SIAM J. OPTIM.
Vol. 17, No. 4, pp. 943–968

STATIONARITY RESULTS FOR GENERATING SET SEARCH FOR LINEARLY CONSTRAINED OPTIMIZATION*

TAMARA G. KOLDA†, ROBERT MICHAEL LEWIS‡, AND VIRGINIA TORCZON§

Abstract. We present a new generating set search (GSS) approach for minimizing functions subject to linear constraints. GSS is a class of direct search optimization methods that includes generalized pattern search. One of our main contributions in this paper is a new condition to define the set of conforming search directions that admits several computational advantages. For continuously differentiable functions we also derive a bound relating a measure of stationarity, which is equivalent to the norm of the gradient of the objective in the unconstrained case, and a parameter used by GSS algorithms to control the lengths of the steps. With the additional assumption that the derivative is Lipschitz, we obtain a big-O bound. As a consequence of this relationship, we obtain subsequence convergence to a KKT point, even though GSS algorithms lack explicit gradient information. Numerical results indicate that the bound provides a reasonable estimate of stationarity.

Key words. constrained optimization, linear constraints, global convergence analysis, direct search, generating set search, generalized pattern search, derivative-free methods, stopping criteria

AMS subject classifications. 90C56, 90C30, 65K05

DOI. 10.1137/S1052623403433638

1. Introduction.

We consider a class of direct search methods called *generating set search* (GSS) [15] which encompasses methods such as generalized pattern search [33, 18, 19] and certain classes of derivative-free optimization methods [21, 22, 23, 24]. The problem of interest is the linearly constrained minimization problem:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && Ax \le b. \end{aligned} \tag{1.1}$$

Here $f : \mathbb{R}^n \to \mathbb{R}$, A is an $m \times n$ matrix, and b is a vector in $\mathbb{R}^m$. Both A and b are assumed to be explicitly available. No assumption of nondegeneracy of the constraints is made. Let Ω denote the feasible region

$$\Omega = \{\, x \;\mid\; Ax \le b \,\}.$$

We assume that the objective f is continuously differentiable on Ω but that the gradient is not computationally available because no procedure exists for computing the gradient and it cannot be approximated accurately.

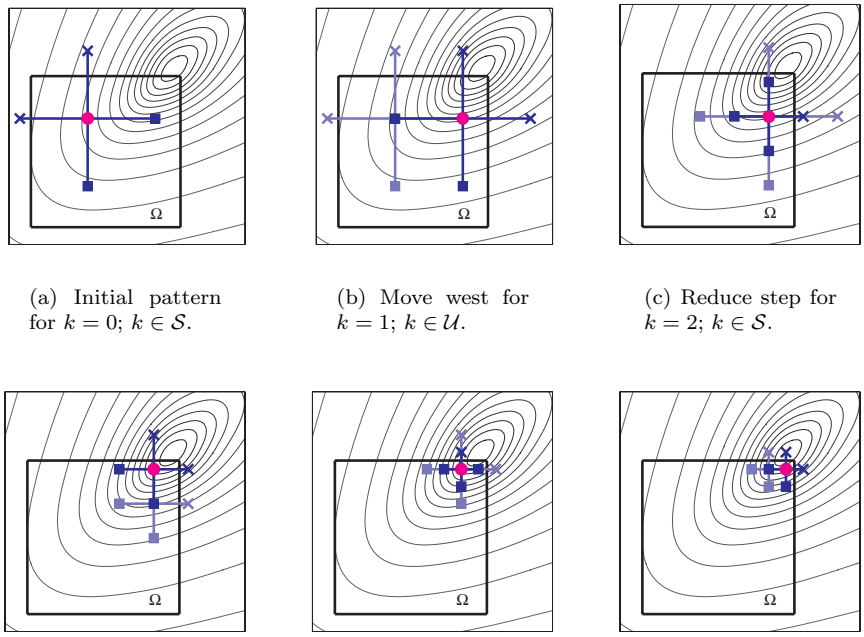
*Received by the editors August 21, 2003; accepted for publication (in revised form) April 24, 2006; published electronically November 22, 2006.

http://www.siam.org/journals/siopt/17-4/43363.html

†Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94551-9217 (tgkolda@sandia.gov). This work was supported by the Mathematical, Information, and Computational Sciences Program of the U.S. Department of Energy, under contract DE-AC04-94AL85000 with Sandia Corporation.

