definition ensures that the discrete variable always remain binary. Iteration k is declared successful and stops as soon as the incumbent is improved and $\Delta_{k+1} = \Delta_k$. Otherwise $\Delta_{k+1} = \frac{\Delta_k}{2}$.

The algorithm is initiated at $x_0 = (1, 0, 0)$ with $\Delta_0 = \frac{1}{4}$ and with incumbent value $f(x_0) = 1$. The POLL step evaluates the function at the points of $\mathcal{N}^c(x_0) : f(1, \frac{1}{4}, 0) = \frac{17}{16}, f(1, \frac{-1}{4}, 0) = \frac{17}{16}, f(\frac{9}{4}, 0, 0) = \frac{81}{16}$, and $f(\frac{-3}{4}, 0, 0) = \frac{9}{16}$. This first iteration is successful.

Iteration 1 starts at $x_1 = (\frac{-3}{4}, 0, 0)$ with $\Delta_0 = \frac{1}{4}$ and $f(x_1) = \frac{9}{16}$. The POLL step computes f in $\mathcal{N}^c(x_1) : f(\frac{-3}{4}, \frac{1}{4}, 0) = \frac{10}{16}, f(\frac{-3}{4}, \frac{-1}{4}, 0) = \frac{10}{16}, f(\frac{1}{2}, 0, 0) = \frac{1}{4}$. This iteration is also successful.

Iteration 2 starts at $x_2 = (\frac{1}{2}, 0, 0)$ with $\Delta_0 = \frac{1}{4}$ and $f(x_2) = \frac{1}{4}$. The POLL step computes f in $\mathcal{N}^c(x_2) : f(\frac{1}{2}, \frac{1}{4}, 0) = \frac{5}{16}, f(\frac{1}{2}, \frac{-1}{4}, 0) = \frac{5}{16}, f(\frac{7}{4}, 0, 0) = \frac{49}{16}$, and $f(\frac{-5}{4}, 0, 0) = \frac{25}{16}$. Before declaring this iteration unsuccessful, polling must be conducted on the set of neighbors $\mathcal{N}(x_2) : f(\frac{1}{2}, 0, 1) = \frac{1}{2}$. This value is within ξ of $f(x_2)$ and so extended polling must be conducted around this last point y_2^0. The EXTENDED POLL step finds $y_2^1 = (\frac{1}{2}, \frac{1}{4}, 1)$ in $\mathcal{N}^c(y_2^0)$ with $f(y_2^1) = \frac{7}{16}$, then $y_2^2 = (\frac{1}{2}, \frac{1}{2}, 1)$ in $\mathcal{N}^c(y_2^1)$ with $f(y_2^1) = \frac{3}{8}$, and $y_2^3 = (\frac{1}{2}, \frac{3}{4}, 1)$ in $\mathcal{N}^c(y_2^2)$ with $f(y_2^1) = \frac{5}{16}$. It does not succeed in improving this last value in $\mathcal{N}^c(y_2^3) : f(\frac{1}{2}, \frac{1}{2}, 1) = \frac{15}{32}, f(\frac{7}{4}, 1, 1) = \frac{49}{16}$, and $f(\frac{-5}{4}, 1, 1) = \frac{25}{16}$. Thus, iteration 2 is unsuccessful and iteration 3 starts at the same point $x_2 = (\frac{1}{2}, 0, 0)$ with $\Delta_0 = \frac{1}{8}$ and $f(x_3) = \frac{1}{4}$.

Table 5.1 shows that the algorithm generates cycles composed of two successful iterations followed by an unsuccessful one. The three iterations detailed above, i.e., the first cycle, appear in the table by letting $\alpha = \frac{1}{4}$. Iteration 3 initiates a new cycle with $\alpha = \frac{1}{8}$.

Figure 5.1 displays the iterates of the EXTENDED POLL step from y_k to z_k. The circles represent the points y_k^j for $j = 0, 1, \ldots, J$. All these points are on the same line as the function decreases linearly when the variable a is fixed to 2α. At the last point z_k, the current mesh neighborhood is evaluated using a different positive basis. The set $\mathcal{N}^c(z_k)$ is represented by the three circled crosses. As k goes to infinity, the points $\{y_k^j : j = 0, 1, \ldots, J\}$ converge to the line segment from $\hat{y} = (0, 0, 1)$ to $\hat{z} = (0, 1, 1)$, which is the thick line in Figure 5.1. The objective function value is equal to zero there. The gradient norm is nonzero at $\hat{y}$ but decreases to zero at $\hat{z}$.

