GENERALIZED PATTERN SEARCH ALGORITHMS WITH ADAPTIVE PRECISION FUNCTION EVALUATIONS

by

Elijah Polak and Michael Wetter

GENERALIZED PATTERN SEARCH
ALGORITHMS WITH ADAPTIVE
PRECISION FUNCTION EVALUATIONS

by

Elijah Polak and Michael Wetter

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations [1]

Elijah Polak[2] and Michael Wetter[3]

September 7, 2001

Abstract

In the literature on generalized pattern search algorithms, it is usually assumed that the cost function can be evaluated exactly. However, there is a large class of engineering problems where the numerical evaluation of the cost function involves the solution of systems of ordinary and partial differential equations as well as algebraic equations. Computer code for solving these systems can be millions of lines long, and often defines a discontinuous numerical approximation to the cost function. In such cases, standard generalized pattern search algorithms are not applicable.

In this paper we extend a class of generalized pattern search algorithms to a form that uses adaptive precision approximations to the cost function. These numerical approximations need not define a continuous function. Our algorithms can be used for solving linearly constrained problems with cost functions that are at least locally Lipschitz continuous.

Assuming that the cost function is smooth, we prove that our algorithms converge to a stationary point. Under the weaker assumption that the cost function is only locally Lipschitz continuous, we show that our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions.

An important feature of our adaptive precision scheme is the use of coarse approximations in the early iterations, with the approximation precision controlled by a test. Such an approach leads to substantial time savings in minimizing computationally expensive functions.

Key words: Algorithm implementation, approximations, generalized pattern search, Hooke-Jeeves, Clarke's generalized directional derivative, nonsmooth optimization.

[2]Department of Electrical Engineering, University of California at Berkeley, Berkeley, CA 94720, USA (polak@eecs.berkeley.edu).

[3]Simulation Research Group, Building Technologies Department, Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA (MWetter@lbl.gov).

1

1 Introduction

Generalized pattern search (GPS) algorithms are derivative free methods for the minimization of smooth functions, possibly with linear inequality constraints. Examples of pattern search algorithms are the coordinate search [9], the pattern search algorithm of Hooke and Jeeves [5], and the multidirectional search algorithm of Dennis and Torczon [6]. What they all have in common is that they construct a mesh, or pattern, which is then explored according to some rule, and if a decrease in cost is not obtained on the current mesh, then the mesh is refined and the process is repeated. In 1997, Torczon [11] was the first to show that all the existing pattern search algorithms are specific implementations of an abstract pattern search scheme and to establish that for unconstrained problems with smooth cost functions, the gradient of the cost function vanishes at accumulation points of sequences constructed by this scheme. Lewis and Torczon extended her theory to address bound constrained problems [7] and problems with linear inequality constraints [8]. In both cases, convergence to a feasible point x^* satisfying $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all feasible x is proven under the condition that $f(\cdot)$ is once continuously differentiable. Audet and Dennis [1] present a simpler abstraction of GPS algorithms, and, in addition to reestablishing the Torczon and the Lewis and Torczon results, they relax the assumption that the cost function is smooth to that it is locally Lipschitz continuous. However, their characterization of accumulation points of sequences constructed by a GPS algorithm, on a locally Lipschitz continuous cost function, while not without merit, falls short of showing that the accumulation points are stationary in the Clarke sense [3] (i.e., $0 \in \partial^0 f(x^*)$). It does not seem possible to improve their result.

In principle, a natural area for the application of GPS algorithms is engineering optimization, where the cost functions are defined on the solution of complex systems of simultaneous equations including algebraic equations, ordinary differential equations, and partial differential equations. In such cases, obtaining an accurate approximation to the cost function often takes many hours, if not days, and there is no straightforward way of approximating gradients. Furthermore, it is common that the termination criteria of the numerical solvers introduce discontinuities in the resulting approximations to the cost functions. Hence, standard GPS algorithms are not applicable because the cost function is discontinuous and too expensive to evaluate with fixed precision throughout the optimization.

In this paper we present a modified class of GPS algorithms which adjust the precision of the function evaluations adaptively: low precision in the early iterations, with precision progressively increasing as a solution is approached. The fact, that the approximations to the cost function may be discontinuous, does not affect the convergence of these methods to stationary points.

The GPS algorithms that we present are somewhat simpler in structure than those presented in [11, 7, 8] and [1]. We assume that the cost function $f(\cdot)$ is at least locally Lipschitz continuous and that it can be approximated by a family of functions, say $\{f_{\epsilon_k}(\cdot)\}_{k=0}^{\infty}$, where each $f_{\epsilon_k}(\cdot)$ may be discontinuous but converges to $f(\cdot)$ uniformly on bounded sets. A test in the algorithm determines when precision must be increased. This

test makes use only of the current mesh size and includes parameters that can be used to control the speed with which precision is increased. This flexibility can be exploited to obtain an order of magnitude reduction in computing times, as compared to using high precision throughout the computation. Since our GPS algorithms include global search and local search stages, as is typical in GPS algorithms, our GPS algorithms can also be used with surrogate cost functions for the the global search, as in [10, 2].

Under the assumption that the cost function is continuously differentiable, all the accumulation points constructed by our GPS are stationary, while under the assumption that $f(\cdot)$ is only locally Lipschitz continuous, we regain the results of [1].

2 Notation

1. We denote by $\mathbb{Z}$ the set of integers, by $\mathbb{Q}$ the set of rational numbers, and by $\mathbb{N} \triangleq \{0, 1, \ldots\}$ the set of natural numbers. The set $\mathbb{N}_+$ is defined as $\mathbb{N}_+ \triangleq \{1, 2, \ldots\}$. Similarly, vectors in $\mathbb{R}^n$ with strictly positive elements are denoted by $\mathbb{R}^n_+ \triangleq \{x \in \mathbb{R}^n \mid x^i > 0, \forall i = 1, \ldots, n\}$ and the set $\mathbb{Q}_+$ is defined as $\mathbb{Q}_+ \triangleq \{q \in \mathbb{Q} \mid q > 0\}$.

2. The inner product in $\mathbb{R}^n$ is denoted by $\langle \cdot, \cdot \rangle$ and for $x, y \in \mathbb{R}^n$ defined by $\langle x, y \rangle \triangleq \sum_{i=1}^n x^i y^i$.

3. If a subsequence $\{x_i\}_{i \in K} \subset \{x_i\}_{i=0}^\infty$ converges to some point x, we write $x_i \to^K x$.

4. Let $\mathbb{W}$ be a set containing a sequence $\{w_i\}_{i=0}^k$. Then, we denote by $\underline{w}_k$ the sequence $\{w_i\}_{i=0}^k$ and by $\underline{\mathbb{W}}_k$ the set of all k element sequences in $\mathbb{W}$.

5. We denote by $\{e_i\}_{i=1}^n$ the unit vectors in $\mathbb{R}^n$.

6. If $\mathbb{X}$ is a set, we denote by $\partial \mathbb{X}$ its boundary and by $\overline{\mathbb{X}}$ its closure.

7. If $\mathbb{S}$ is a set, we denote by $2^{\mathbb{S}}$ the set of all nonempty subsets of $\mathbb{S}$.

8. If $\widehat{D} \in \mathbb{Q}^{n \times q}$ is a matrix, we will use the notation $\widehat{d} \in \widehat{D}$ to denote the fact that $\widehat{d} \in \mathbb{Q}^n$ is a column vector of the matrix $\widehat{D}$. Similarly, by $D \subset \widehat{D}$ we mean that $D \in \mathbb{Q}^{n \times p}$ $(1 \le p \le q)$ is a matrix containing only columns of $\widehat{D}$. Further, card(D) denotes the number of columns of D.