‡Department of Mathematics, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187-8795 (buckaroo@math.wm.edu). This research was supported by the Computer Science Research Institute at Sandia National Laboratories.

§Department of Computer Science, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187-8795 (va@cs.wm.edu). This research was supported by the Computer Science Research Institute at Sandia National Laboratories.



(a) Initial pattern for $k = 0$; $k \in \mathcal{S}$.

(b) Move west for $k = 1$; $k \in \mathcal{U}$.

(c) Reduce step for $k = 2$; $k \in \mathcal{S}$.

(d) Move north for $k = 3$; $k \in \mathcal{U}$.

(e) Reduce step for $k = 4$; $k \in \mathcal{S}$.

(f) Move west for $k = 5$; $k \in \mathcal{U}$.

FIG. 1.1. *Coordinate search with exact penalization applied to the modified Broyden tridiagonal function with bound constraints.*

1.1. An illustrative example. We illustrate an instance of a GSS method in Figure 1.1. We consider coordinate search applied to the two-dimensional modified Broyden tridiagonal function [4, 26], a standard test problem, with the addition of bounds on the variables. Level curves of the function are shown in the background, and the feasible region is the box labeled Ω. The current iterate x_k is indicated by a circle; this is the point with the lowest value of f found so far, also known as the *best point*. If there are no constraints, a coordinate search method evaluates the function at the $2n$ *trial points* defined by taking a step of a specified length from x_k along the positive and negative coordinate directions, i.e., the *search directions*. The iterates must remain feasible with respect to the bound constraints present in this problem, which means that infeasible trial points are not considered. Terminal crosses show infeasible trial points; solid squares indicate feasible trial points. The lighter versions given in (b)–(f) indicate the search directions and trial points from the previous iteration.

To establish notation and give context for the discussion that follows, we give an outline of a GSS method. Details are developed throughout the paper; complete statements of the algorithms can be found in section 5.

Let $x_0 \in \Omega$ be the initial iterate, and let Δ_0 be the initial choice for the *step-length control parameter* with $\Delta_0 > \Delta_{\text{tol}} > 0$, where Δ_{tol} serves as a measure for termination. The search proceeds for iterations $k = 0, 1, 2, \ldots$ until $\Delta_k < \Delta_{\text{tol}}$.

The first step in each iteration is to select a set of search directions. The number

of search directions is denoted by p_k and the set of search directions by

$$\mathcal{D}_k = \{d_k^{(1)}, \ldots, d_k^{(p_k)}\}.$$

The second step in each iteration is to construct feasible trial points of the form

$$x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)}, \quad i \in \{1, \ldots, p_k\},$$

with $\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$ chosen to ensure feasibility. These trial points are where the objective function may be evaluated in the search for a new best point to replace x_k.

The third step is to determine whether the iteration is successful or unsuccessful and correspondingly update x and Δ. If one of the trial points reduces the objective function value by an acceptable amount, then that trial point becomes the new iterate x_{k+1}. The step-length control parameter may either be increased or, more usually, left unchanged so that $\Delta_{k+1} = \Delta_k$. In this case the iteration is deemed *successful* and k is assigned to the set of successful iterates denoted by $\mathcal{S}$. Otherwise, none of the trial points improves the value of the objective function, so the step Δ_k is reduced, e.g., $\Delta_{k+1} = \frac{1}{2}\Delta_k$, and the next iterate is unchanged, i.e., $x_{k+1} = x_k$. In this case the iteration is deemed *unsuccessful* and k is assigned to the set of unsuccessful iterates denoted by $\mathcal{U}$.

1.2. Goals of this paper. A primary contribution of this paper is a new condition on the set of search directions $\mathcal{D}_k$ that is flexible but also sufficient to ensure desirable convergence properties of the algorithm. Key to our new results is the way in which the classification of constraints as being nearly binding is tied to Δ_k, the step-length control parameter.

The following measure of stationarity, introduced in [5], is central to our analysis: for $x \in \Omega$,

$$\chi(x) \equiv \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} -\nabla f(x)^T w.$$

As discussed in [6], $\chi(x)$ is a continuous function on Ω. Furthermore, $\chi(x) = 0$ for $x \in \Omega$ if and only if x is a Karush–Kuhn–Tucker (KKT) point of the linearly constrained problem.