In order to ensure that the gradient norm is zero at all points of $\mathcal{N}(\hat{x})$, the stronger version of the algorithm must be used. By doing this, the EXTENDED POLL step at iteration 2 discovers the point $y^1 = (\frac{-5}{4}, 0, 1)$ of $\mathcal{N}^c(y^0)$, whose function value is $\frac{-5}{4}$. This iteration is successful, and the iterates eventually converge to the global minimizer of f over Ω.

TABLE 5.1
In three consecutive iterations, the iterates go from $x_k = 4\alpha, \Delta_k = \alpha$ to $x_{k+3} = 2\alpha, \Delta_{k+3} = \frac{\alpha}{2}$.

x_k	$x_k + \Delta_k(0,1,0)$	$x_k + \Delta_k(0,-1,0)$	$x_k + \Delta_k(5,0,0)$	$x_k + \Delta_k(-7,0,0)$
$(4\alpha,0,0)$	$(4\alpha,\alpha,0)$	$(4\alpha,-\alpha,0)$	$(9\alpha,0,0)$	$(-3\alpha,0,0)$
$16\alpha^2$	$17\alpha^2$	$17\alpha^2$	$81\alpha^2$	$9\alpha^2$
$(-3\alpha,0,0)$	$(-3\alpha,\alpha,0)$	$(-3\alpha,-\alpha,0)$	$(-2\alpha,0,0)$	$(-10\alpha,0,0)$
$9\alpha^2$	$10\alpha^2$	$10\alpha^2$	$4\alpha^2$	$100\alpha^2$
$(2\alpha,0,0)$	$(2\alpha,\alpha,0)$	$(2\alpha,-\alpha,0)$	$(7\alpha,0,0)$	$(-5\alpha,0,0)$
$4\alpha^2$	$5\alpha^2$	$5\alpha^2$	$49\alpha^2$	$25\alpha^2$
EXTENDED	$y_k = y_k^0$	y_k^1	$y_k^2 \quad \ldots$	$z_k = y_k^J$
POLL:	$(2\alpha,0,1)$	$(2\alpha,\alpha,1)$	$(2\alpha,2\alpha,1)$	$(2\alpha,1-\alpha,1)$
	2α	$2\alpha(1-\alpha(1-2\alpha))$	$2\alpha(1-2\alpha(1-2\alpha))$	$2\alpha^2(3-2\alpha))$
$\mathcal{N}^c(z_k):$	$z_k + \Delta_k(0,-1,0)$	$z_k + \Delta_k(-7,1,0)$	$z_k + \Delta_k(5,1,0)$	
	$(2\alpha,1-2\alpha,1)$	$(7\alpha,1,1)$	$(-5\alpha,1,1)$	
	$2\alpha^2(4-4\alpha^2)$	$49\alpha^2$	$25\alpha^2$	

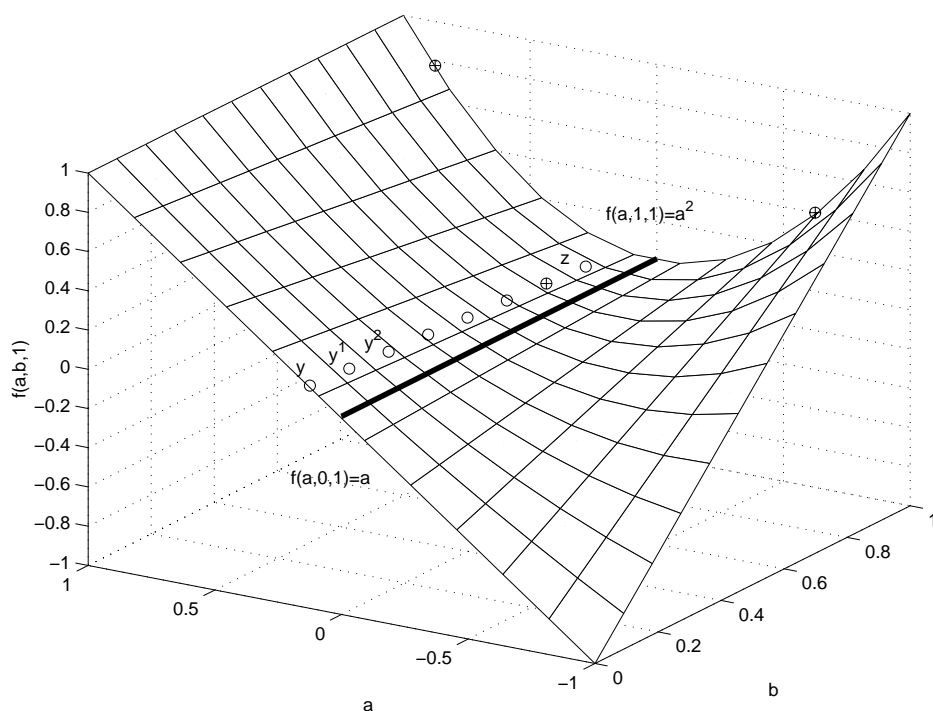FIG. 5.1. *Extended polling from $y = (2\alpha,0,1)$ to $z = (2\alpha,1-\alpha,1)$.*