9. The least common multiple of a set of natural numbers is the smallest nonzero natural number that is a multiple of all the elements in the set.

3 Minimization Problem

We want to solve the linearly constrained problem

$$\min_{x \in \mathbb{X}} f(x) \tag{1a}$$

$$\mathbb{X} \triangleq \left\{ x \in \mathbb{R}^n \mid l \le Qx \le u; \ l, u \in \mathbb{R}^{n_c} \cup \{\pm\infty\}; \ l < u; \ Q \in \mathbb{Q}^{n_c \times n} \right\} \tag{1b}$$

where the cost function $f \colon \mathbb{R}^n \to \mathbb{R}$ is (at least) Lipschitz continuous and the number of constraints n_c is finite.

We assume that the function $f(\cdot)$ cannot be evaluated exactly, but that it can be approximated by functions $f_{\epsilon_k} \colon \mathbb{R}^n \to \mathbb{R}$, $k \in \mathbb{N}$, where $\epsilon_k \in \mathbb{R}_+^q$ is a vector containing precision parameters. We will assume that $f(\cdot)$ and its approximating functions $\{f_{\epsilon_k}(\cdot)\}_{k=0}^{\infty}$ have the following properties.

Assumption 3.1

1. There exists an error bound function $\varphi \colon \mathbb{R}_+^q \to \mathbb{R}_+$ such that for any bounded set $S \subset X$, there exists an $\epsilon_S \in \mathbb{R}_+^q$ and a scalar $K_S \in (0, \infty)$ such that for all $x \in S$ and for all $\epsilon_k \in \mathbb{R}_+^q$, with $\epsilon_k \leq \epsilon_S$ [4],

$$| f_{\epsilon_k}(x) - f(x)| \leq K_S \, \varphi(\epsilon_k). \tag{2}$$

Furthermore,

$$\lim_{\epsilon_k \to 0} \varphi(\epsilon_k) = 0. \tag{3}$$

2. The function $f \colon \mathbb{R}^n \to \mathbb{R}$ is at least locally Lipschitz continuous. □

Remark 3.2 The functions $f_{\epsilon_k} \colon \mathbb{R}^n \to \mathbb{R}$ may be discontinuous. □

Next, we state an assumption on the level sets of the family of approximate functions. To do so, we first define the notion of a level set.

Definition 3.3 (Level Set) *Given a function $f \colon \mathbb{R}^n \to \mathbb{R}$ and an $\alpha \in \mathbb{R}$, such that $\alpha \geq \inf_{x \in \mathbb{R}^n} f(x)$, we will say that the set $L_\alpha(f) \subset \mathbb{R}^n$, defined as*

$$L_\alpha(f) \triangleq \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}, \tag{4}$$

is a level set *of $f(\cdot)$, parametrized by α.* □

Assumption 3.4 (Compactness of Level Sets) *Let $\{f_{\epsilon_k}(\cdot)\}_{k=0}^{\infty}$ be as in Assumption 3.1 and let $X \subset \mathbb{R}^n$ be the constraint set. Let $x_0 \in X$ be the initial iterate and $\epsilon_0 \in \mathbb{R}_+^q$ be the initial precision parameter. Then, we assume that there exists a compact set $C \subset \mathbb{R}^n$ such that*

$$L_{f_{\epsilon_0}(x_0)}(f_{\epsilon_k}) \cap X \subset C \qquad \forall k \in \mathbb{N}. \tag{5}$$

□

---

[4] For $\epsilon \in \mathbb{R}_+^q$, by $\epsilon \leq \epsilon_S$, we mean that $\epsilon^i \leq \epsilon_S^i$, for all $i = 1, \ldots, q$.

4 Generalized Pattern Search Algorithms

4.1 Geometric Aspects of the Algorithms

A major aspect of any GPS algorithm is the rule for constructing the meshes on which the searches are conducted. The main difference between our rule for mesh construction and those of others, such as the one of Audet's and Dennis [1], is that we use a different rule for mesh refinement, which results in our meshes being nested, and hence simplifies the explanation of the geometry of mesh construction. As far as we can tell, our simplification has no impact on computational efficiency.

The k-th iteration of our GPS algorithms has the following structure. We begin with the current iterate x_k, precision parameter ϵ_k, and mesh $\mathbb{M}_k$. A set-valued map is used to select a finite subset of mesh points in $\mathbb{M}_k$, for the so-called "global search". If this set contains a point x' such that $f_{\epsilon_k}(x') < f_{\epsilon_k}(x_k)$, then we set $x_{k+1} = x'$, $\epsilon_{k+1} = \epsilon_k$, $\mathbb{M}_{k+1} = \mathbb{M}_k$, and update the index k to $k + 1$. If the global search set does not yield a point of lower cost, we proceed to a "local search", which consists of evaluating $f_{\epsilon_k}(\cdot)$ on a set of neighbors of x_k in the mesh $\mathbb{M}_k$. If a point x' of lower cost is found, then we set $x_{k+1} = x'$, $\epsilon_{k+1} = \epsilon_k$, $\mathbb{M}_{k+1} = \mathbb{M}_k$, and update the index k to $k + 1$. If the local search also fails to produce an improvement, then the mesh $\mathbb{M}_k$ is subdivided to yield a finer mesh $\mathbb{M}_{k+1}$, and the precision parameter is reduced according to a prescribed rule. After updating k to $k + 1$, the entire process is repeated.

We will now flesh out the geometric details of our GPS algorithms. We begin with the construction of the meshes.

4.1.1 Construction of the Meshes

Before we can explain how the mesh is to be constructed, we must introduce the notions of a positive combination and of a positive span, as defined by Davis [4], and that of a base direction matrix.

Definition 4.1 (Positive Combination, Positive Span)

  1. A positive combination *of vectors* $\{v_i\}_{i=1}^{p}$ *is a linear combination* $\sum_{i=1}^{p} \lambda_i v_i$ *with* $\lambda_i \geq 0$ *for all* $i \in \{1, \ldots, p\}$.

  2. Let V be a $n \times p$ matrix and let $\mathbb{C} \subset \mathbb{R}^n$ be a convex cone. If every $c \in \mathbb{C}$ can be written as a positive combination of columns of V, then the columns of V are said to positively span $\mathbb{C}$, and the matrix V is said to be a positive spanning set.

  3. If $p = n + 1$, then V is said to be a minimal positive spanning set. $\square$

We will denote by $\mathbb{S}$ the set of all matrices whose columns positively span $\mathbb{R}^n$.

Next, we define a *base direction matrix*. We will use the columns of the base direction matrix to specify the mesh and hence the search directions. The base direction matrix will be fixed for all iterations.

Definition 4.2 (Base Direction Matrix) *Let* $\mathbb{S}$ *be the set of all matrices whose columns positively span* $\mathbb{R}^n$. *Then, the* base direction matrix $\widehat{D}$ *is any matrix satisfying*

$$\widehat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S} \tag{6}$$

where $p > n$ *is any arbitrary but finite natural number.* □

Remark 4.3 The fact that the matrix $\widehat{D}$ has only rational elements makes it very easy to establish the minimal distance between distinct mesh points (Lemma 5.1). At the same time, from a computational point of view, requiring $\widehat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ rather than $\widehat{D} \in \mathbb{R}^{n \times p} \cap \mathbb{S}$ does not result in any practical inconvenience. □

Note that the base direction matrix $\widehat{D}$ may not be a minimal positive spanning set, e.g., for the one-dimensional case, $\widehat{D} = [-1, \ 1, \ 1.1]$ would not be minimal. Hence, $\widehat{D}$ can be used to generate a set $\mathbb{D}_{\widehat{D}}$, which we define as the set of all submatrices of $\widehat{D}$ (constructed by deleting columns of $\widehat{D}$) whose column vectors positively span $\mathbb{R}^n$.