In Theorem 6.4, under certain assumptions, we show that at unsuccessful iterations there is a big-O relationship between the step-length control parameter and the measure of stationarity:

$$\chi(x_k) = O(\Delta_k) \quad \text{for } k \in \mathcal{U}. \tag{1.2}$$

This means that as Δ_k is reduced, the upper bound on the value of the measure of stationarity is also reduced. Relationship (1.2) is analogous to the unconstrained minimization result (see [8, section 3] or [15, section 3.6]):

$$\| \nabla f(x_k) \| = O(\Delta_k) \quad \text{for } k \in \mathcal{U}. \tag{1.3}$$

Results (1.2) and (1.3) support using the magnitude of Δ_k as a test for termination. In section 7 we give numerical illustrations of relationship (1.2).

Another consequence of (1.2) is that it leads directly to a global convergence result (Theorem 6.5) showing that a subsequence of the iterates converges to a KKT point:

$$\liminf_{k \to \infty} \chi(x_k) = 0. \tag{1.4}$$

The latter follows immediately from (1.2) once the result $\liminf_{k \to \infty} \Delta_k = 0$ from [33] is invoked, thus further simplifying prior global convergence analyses.

1.3. Related work. The GSS methods we propose for solving linearly constrained problems are *feasible-point* methods; i.e., they require all iterates to be feasible. They also share many features with classical feasible directions methods that rely on derivatives [2, 35, 36], especially in the way in which they handle the proximity of the current iterate to the boundary of the feasible region.

Most prior related work has used similar mechanisms for identifying the set of nearly binding linear constraints [25, 34, 19, 1, 30] and [24, Algorithm 2]. Constraints were identified as being nearly binding by considering either the Euclidean distance from the current iterate to the constraint faces [25, 19, 24, 1] or the magnitude of the constraint residual $|\, a_i^T x - b_i \,|$ at the current iterate [34, 30]. A constraint was treated as binding if one of the preceding measures fell below some fixed threshold.

The convergence properties of GSS algorithms rely on the presence at each iteration of a theoretically necessary set of search directions, which we call *core directions*. In the work just cited ([25, 34, 19, 1, 30] and [24, Algorithm 2]), the core directions are all the generators for a *set* of cones. There are situations where the resulting number of search directions is quite large. Since Δ_k can be reduced only at the conclusion of an unsuccessful iteration, and each unsuccessful iteration requires the evaluation of the function at the trial points defined by core directions, there is incentive to try and keep the cardinality of the set of core directions small when the cost of computing f at a feasible point is appreciable.

Algorithm 1 of [24] addresses this concern. Its core directions are the generators of a single cone. However, the *only* allowable search directions are the core directions—the set of search directions cannot be augmented.

The approach we advocate here is a compromise. Our set of core directions is smaller than in [25, 34, 19, 1, 30] and [24, Algorithm 2], but the choice of search directions is more flexible than Algorithm 1 of [24]. The core set need only contain generators for a single cone, but accommodates additional search directions. As reported in [17], the computational advantages of this compromise are appreciable in terms of reducing the number of search directions per iteration, reducing the total number of iterations, and reducing the total number of function evaluations.

Another focus of the work reported here is on establishing (1.2) and a related result regarding the projection of the direction of steepest descent onto the polar of the cone defined by the working set of constraints. Proposition 7.1 in [19] also established a relationship between Δ_k and a different measure of stationarity. The quantity

$$q(x) \equiv P_\Omega\left(x - \nabla f(x)\right) - x, \tag{1.5}$$

where P_Ω denotes the projection onto Ω, is a continuous function of x with the property that $q(x) = 0$ for $x \in \Omega$ if and only if x is a KKT point. In [19, Proposition 7.1] it is shown that

$$\| \, q(x_k) \, \| = O(\sqrt{\Delta_k}) \quad \text{for } k \in \mathcal{U}, \tag{1.6}$$

a result that is neither as satisfying nor as useful as that in (1.2).

Continuing along the lines we began in [15], here we incorporate the sufficient decrease step acceptance criterion from [23, 22, 24], while also preserving a version of the algorithm that requires only simple decrease, as in the work in [19, 1, 30]. The sufficient decrease condition simplifies the analysis. More importantly, the sufficient decrease condition gives us greater flexibility in how we maintain feasibility in the

presence of linear constraints. In particular, using a sufficient decrease acceptance criterion makes steps onto the boundary straightforward.