5.2. A thermal insulation system. We ran the GMVP algorithm on the example of section 2.2. The behavior was typical of derivative-free algorithms in making rapid improvement of the objective function value and then reaching a plateau, where the objective function value does not decrease significantly.

The initial point was a single shield at temperature $T_1 = 150$ surrounded by $M_1 =$ Teflon and $M_2 =$ nylon of thickness $\Delta x_1 = \Delta x_2 = 50$. An upper bound of 100 was imposed on the number of shields. (This bound was large enough so that it was never reached.) The GMVP algorithm later used the third insulator (epoxy-fiberglass). We
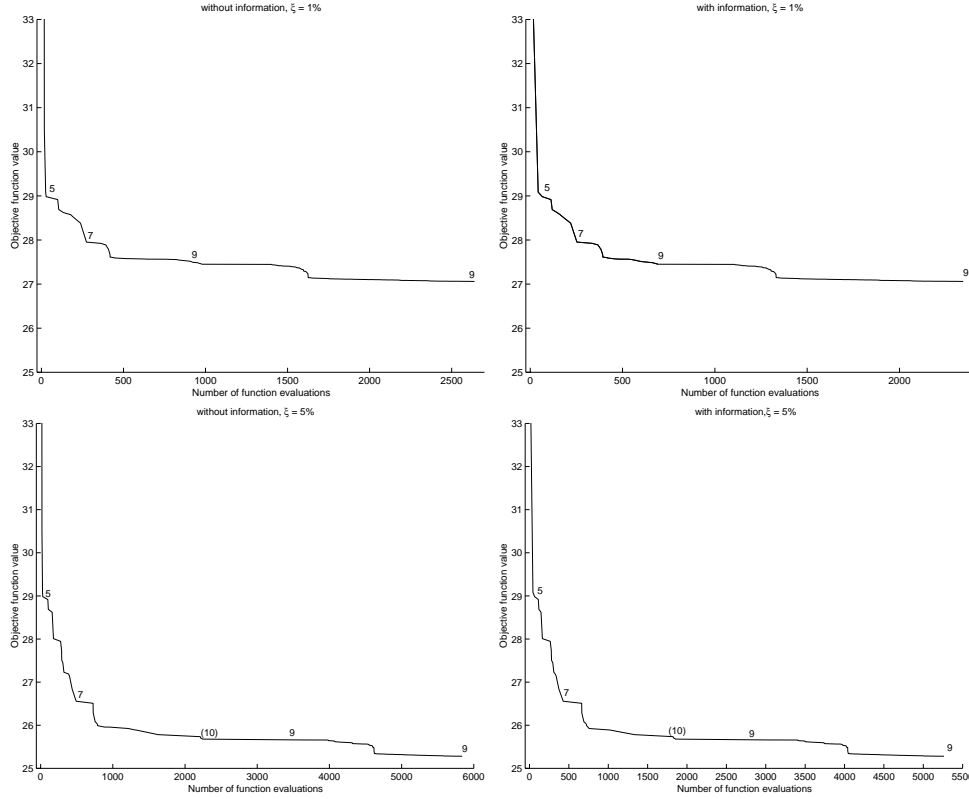
FIG. 5.2. *Progress of the objective function value, without or with extra information, and with* $\xi = 1\%$ *or* 5%.

used the set of discrete neighbors suggested in section 2.2. The algorithm consisted mainly of POLL steps. A simple SEARCH step was invoked if the previous iteration was successful and the incumbent solution (x_k^c, x_k^d) differed from the previous one only in its continuous components: $x_k^c \neq x_{k-1}^c$. This SEARCH step consisted of a point further in the same successful direction: $(x_k^c + 2(x_k^c - x_{k-1}^c), x_k^d)$.

The mesh size parameter was initially set at 10 and not increased. It decreased at local mesh optimizers. Figure 5.2 shows the improvement of the objective function value with the number of function evaluations for four runs. They illustrate the user's control in defining local optimality and in triggering extended polling with respect to the categorical variables. The y-axis is truncated for readability since $f(x_0) \approx 262$. The line on the graph starts at the 18th iteration.