The meshes, over which our algorithms search, are constructed iteratively, as follows.

Definition 4.4 (*k*-th Mesh) *Let* $x_0 \in \mathbb{X}$, $r, s_0, k \in \mathbb{N}$, *with* $r > 1$, $\{t_i\}_{i=0}^{k-1} \subset \mathbb{N}$, *and the base direction matrix* $\widehat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ *be given, and let*

$$\Delta_k \triangleq \frac{1}{r^{s_k}}, \tag{7}$$

where for $k > 0$

$$s_k = s_0 + \sum_{i=0}^{k-1} t_i. \tag{8}$$

Then we define the mesh $\mathbb{M}_k$ *by*

$$\mathbb{M}_k \triangleq \{x_0 + \Delta_k \widehat{D} m \mid m \in \mathbb{N}^p\}. \tag{9}$$

□

It should be clear from the construction of the meshes that whenever $t_k > 0$, the mesh $\mathbb{M}_{k+1}$ is obtained from the mesh $\mathbb{M}_k$ by dividing the intervals between neighboring points of the mesh $\mathbb{M}_k$ into r^{t_k} subintervals by adding additional mesh points. Therefore, it is clear that the meshes are nested, i.e., $\mathbb{M}_k \subset \mathbb{M}_{k+1}$ with equality if $\Delta_{k+1} = \Delta_k$.

We now present two examples: first a simple example of a mesh that is generated by a minimal positive spanning set, and then an example of a mesh generation using a more complicated base direction matrix $\widehat{D}$.
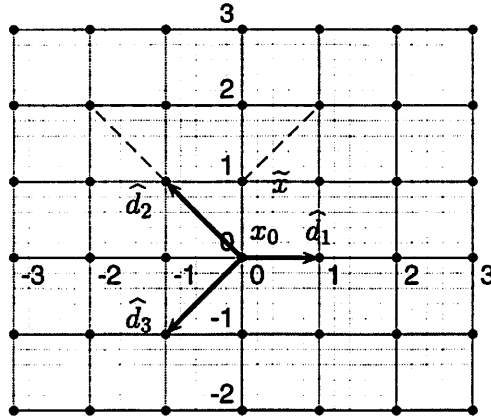
Figure 1: Minimal positive spanning set $\widehat{D} = [\widehat{d_1}, \widehat{d_2}, \widehat{d_3}]$ and generated mesh in $\mathbb{R}^2$

Example 4.5 The vectors $\{\widehat{d_i}\}_{i=1}^3$ in Fig. 1 form a minimal positive spanning set. The base direction matrix $\widehat{D}$ is defined by

$$\widehat{D} = \begin{pmatrix} \widehat{d_1} & \widehat{d_2} & \widehat{d_3} \end{pmatrix} \triangleq \begin{pmatrix} 1 & -1 & -1 \\ 0 & 1 & -1 \end{pmatrix}. \tag{10}$$

In Fig. 1, the bullets (•) are the mesh points of the mesh $\mathbb{M}_k = \{0 + 1\widehat{D}m \mid m \in \mathbb{N}^3\}$. For example, in Fig. 1, $\widetilde{x} = \widehat{D}m$, with $m = (3, 2, 1)^T$. □

Next we present a mesh constructed using a more complicated base direction matrix.

Example 4.6 Fig. 2 shows a mesh generated using $x_0 = 0$, $\Delta_k = 1$ and the base direction matrix

$$\widehat{D} = \begin{pmatrix} \widehat{d_1} & \widehat{d_2} & \widehat{d_3} \end{pmatrix} = \begin{pmatrix} 1 & -0.5 & -0.75 \\ 0 & 1 & -0.75 \end{pmatrix}. \tag{11}$$

Fig. 2(a) shows the vectors $\{\widehat{d_i}\}_{i=1}^3$ (bold arrows) and all possible mesh points of the form $\widehat{D}v$ with $v = (n, 0, 0)^T$, $v = (0, n, 0)^T$, and $v = (0, 0, n)^T$ where $n \in \mathbb{N}$. Each arrow points to a mesh point and indicates how the base vectors $\{\widehat{d_i}\}_{i=1}^3$ are added to obtain the mesh points. Fig. 2(b) shows the set of all mesh points of the form $\widehat{D}v$ with $v = (n, m, 0)^T$ and $v = (n, 0, m)^T$ where $n, m \in \mathbb{N}$. For example, the point labeled with $\widetilde{x}$ is given by $\widetilde{x} = \widehat{D}v$ where $v = (2, 1, 0)^T$. In Fig. 2(c), more mesh points are drawn by adding some positive multiple of $\widehat{d_2}$ to some mesh points that have been generated in Fig. 2(b). For clarity, not all possible mesh points are drawn. In Fig. 2(d), additional mesh points are generated by adding some positive multiple of $\widehat{d_3}$ to some mesh points of Fig. 2(c). Fig. 2(e) finally contains all possible mesh points, now indicated by bullets (•). For clarity, only the vectors $\{\widehat{d_i}\}_{i=1}^3$ are drawn in Fig. 2(e). □
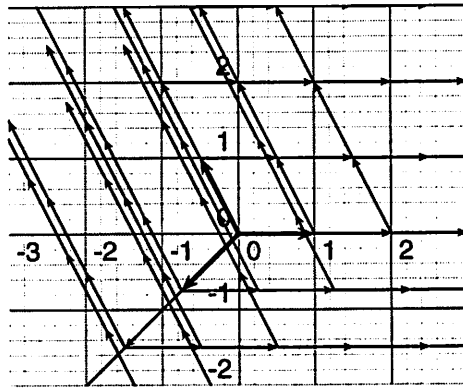
4.1.2 Global and Local Search Set

We will now characterize the set-valued maps that determine the mesh points for the "global" and "local" searches. Note that the images of these maps may depend on the entire history of the computation.
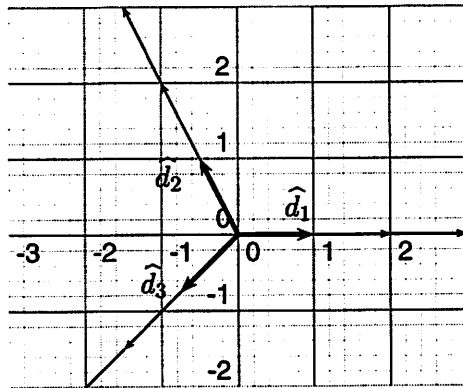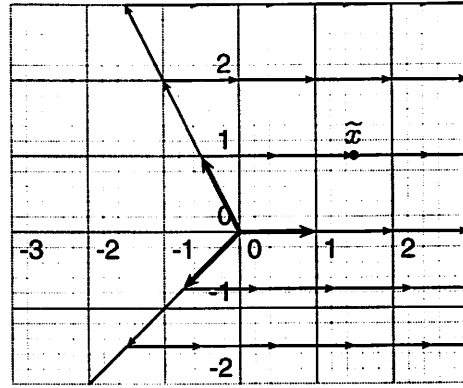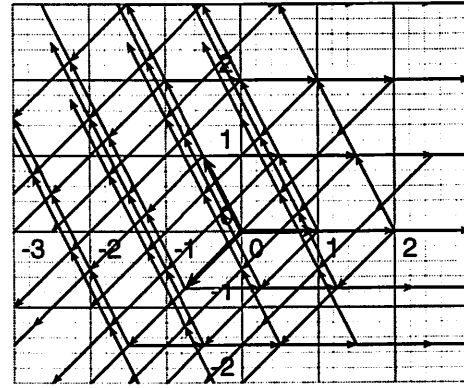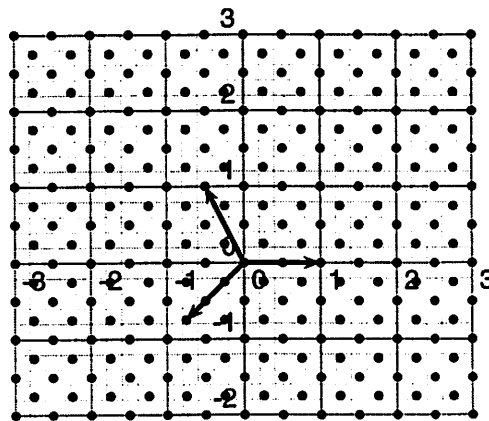
Figure 2: Generation of a mesh in $\mathbb{R}^2$