As mentioned in section 1.2, given (1.2) it is straightforward to prove convergence of a subsequence to a KKT point. The approach to convergence analysis in [1, 30] takes a different tack by focusing on the directional derivatives along the search directions and considering whether limit points of the sequence of iterates are KKT points. This allows a relaxation of the smoothness assumptions on f. If f is not assumed to be continuously differentiable, but is only assumed to be strictly differentiable at limit points of the sequence of iterates, the results in [1, 30] show that those limit points are KKT points. However, subsequence convergence to KKT points in the nonsmooth case is not guaranteed by the results in [1, 30] and, in fact, may not be realized [15].

1.4. Organization. The paper is organized as follows. In section 2, we describe the conditions on the set of core directions for GSS methods applied to problems with linear constraints. As we saw in Figure 1.1, GSS algorithms may generate trial points that are infeasible, so in section 3 we describe how feasibility is maintained. In section 4 we discuss the globalization strategies. Formal statements of GSS algorithms for solving linearly constrained problems are given in section 5. We present two general algorithms. The first (Algorithm 5.1) uses a sufficient decrease condition as in [22, 24]. The second (Algorithm 5.2) uses a simple decrease condition as in [18, 19]. Results showing the stationarity properties of these algorithms are derived in section 6. In section 7 we discuss what the analysis reveals about using Δ_k to test for stationarity and demonstrate its effectiveness on two test problems. In section 8, we summarize the results and their importance. Appendix A contains a discussion of $\chi(x)$ and its use as a measure of stationarity. Appendix B contains geometric results on cones and polyhedra.

2. Search directions. GSS methods for linearly constrained optimization need to choose $\mathcal{D}_k$, the set of search directions, at each iteration. In this section, we describe the conditions we place on $\mathcal{D}_k$ to guarantee (1.2), and thus (1.4). Since GSS methods do not use gradient information, they cannot directly identify descent directions. Instead, the set $\mathcal{D}_k$ must include enough search directions to guarantee that at least one of them is a descent direction and, moreover, allows a sufficiently long step within the feasible region if x_k is not a KKT point. To describe the conditions on the sets of search directions, we start in section 2.1 by reviewing some standard concepts regarding finitely generated cones. Then, in section 2.2, we show how to use the constraints $Ax \leq b$ to define cones that mirror the geometry of the boundary of the polyhedron Ω near the current iterate x_k. Finally, in section 2.3, we detail the conditions placed on the set $\mathcal{D}_k$ to ensure that, for every iteration of any GSS algorithm, there exists at least one direction along which it is possible to take a step of sufficient length while remaining inside Ω.

2.1. Cones and generators. A *cone* K is a set that is closed under nonnegative scalar multiplication, i.e., K is a cone if $x \in K$ implies $\alpha x \in K$ for all $\alpha \geq 0$. The *polar* of a cone K, denoted K°, is defined by

$$K^\circ = \{\ v \ \mid\ w^T v \leq 0 \text{ for all } w \in K\ \}$$

and is itself a cone. Given a convex cone K and any vector v, there is a unique closest point of K to v, the *projection* of v onto K, which we denote by v_K. Given a vector v and a convex cone K, any vector v can be written as $v = v_K + v_{K^\circ}$ and $v_K^T v_{K^\circ} = 0$ [27, 12].

A set of vectors $\mathcal{G}$ *generates* a cone K if K is the set of all nonnegative linear combinations of elements of $\mathcal{G}$. A cone K is *finitely generated* if it can be generated by a finite set of vectors. For any finite set of vectors $\mathcal{G}$, we define

$$\kappa(\mathcal{G}) = \inf_{\substack{v \in \mathbb{R}^n \\ v_K \neq 0}} \max_{d \in \mathcal{G}} \frac{v^T d}{\| v_K \| \, \| d \|}, \quad \text{where } K \text{ is the cone generated by } \mathcal{G}. \tag{2.1}$$

This is a generalization of the quantity given in [15, (3.10)], where $\mathcal{G}$ generates $\mathbb{R}^n$. Note that the value $\kappa(\mathcal{G})$ is a property of the set $\mathcal{G}$—not of the cone K. See Proposition 10.3 in [19] for a proof of the following result.

PROPOSITION 2.1. *If* $\mathcal{G} \neq \{\mathbf{0}\}$, *then* $\kappa(\mathcal{G}) > 0$.

A special case occurs if $\mathcal{G}$ generates $\mathbb{R}^n$. In this case, a set of generators is a *positive spanning set* [7]. Thus a positive spanning set is like a linear spanning set but with the additional requirement that all the coefficients be nonnegative. One particular choice of generating set for $\mathbb{R}^n$ is the set of the positive and negative unit coordinate vectors

$$\{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\},$$

which is the set of search directions used for the illustration of coordinate search in Figure 1.1.