The right-hand graphs use properties of the insulators (such as the fact that Teflon is a much better insulator than nylon at high temperatures) to further restrict the discrete POLL set. The top graphs use $\xi_i = 1\% f(x_i)$ to trigger the EXTENDED POLL step. The bottom graphs use the larger value $\xi_i = 5\% f(x_i)$. As expected, a larger value triggers more EXTENDED POLL steps and uses more objective function evaluations, but finds a better solution. Also, the runs that use the extra information converge using fewer function evaluations.

The numbers on the curves indicate the number of shields at local mesh optimizers, i.e., at unsuccessful iterations. For the top left graph and for $\Delta_k = 10, 5, \frac{5}{4}$, and

$\frac{5}{32}$, there were $5, 7, 9$ and 9 shields, respectively. The number of shields did not increase monotonically. Before reaching five shields, the GMVP algorithm added a third one then immediately removed another. On the bottom graphs, it added a tenth shield and later removed it. (This is indicated by the number in parentheses on the graphs.)

On the top left graph, the algorithm converged to a solution containing 9 shields and a combination of all three types of insulators. The best solution was found after 2639 function evaluations, and an additional 2083 showed it to be a local mesh optimizer—a total of 4722 function evaluations. The top right graph gives the progress using the more restrictive definition of the set of neighbors discussed above. It found the same local mesh optimizer but with 10% fewer function evaluations. It took 2345 evaluations to find the solution and 1861 more to show it to be a local mesh optimizer—a total of 4206 function evaluations.

On the bottom left graph, the algorithm converged after a total of 13,329 function evaluations to a solution that uses the three insulators and whose objective function value is reduced by more than 6%. The same solution was found on the bottom right, but using 10,053 function evaluations (approximately 25% fewer). All four runs produced a solution having 9 shields. The difference in the objective function values suggests the presence of local optimum solutions.

Further computational results on this and related problems can be found with a more engineering slant in [8].

Acknowledgments. The authors would like to thank the referees, David Applegate, and Yin Zhang for discussions which helped improve the quality of this paper. Special thanks are due to our collaborator Michael Kokkolaras who suggested the thermal insulation problem and provided Figure 2.1.

REFERENCES

[1] C. AUDET, *Convergence Results for Pattern Search Algorithms are Tight,* Tech. report TR98-24, Department of Computational and Applied Mathematics, Rice University, Houston TX, 1998.

[2] A.J. BOOKER, J.E. DENNIS, JR., P.D. FRANK, D.B. SERAFINI, V. TORCZON, AND M.W. TROSSET, *A rigorous framework for optimization of expensive functions by surrogates,* Structural Optim. 17 (1999), pp. 1–13.

[3] G.E.P. BOX, *Evolutionary operation: A method for increasing industrial productivity,* Appl. Statist., 6 (1957) pp. 81–101.

[4] C. DAVIS, *Theory of positive linear dependence,* Amer. J. Math., (1954), pp. 448–474.

[5] J.E. DENNIS, JR. AND V. TORCZON, *Direct search methods on parallel machines,* SIAM J. Optim., 1 (1991), pp. 448–474.

[6] M.A. HILAL AND R.W. BOOM, *Optimization of mechanical supports for large superconductive magnets*, Adv. Cryogenic Engrg., 22 (1977), pp. 224–232.

[7] R. HOOKE AND T.A. JEEVES, *Direct search solution of numerical and statistical problems,* J. ACM, 8 (1961), pp. 212–229.

[8] M. KOKKOLARAS, C. AUDET, AND J.E. DENNIS, JR., *Mixed Variable Optimization of the Number and Composition of Heat Intercepts in a Thermal Insulation System,* Tech. report TR00-21, Department of Computational and Applied Mathematics, Rice University, Houston TX, 2000.

[9] R.M. LEWIS AND V. TORCZON, *Pattern search algorithms for bound constrained minimization,* SIAM J. Optim., 9 (1999), pp. 1082–1099.

[10] R.M. LEWIS AND V. TORCZON, *Rank Ordering and Positive Basis in Pattern Search Algorithms,* Tech. report TR 96-71, ICASE NASA Research Center, Hampton, VA, 1996.

[11] R.M. LEWIS AND V. TORCZON, *Pattern search methods for linearly constrained minimization,* SIAM J. Optim., 10 (2000), pp. 917–941.

[12] V. TORCZON, *On the convergence of pattern search algorithms,* SIAM J. Optim., 7 (1997), pp. 1–25.