Definition 4.7 (Candidates for Direction Matrix, Direction Matrix)

1. *Let $\mathbb{S}$ be the set of all matrices whose column vectors positively span $\mathbb{R}^n$. Given a base direction matrix $\widehat{D}$, we define the* candidates for direction matrices *to be the set*

$$\mathbb{D}_{\widehat{D}} \triangleq \{D \mid D \subset \widehat{D} \cap \mathbb{S}\} \tag{12}$$

where the matrix D is constructed by deleting columns of $\widehat{D}$.

2. *We will call the elements of $\mathbb{D}_{\widehat{D}}$ to be the* direction matrices. □

Definition 4.8 *Let $\underline{\mathbf{X}}_k \subset \mathbb{R}^n$ and $\underline{\Delta}_k \subset \mathbb{Q}_+$ be the sets of all sequences containing k elements, let $\mathbb{M}_k$ be the current mesh, and let $\mathbb{D}_{\widehat{D}}$ be the set of all candidates for the direction matrix.*

1. *We define the* global search set map *to be any set-valued map*

$$\gamma_k : \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{R}^q_+ \to \left(2^{\mathbb{M}_k} \cap \mathbf{X}\right) \cup \emptyset \tag{13a}$$

whose image $\gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$ contains only a finite number of mesh points.

2. *We define the* local search direction map *to be any map*

$$\delta_{\widehat{D},k} : \underline{\mathbf{X}}_k \times \underline{\Delta}_k \to \mathbb{D}_{\widehat{D}}. \tag{13b}$$

3. *We will call $\mathcal{G}_k \triangleq \gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$ the* global search set.

4. *With $D_k = \delta_{\widehat{D},k}(\underline{x}_k, \underline{\Delta}_k)$, we will call*

$$\mathcal{L}_k \triangleq \{x_k + \Delta_k D_k e_j \mid j = 1, \ldots, \mathrm{card}(D_k)\} \cap \mathbf{X} \tag{13c}$$

the local search set. □

Remark 4.9

1. The map $\gamma_k(\cdot, \cdot, \cdot)$ can be dynamic in the sense that if $\{x_{k_i}\}_{i=0}^I \triangleq \gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$, then the rule for selecting $x_{k_{\widehat{i}}}$, $1 \leq \widehat{i} \leq I$, can depend on $\{x_{k_i}\}_{i=0}^{\widehat{i}-1}$ and $\{f_{\epsilon_k}(x_{k_i})\}_{i=0}^{\widehat{i}-1}$. It is only important that the global search terminates after a finite number of computations, and that $\mathcal{G}_k \subset (2^{\mathbb{M}_k} \cap \mathbf{X}) \cup \emptyset$.

2. As we shall see, the global search affects only the efficiency of the algorithm but not its convergence properties. Any heuristic procedure that leads to a finite number of function evaluations can be used for $\gamma_k(\cdot, \cdot, \cdot)$.

3. The empty set is included in the range of $\gamma_k(\cdot, \cdot, \cdot)$ to allow omitting the global search of in the GPS algorithm.

4. Since the range of $\delta_{\widehat{D},k}(\cdot, \cdot)$ is $\mathbb{D}_{\widehat{D}}$, any image of $\delta_{\widehat{D},k}(\cdot, \cdot)$ is a positive spanning set.

□

4.2 A Model Adaptive Precision GPS Algorithm

We are now ready to present a model generalized pattern search algorithm with adaptive precision function evaluations.

Algorithm 4.10 (Model GPS Algorithm)

Data:	Initial iterate $x_0 \in \mathbf{X}$;
	Precision parameter $\epsilon_0 \in \mathbb{R}_+^q$;
	Initial mesh size exponent $s_0 \in \mathbb{N}$;
	Base direction matrix $\widehat{D} \in \mathbb{Q}^{n \times p} \cap \mathbb{S}$ (see Definition 4.2).
Maps:	Global search set map $\gamma_k \colon \underline{\mathbf{X}_k} \times \underline{\boldsymbol{\Delta}_k} \times \mathbb{R}_+^q \to \left(2^{M_k} \cap \mathbf{X}\right) \cup \emptyset$;
	Local search direction map $\delta_{\widehat{D},k} \colon \underline{\mathbf{X}_k} \times \underline{\boldsymbol{\Delta}_k} \to \mathbb{D}_{\widehat{D}}$ (see Definition 4.8).
	Function $\rho \colon \mathbb{R}_+ \to \mathbb{R}_+^q$ (to assign ϵ_k), such that the composition
	$\varphi \circ \rho \colon \mathbb{R}_+ \to \mathbb{R}_+$ is strictly monotone decreasing and satisfies
	$\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.
Step 0:	Initialize $k = 0$ and $\Delta_0 = 1/r^{s_0}$.
Step 1:	<u>Global Search</u>
	Construct the global search set $\mathcal{G}_k = \gamma_k(\underline{x}_k, \underline{\boldsymbol{\Delta}}_k, \epsilon_k)$.
	If $f_{\epsilon_k}(x') < f_{\epsilon_k}(x_k)$ for any $x' \in \mathcal{G}_k$, go to Step 3; else, go to Step 2.
Step 2:	<u>Local Search</u>
	Construct the direction matrix $D_k = \delta_{\widehat{D},k}(\underline{x}_k, \underline{\boldsymbol{\Delta}}_k)$.
	Construct $\mathcal{L}_k \triangleq \{x_k + \Delta_k D_k e_j \mid j = 1, \ldots, \operatorname{card}(D_k)\} \cap \mathbf{X}$ and
	evaluate $f_{\epsilon_k}(\cdot)$ for any $x' \in \mathcal{L}_k$ until some $x' \in \mathcal{L}_k$
	satisfying $f_{\epsilon_k}(x') < f_{\epsilon_k}(x_k)$ is obtained, or until all points in $\mathcal{L}_k$
	are evaluated.
Step 3:	<u>Parameter Update</u>
	If there exists an $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f_{\epsilon_k}(x') < f_{\epsilon_k}(x_k)$,
	set $x_{k+1} = x'$, $s_{k+1} = s_k$, $\Delta_{k+1} = \Delta_k$, $\epsilon_{k+1} = \epsilon_k$;
	else, set $x_{k+1} = x_k$, $s_{k+1} = s_k + t_k$, with $t_k \in \mathbb{N}_+$ arbitrary,
	$\Delta_{k+1} = 1/r^{s_{k+1}}$, $\epsilon_{k+1} = \rho(\Delta_{k+1})$.
Step 4:	Replace k by $k + 1$, and go to Step 1.

Remark 4.11

1. If the optimization is started with the precision parameter $\epsilon_0 \in \mathbb{R}_+^q$ too small, the computation time may become unnecessary large. Therefore, in implementing the Model GPS Algorithm, one may allow ϵ_k to increase arbitrarily over a preset number of iterations K.

2. Audet and Dennis [1] increase and decrease the mesh divider using the formula $\Delta_{k+1} = \tau^m \Delta_k$ where $\tau \in \mathbb{Q}$, $\tau > 1$, and m is any element of $\mathbb{Z}$. Thus, our mesh

construction is a special case of Audet's and Dennis' construction since we set $\tau = 1/r$, with $r \in \mathbb{N}_+$, $r \geq 2$ (so that, $\tau < 1$) and $m \in \mathbb{N}$. We prefer our construction because it leads to simpler geometric constructions, and because we do not see any reason why it should affect the computational performance negatively.

3. In Step 2, once a decrease of the cost function is obtained, one can proceed to Step 3. However, one is allowed to evaluate the approximating cost function at more points in $\mathcal{L}_k$ in an attempt to obtain a bigger reduction in cost. However, one is allowed to proceed to Step 3 only after either a cost decrease has been found, or after *all* points in $\mathcal{L}_k$ are tested.

4. In Step 3, we are not restricted to accepting the $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ that gives lowest cost value. But the mesh divider Δ_k is reduced *only* if there exists no $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f_{\epsilon_k}(x') < f_{\epsilon_k}(x_k)$. $\qquad\qquad\square$

4.3 An Extension of the Hooke-Jeeves Algorithm

To illustrate the use of our Model GPS Algorithm 4.10, we will now use it to obtain an extension of the Hooke-Jeeves algorithm [5]. To simplify exposition, we will assume that $\mathbf{X} = \mathbb{R}^n$.

4.3.1 Algorithm Parameters $\widehat{D}$, r, s_0 and t_k

Hooke and Jeeves decrease the "current step size" ($\Delta \in \mathbb{R}_+$ in [5]) by a factor $\rho \in (0,1)$, when necessary. To fit their algorithm into our framework, we have to set $\rho \triangleq 1/q$ for some $q \in \mathbb{N}_+ \setminus \{1\}$ [5] and restrict the initial value of their variable Δ to take on rational values only [6].

In view of the above, for our extension of the Hooke and Jeeves algorithm, we define our base direction matrix as $\widehat{D} \triangleq \Delta\,[+e_1, -e_1, \ldots, +e_n, -e_n]$ (where Δ is the *initial* value of the "step size" in [5]) and our other parameters to be $r \triangleq q$, $s_0 = 0$, and $t_k \in \{0,1\}$, for all $k \in \mathbb{N}$.

4.3.2 Map for Exploratory Moves

To facilitate the algorithm explanation, we first introduce a set-valued map $E \colon \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{R}_+^q \to 2^{M_k}$, which defines the "exploratory moves" in [5]. The map $E \colon \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{R}_+^q \to 2^{M_k}$ will then be used in Section 4.3.3 to define the global search set map and, under conditions to be seen in Section 4.3.4, the local search direction map as well.

---

[5]The Restriction $\rho \triangleq 1/q$ is not serious because one usually has no knowledge that justifies requiring another value.

[6]In numerical computer programs, the restriction $\Delta \in \mathbb{Q}_+$ is automatically fulfilled since irrational numbers cannot be represented.

Algorithm 4.12 (Map $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{R}_+^q \to 2^{M_k}$ for "Exploratory Moves")

Parameter:	Base direction matrix $\widehat{D} = \Delta\,[+e_1, -e_1, \ldots, +e_n, -e_n] \in \mathbb{Q}^{n \times 2n}$
	(Δ being the initial step size of [5]).
Input:	Base point $x \in \mathbb{R}^n$;
	Mesh divider $\Delta_k \in \mathbb{Q}_+$;
	Precision parameter $\epsilon_k \in \mathbb{R}_+^q$.
Output:	Set of trial points $\mathcal{T}$.
Step 0:	Initialize $\mathcal{T} = \emptyset$.
Step 1:	For $i = 1, \ldots, n$

$$\text{Set } \widetilde{x} = x + \Delta_k\,\widehat{D}\,e_{2i-1} \text{ and } \mathcal{T} \leftarrow \mathcal{T} \cup \{\widetilde{x}\}.$$
$$\text{If } f_{\epsilon_k}(\widetilde{x}) < f_{\epsilon_k}(x)$$
$$\text{Set } x = \widetilde{x}.$$
$$\text{else}$$
$$\text{Set } \widetilde{x} = x + \Delta_k\,\widehat{D}\,e_{2i} \text{ and } \mathcal{T} \leftarrow \mathcal{T} \cup \{\widetilde{x}\}.$$
$$\text{If } f_{\epsilon_k}(\widetilde{x}) < f_{\epsilon_k}(x)$$
$$\text{Set } x = \widetilde{x}.$$
$$\text{end if.}$$
$$\text{end if.}$$
$$\text{end for.}$$

Step 2:	Return $\mathcal{T}$.

Thus, $E(x, \Delta_k, \epsilon_k) = \mathcal{T}$.

4.3.3 Global Search Set Map $\gamma_k: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{R}_+^q \to 2^{M_k}$

We define the global search set map $\gamma_k(\cdot, \cdot, \cdot)$ as follows:

Algorithm 4.13 (Global Search Set Map $\gamma_k: \underline{\mathbf{X}}_k \times \underline{\Delta}_k \times \mathbb{R}_+^q \to 2^{M_k}$)

Map:	Map for "exploratory moves" $E: \mathbb{R}^n \times \mathbb{Q}_+ \times \mathbb{R}_+^q \to 2^{M_k}$.
Input:	Previous and current iterate, $x_{k-1} \in \mathbb{R}^n$ and $x_k \in \mathbb{R}^n$;
	Mesh divider $\Delta_k \in \mathbb{Q}_+$;
	Precision parameter $\epsilon_k \in \mathbb{R}_+^q$.
Output:	Global search set $\mathcal{G}_k$.
Step 1:	Set $x = x_k + (x_k - x_{k-1})$.
Step 2:	Compute $\mathcal{G}_k = E(x, \Delta_k, \epsilon_k)$.
Step 3:	If $\left(\min_{x \in \mathcal{G}_k} f_{\epsilon_k}(x)\right) > f_{\epsilon_k}(x_k)$
	$\qquad$ Set $\mathcal{G}_k \leftarrow \mathcal{G}_k \cup E(x_k, \Delta_k, \epsilon_k)$.
	end if.
Step 4:	Return $\mathcal{G}_k$.

Thus, $\gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k) = \mathcal{G}_k$.

4.3.4 Local Search Direction Map $\delta_{\widehat{D},k} \colon \underline{\mathbf{X}}_k \times \underline{\Delta}_k \to \mathbb{D}_{\widehat{D}}$

If the global search, as defined by Algorithm 4.13, has failed in reducing $f_{\epsilon_k}(\cdot)$, then Algorithm 4.13 has constructed a set $\mathcal{G}_k$ that contains the set $\{x_k + \Delta_k \widehat{D} \, e_i \mid i = 1, \ldots, 2n\}$. This is because in the evaluation of $E(x_k, \Delta_k, \epsilon_k)$, all "if($\cdot$)" statements yield false, and, hence, one has constructed $\{x_k + \Delta_k \widehat{D} \, e_i \mid i = 1, \ldots, 2n\} = E(x_k, \Delta_k, \epsilon_k)$.

Because the columns of $\widehat{D}$ span $\mathbb{R}^n$ positively, it follows that the search on the set $\{x_k + \Delta_k \widehat{D} \, e_i \mid i = 1, \ldots, 2n\}$ is a local search. Hence, the constructed set

$$\mathcal{L}_k \triangleq \{x_k + \Delta_k \widehat{D} \, e_i \mid i = 1, \ldots, 2n\} \subset \mathcal{G}_k \tag{14}$$

is a local search set. Consequently, $f_{\epsilon_k}(\cdot)$ has already been evaluated at all points of $\mathcal{L}_k$ (during the construction of $\mathcal{G}_k$) and, hence, one does not need to evaluate $f_{\epsilon_k}(\cdot)$ again in a local search. In view of (13c) and (14), the local search direction map is given by $D_k = \delta_{\widehat{D},k}(x_k, \Delta_k) \triangleq \widehat{D}$.

4.3.5 Parameter Update

The point x' in Step 3 of the GPS Model Algorithm 4.10 corresponds to $x' \triangleq \arg\min_{x \in \mathcal{G}_k} f_{\epsilon_k, \mathbf{X}}(x)$ in the Hooke-Jeeves algorithm. (Note that $\mathcal{L}_k \subset \mathcal{G}_k$ if a local search has been done as explained in the above paragraph.)

5 Convergence Results

5.1 Unconstrained Minimization

We will now establish the convergence properties of the Model GPS Algorithm 4.10 on unconstrained minimization problems, i.e., for $\mathbf{X} = \mathbb{R}^n$.

First, we will show that for any mesh $\mathbb{M}_k$, the minimal Euclidean distance between all distinct mesh points is bounded from below by a constant times the mesh divider Δ_k.

Lemma 5.1 (Minimal Distance between Distinct Mesh Points) *Consider the sequences $\{\Delta_k\}_{k=0}^{\infty} \subset \mathbb{Q}_+$ of mesh dividers, and $\{\mathbb{M}_k\}_{k=0}^{\infty}$ of meshes, constructed by Model GPS Algorithm 4.10. Then there exists a constant $c > 0$, independent of k, such that*

$$\min_{\substack{u \neq v \\ u,v \in \mathbb{M}_k}} \|u - v\| \geq \Delta_k \, c. \tag{15}$$

Proof. By Definition 4.4, for any given k, we have $\mathbb{M}_k \triangleq \{x_0 + \Delta_k \widehat{D} \, m \mid m \in \mathbb{N}^p\}$ where $\widehat{D} \in \mathbb{Q}^{n \times p}$ is fixed for all k. Let l be the least common multiple of all denominators of the

elements of $\widehat{D}$. Then, $\widehat{Z} \triangleq l\,\widehat{D}$ is in $\mathbb{Z}^{n \times p}$. Furthermore, any pair of mesh points u, v can be represented as $u \triangleq x_0 + \Delta_k\,\widehat{D}\,m_u$ and $v \triangleq x_0 + \Delta_k\,\widehat{D}\,m_v$, where $m_u, m_v \in \mathbb{N}^p$. Hence,

$$
\begin{aligned}
\min_{\substack{u \neq v \\ u,v \in \mathbb{M}_k}} \|u - v\| &= \min_{\substack{\|\widehat{D}(m_u - m_v)\| \neq 0 \\ m_u, m_v \in \mathbb{N}^p}} \Delta_k \|\widehat{D}(m_u - m_v)\| \\
&= \min_{\substack{\|\widehat{D}\,m\| \neq 0 \\ m \in \mathbb{Z}^p}} \Delta_k \|\widehat{D}\,m\| = \frac{\Delta_k}{l} \min_{\substack{\|\widehat{Z}\,m\| \neq 0 \\ m \in \mathbb{Z}^p}} \|\widehat{Z}\,m\| \geq \frac{\Delta_k}{l}.
\end{aligned}
\tag{16}
$$

The inequality holds because $\widehat{Z}\,m$ is a nonzero integer vector. $\qquad\square$

The following corollary follows directly from Lemma 5.1 and will be used to show that $\Delta_k \to 0$ as $k \to \infty$.

Corollary 5.2 *Any bounded subset of a mesh* $\mathbb{M}_k$, *constructed by Model GPS Algorithm 4.10, contains only a finite number of mesh points.* $\qquad\square$

Proposition 5.3 *Consider the sequence of mesh dividers* $\{\Delta_k\}_{k=0}^{\infty} \subset \mathbb{Q}_+$ *constructed by Model GPS Algorithm 4.10. Then, the mesh dividers satisfy* $\liminf_{k \to \infty} \Delta_k = 0$.

Proof. By (7), $\Delta_k = 1/r^{s_k}$, where $r \in \mathbb{N}$ with $r > 1$, and $\underline{s}_k \subset \mathbb{N}$ is a nondecreasing sequence. For the sake of contradiction, suppose that there exists a $\Delta_{k^*} \in \mathbb{Q}_+$, such that $\Delta_k \geq \Delta_{k^*}$ for all $k \in \mathbb{N}$. Then there exists a corresponding $s_{k^*} = \max_{k \in \mathbb{N}} s_k$, and the finest possible mesh is $\mathbb{M}_{k^*} \triangleq \{x_0 + (1/r^{s_{k^*}})\,\widehat{D}\,m \mid m \in \mathbb{N}^p\}$.
Next, since by Assumption 3.4, there exists a compact set $\mathbf{C}$, such that $\mathbf{L}_{f_{\epsilon_0}(x_0)}(f_{\epsilon_k}) \cap \mathbf{X} \subset \mathbf{C}$ for all $k \in \mathbb{N}$, it follows from Corollary 5.2 that $\mathbb{M}_{k^*} \cap \mathbf{L}_{f_{\epsilon_0}(x_0)}(f_{\epsilon_k})$ contains only a finite number of points for any $k \in \mathbb{N}$. Thus, at least one point in $\mathbb{M}_{k^*}$ must belong to the sequence $\{x_k\}_{k=0}^{\infty}$ infinitely many times. Furthermore, because $\{s_k\}_{k=0}^{\infty} \subset \mathbb{N}$ is nondecreasing with s_{k^*} being its maximal element, it follows that $\epsilon_k = \epsilon_{k^*}$ for all $k \geq k^*$. Hence the sequence $\{f_{\epsilon_{k^*}}(x_k)\}_{k=0}^{\infty}$ cannot be strictly monotone decreasing, which contradicts the constructions in Algorithm 4.10. $\qquad\square$

Having shown that $\liminf_{k \to \infty} \Delta_k = 0$, we can introduce the notion of a refining subsequence as used by Audet and Dennis [1].

Definition 5.4 (Refining Subsequence) *Consider a sequence* $\{x_k\}_{k=0}^{\infty}$ *constructed by Model GPS Algorithm 4.10. We will say that the subsequence* $\{x_k\}_{k \in \mathbb{K}}$ *is the refining subsequence, if* $\Delta_{k+1} < \Delta_k$ *for all* $k \in \mathbb{K}$, *and* $\Delta_{k+1} = \Delta_k$ *for all* $k \notin \mathbb{K}$. $\qquad\square$

When the cost function $f(\cdot)$ is only locally Lipschitz continuous, we, as well as Audet and Dennis [1], only get a weak characterization of limit points of refining sequences. As we will now see.

We recall the definition of Clarke's generalized directional derivative [3]:

Definition 5.5 (Clarke's Generalized Directional Derivative) *Let* $f\colon \mathbb{R}^n \to \mathbb{R}$ *be locally Lipschitz continuous at the point* $x^* \in \mathbb{R}^n$. *Then,* Clarke's *generalized directional derivative of* $f(\cdot)$ *at* x^* *in the direction* $h \in \mathbb{R}^n$ *is defined by*

$$\mathrm{d}^0 f(x^*; h) \triangleq \limsup_{\substack{x \to x^* \\ t \downarrow 0}} \frac{f(x + t\,h) - f(x)}{t}.$$ (17)

□

Theorem 5.6 *Suppose that Assumptions 3.1 and 3.4 are satisfied and let* $x^* \in \mathbb{R}^n$ *be an accumulation point of a refining subsequence* $\{x_k\}_{k \in K}$, *constructed by Model GPS Algorithm 4.10. Let d be any column of the base direction matrix* $\widehat{D}$ *along which* $f_{\epsilon_k}(\cdot)$ *was evaluated for infinitely many iterates in the subsequence* $\{x_k\}_{k \in K}$. *Then,*

$$\mathrm{d}^0 f(x^*; d) \geq 0.$$ (18)

Proof. Let $\{x_k\}_{k \in K}$ be the refining subsequence and, WLOG, suppose that $x_k \to^K x^*$. By Assumption 3.4, there exists a compact set C such that $L_{f_{\epsilon_0}(x_0)}(f_{\epsilon_k}) \cap X \subset C$ for all $k \in \mathbb{N}$. Therefore, by Assumption 3.1, there exists an $\epsilon_L \in \mathbb{R}_+^q$ and a scalar $K_L \in (0, \infty)$ such that, for all $x \in C$ and for all $0 < \epsilon \leq \epsilon_L$, we have $|f_\epsilon(x) - f(x)| \leq K_L\, \varphi(\epsilon)$. Because $f(\cdot)$ is locally Lipschitz continuous, its directional derivative $\mathrm{d}^0 f(\cdot\,;\cdot)$ exists. Hence,

$$
\begin{aligned}
\mathrm{d}^0 f(x^*; d) \ \triangleq\ & \limsup_{\substack{x \to x^* \\ t \downarrow 0}} \frac{f(x + t\,d) - f(x)}{t} \\[2mm]
\geq\ & \limsup_{k \in K} \frac{f(x_k + \Delta_k\,d) - f(x_k)}{\Delta_k} \\[2mm]
\geq\ & \limsup_{k \in K} \frac{f_{\epsilon_k}(x_k + \Delta_k\,d) - f_{\epsilon_k}(x_k) - 2\,K_L\,\varphi(\epsilon_k)}{\Delta_k} \\[2mm]
\geq\ & \limsup_{k \in K} \frac{f_{\epsilon_k}(x_k + \Delta_k\,d) - f_{\epsilon_k}(x_k)}{\Delta_k} - \limsup_{k \in K} 2\,K_L\,\frac{\varphi(\epsilon_k)}{\Delta_k} \\[2mm]
\geq\ & -\limsup_{k \in K} 2\,K_L\,\frac{\varphi(\epsilon_k)}{\Delta_k}.
\end{aligned}
$$ (19)

The last inequality holds because $\{x_k\}_{k \in K}$ is a refining subsequence. Since by Proposition 5.3, $\Delta_k \to 0$, it follows from the constructions in Model GPS Algorithm 4.10 that $\varphi(\epsilon_k)/\Delta_k \to^K 0$. □

Remark 5.7 Note that (18) is not a standard optimality condition since it holds only for certain directions d. Consider, for example, the Lipschitz continuous functions

$$f(x) \triangleq \begin{cases} \|x\|, & \text{if } x^1 > 0 \text{ and } x^2 > 0, \\ \|x\|\,\cos\!\big(4 \arccos\big(x^1/\|x\|\big)\big), & \text{otherwise,} \end{cases}$$ (20)
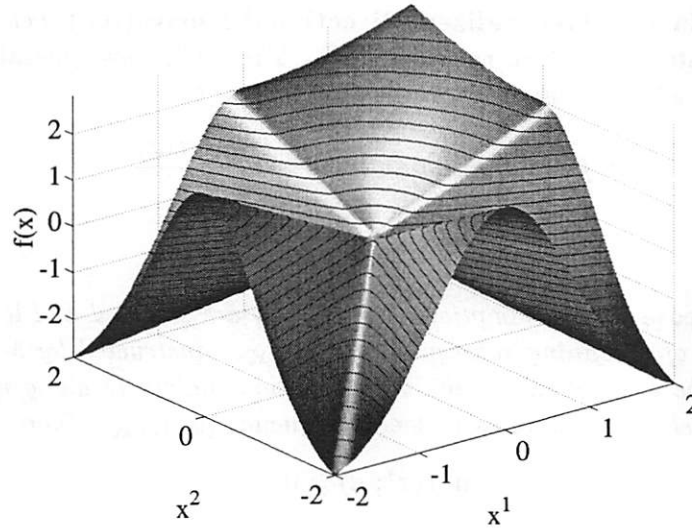
Figure 3: Visualization of equation (20)

which is shown in Fig. 3. This function is not differentiable at the origin, but it does have directional derivatives everywhere. At the origin $x^* = 0$, we have $\mathrm{d}f(x^*; d) = 1$ for $d \in \{\pm e_1, \pm e_2\}$, but the directional derivative along $s = (-1, -1)^T$ is $\mathrm{d}f(x^*; s) = -\sqrt{2}$. Using the Hooke-Jeeves algorithm with initial value $x_0 = (-1, 0)^T$, we would converge to the origin, a point that possess some negative directional derivatives. □

We now state that pattern search algorithms with adaptive precision function evaluations converge to stationary points.

Theorem 5.8 (Convergence to a Stationary Point) *Suppose that Assumptions 3.1 and 3.4 are satisfied and, in addition, that $f(\cdot)$ is once continuously differentiable. Let $x^* \in \mathbb{R}^n$ be an accumulation point of a refining subsequence $\{x_k\}_{k \in \mathbf{K}}$, constructed by Model GPS Algorithm 4.10. Then,*

$$\nabla f(x^*) = 0. \tag{21}$$

Proof. Since $f(\cdot)$ is once continuously differentiable, we have $\mathrm{d}^0 f(x^*; h) = \mathrm{d}f(x^*; h) = \langle \nabla f(x^*), h \rangle$. Now, let $\mathbb{D}_{\widehat{D}}$ be the set of all candidates for the direction matrix, and let $D^* \in \mathbb{D}_{\widehat{D}}$ be any positive spanning matrix that is used infinitely many times in conjunction with the refining subsequence $\{x_k\}_{k \in \mathbf{K}}$. Since the number of distinct columns in $\mathbb{D}_{\widehat{D}}$ is finite, there must be at least one such D^*. It follows from Theorem 5.6 that $0 \leq \langle \nabla f(x^*), d^* \rangle$ for all $d^* \in D^*$. Let l denote the number of columns of D^*. Then, because the columns of D^* positively span $\mathbb{R}^n$, we can express any $h \in \mathbb{R}^n$, as follows,

$$h = \sum_{i=1}^{l} \alpha_i d_i^*, \quad d_i^* \in D^*, \quad \alpha_i \geq 0, \quad \forall i \in \{1, \ldots, l\}. \tag{22a}$$

Hence, $0 \leq \langle \nabla f(x^*), h \rangle$. Similarly, we can express the vector $-h$, as follows,

$$-h = \sum_{i=1}^{l} \beta_i d_i^*, \quad d_i^* \in D^*, \quad \beta_i \geq 0, \quad \forall i \in \{1, \ldots, l\}. \tag{22b}$$

Hence, $0 \geq \langle \nabla f(x^*), h \rangle$, which implies $0 = \langle \nabla f(x^*), h \rangle$, and, since h is arbitrary, that $\nabla f(x^*) = 0$. □

5.2 Linearly Constrained Minimization

We now extend our convergence proofs to the linearly constrained problem (1), by following the arguments in Audet and Dennis [1].

First, we introduce the notion of a tangent cone and a normal cone, which are defined as follows:

Definition 5.9 (Tangent and Normal Cone)

1. Let $\mathbf{X} \subset \mathbb{R}^n$. Then, we define the tangent cone to $\mathbf{X}$ at a point $x^* \in \mathbf{X}$ by

$$\mathbf{T_X}(x^*) \triangleq \overline{\{\mu(x - x^*) \mid \mu \geq 0, x \in \mathbf{X}\}}. \tag{23a}$$

2. Let $\mathbf{T_X}(x^*)$ be as above. Then, we define the normal cone to $\mathbf{X}$ at $x^* \in \mathbf{X}$ by

$$\mathbf{N_X}(x^*) \triangleq \{v \in \mathbb{R}^n \mid \forall t \in \mathbf{T_X}(x^*), \langle v, t \rangle \leq 0\}. \tag{23b}$$

□

Next, we introduce the concept of conformity of a pattern to a constraint set (see [1]), which will enable us to extend the convergence results for our Model GPS Algorithm 4.10 from unconstrained optimization problems to linearly constrained optimization problems.

Definition 5.10 *The function $\delta_{\widehat{D},k} \colon \underline{\mathbf{X}}_k \times \underline{\Delta}_k \to \mathbb{D}_{\widehat{D}}$ is said to conform to the feasible set $\mathbf{X}$, if for some $\rho > 0$ and for each $x^* \in \partial \mathbf{X}$ satisfying $\|x^* - x_k\| < \rho$, the tangent cone $\mathbf{T_X}(x^*)$ can be generated by nonnegative linear combinations of the columns of a subset $D_{x^*}(x_k) \subset D_k = \delta_{\widehat{D},k}(\underline{x}_k, \underline{\Delta}_k)$.*
Furthermore, we define $D_{x^*}(\cdot)$ to be such that all its columns belong to $\mathbf{T_X}(x^*)$. □

Remark 5.11 The definition that all columns of $D_{x^*}(\cdot)$ belong to $\mathbf{T_X}(x^*)$ facilitates the extension of Theorem 5.6 to the constraint case. □

We can now state our final result which states that the accumulation points generated by Model GPS Algorithm 4.10 are feasible stationary points of problem (1).

Theorem 5.12 (Convergence to a Feasible Stationary Point)
Suppose Assumptions 3.1 and 3.4 are satisfied and that $f(\cdot)$ is once continuously differentiable. Let $x^* \in \mathbf{X}$ be an accumulation point of a refining subsequence $\{x_k\}_{k \in \mathbf{K}}$ constructed by Model GPS Algorithm 4.10 in solving problem (1).

If there exists a $k^* \in \mathbb{N}$ such that for all $k > k^*$, the local search direction maps $\delta_{\widehat{D},k} \colon \underline{\mathbf{X}}_k \times \underline{\boldsymbol{\Delta}}_k \to \mathbb{D}_{\widehat{D}}$ conform to the feasible set $\mathbf{X}$, then

$$\langle \nabla f(x^*), t \rangle \geq 0, \qquad \forall\, t \in \mathbf{T_X}(x^*), \tag{24a}$$

and

$$-\nabla f(x^*) \in \mathbf{N_X}(x^*). \tag{24b}$$

Proof. If x^* is in the interior of $\mathbf{X}$, then the result reduces to Theorem 5.8.

Let $x^* \in \partial \mathbf{X}$ and let $D_{x^*}(x_k)$ be as in Definition 5.10. Since the family of maps $\left\{\delta_{\widehat{D},k}(\cdot,\cdot)\right\}_{k \in \mathbf{K}, k > k^*}$ conforms to the feasible set $\mathbf{X}$ and since there are only finitely many linear constraints, we have that $D_{x^*}(x_k)$ converges to $D_{x^*}(x^*)$, as $x_k \to^{\mathbf{K}'} x^*$, for some infinite subset $\mathbf{K}' \subset \mathbf{K}$. By Theorem 5.6, we have $\langle \nabla f(x^*), d \rangle \geq 0$ for all $d \in D_{x^*}(x^*)$. Furthermore, it follows from the conformity of the family of local search direction maps $\left\{\delta_{\widehat{D},k}(\cdot,\cdot)\right\}_{k \in \mathbf{K}, k > k^*}$, that every $t \in \mathbf{T_X}(x^*)$ is a nonnegative linear combination of columns of $D_{x^*}(x^*)$. Therefore, $\langle \nabla f(x^*), t \rangle \geq 0$. It follows directly that $\langle -\nabla f(x^*), t \rangle \leq 0$, which shows that $-\nabla f(x^*) \in \mathbf{N_X}(x^*)$. □

When the function $f(\cdot)$ is only locally Lipschitz continuous, we obtain following corollary which follows directly from Theorem 5.6 and equation (24a).

Corollary 5.13 *Suppose that the assumptions of Theorem 5.12 are satisfied, but $f(\cdot)$ were only locally Lipschitz continuous. Then,*

$$\mathrm{d}^0 f(x^*; d) \geq 0, \qquad \forall\, d \in D_{x^*}(x^*). \tag{25}$$

□

6 Conclusion

We have extended the family of GPS algorithms to a form capable of solving optimization problems in which the cost function $f(\cdot)$ cannot be evaluated exactly, but can be approximated by a family of functions $\{f_{\epsilon_k}(\cdot)\}_{k=0}^{\infty}$. An important feature of our algorithms is that they use presumably low-cost, coarse precision approximations to the cost function when far from a solution, with the precision progressively increased as a solution is approached. This feature is known to lead to considerable time savings over using very high precision approximations to the cost function in all iterations.

In constructing our algorithms, we have adopted a geometric framework that should be easier to grasp than that found in earlier versions of GPS algorithms, but which is not prejudicial to numerical efficiency of the resulting algorithms.

References

[1] Charles Audet and J. E. Dennis Jr. Analysis of generalized pattern search. Technical Report TR 00-07, Rice University, Houston, Department of Computational and Applied Mathematics, 2000.

[2] Andrew J. Booker, J. E. Dennis Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, and Michael W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, Feb. 1999.

[3] F. H. Clarke. *Optimization and nonsmooth analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[4] Chandler Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76(4):733–746, Oct. 1954.

[5] R. Hooke and T. A. Jeeves. 'Direct search' solution of numerical and statistical problems. *J. Assoc. Comp. Mach.*, 8(2):212–229, Apr. 1961.

[6] J. E. Dennis Jr. and V. Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.

[7] Robert Michael Lewis and Virginia Torczon. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099, 1999.

[8] Robert Michael Lewis and Virginia Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[9] Elijah Polak. *Computational methods in optimization; a unified approach*, volume 77 of *Mathematics in science and engineering*. New York, Academic Press, 1971.

[10] David B. Serafini. *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*. PhD thesis, Rice University, 1998.

[11] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.