2.2. Tangent and normal cones. Let a_i^T be the ith row of the constraint matrix A and let

$$\mathcal{C}_i = \{\ y \ \mid \ a_i^T y = b_i \ \}$$

denote the set where the ith constraint is binding. The set of indices for the binding constraints at x is $I(x) = \{\ i \ \mid \ x \in \mathcal{C}_i \ \}$. The *normal cone* at a point x, denoted by $N(x)$, is the cone generated by the binding constraints, i.e., the cone generated by the set $\{\ a_i \ \mid \ i \in I(x) \ \} \cup \{\mathbf{0}\}$. The presence of $\{\mathbf{0}\}$ means that $N(x) = \{\mathbf{0}\}$ if there are no binding constraints. The *tangent cone*, denoted by $T(x)$, is the polar of the normal cone. Further discussion of the tangent and polar cones in the context of optimization can be found, for instance, in [31, 12, 13, 29].

In our case, we are not only interested in the binding constraints, but also in the nearby constraints. Given $x \in \Omega$, the indices of the *ε-binding constraints* are given by

$$I(x, \varepsilon) = \{\ i \ \mid \ \operatorname{dist}(x, \mathcal{C}_i) \leq \varepsilon \ \}. \tag{2.2}$$

The vectors a_i for $i \in I(x, \varepsilon)$ are the outward-pointing normals to the faces of the boundary of Ω within distance ε of x. The idea of using ε-binding constraints is identical to one sometimes used in gradient-based feasible directions methods, e.g., [2, section 2.5].

Given $x \in \Omega$, we define the *ε-normal cone* $N(x, \varepsilon)$ to be the cone generated by the set $\{\ a_i \ \mid \ i \in I(x, \varepsilon) \ \} \cup \{\mathbf{0}\}$. The presence of $\{\mathbf{0}\}$ means that $N(x, \varepsilon) = \{\mathbf{0}\}$ if $I(x, \varepsilon) = \emptyset$. The corresponding polar cone is the *ε-tangent cone* $T(x, \varepsilon)$. Observe that if $\varepsilon = 0$, then these are just the standard normal and tangent cones; that is, $N(x, 0) = N(x)$ and $T(x, 0) = T(x)$.

Examples of ε-normal and ε-tangent cones are illustrated in Figure 2.1. The set $x + T(x, \varepsilon)$ approximates the feasible region near x, where "near" is with respect to the value of ε. Note that if $I(x, \varepsilon) = \emptyset$, so that $N(x, \varepsilon) = \{\mathbf{0}\}$, then $T(x, \varepsilon) = \mathbb{R}^n$; in other words, if the boundary is more than distance ε away, then the problem looks

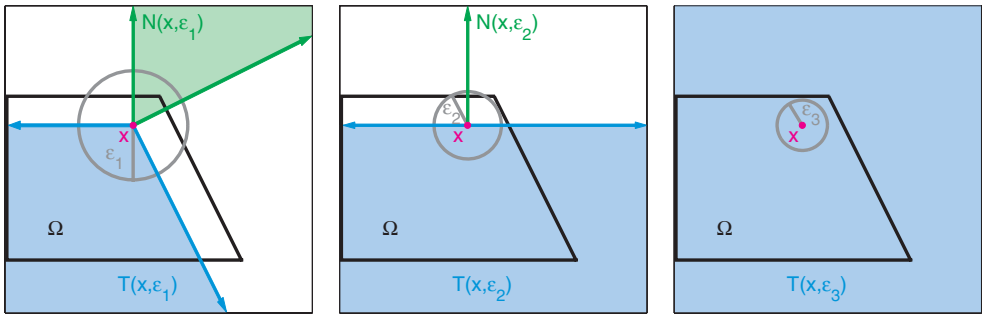


FIG. 2.1. *The cones $N(x,\varepsilon)$ and $T(x,\varepsilon)$ for the values ε_1, ε_2, and ε_3. Note that for this example, as ε varies from ε_1 to 0, there are only the three distinct pairs of cones illustrated ($N(x,\varepsilon_3) = \{\mathbf{0}\}$).*

unconstrained in the ε-neighborhood of x, as can be seen in the third example in Figure 2.1. Observe that one can proceed from x along any direction in $T(x,\varepsilon)$ for a distance of at least ε, and remain inside the feasible region; this is formalized in Proposition 2.2. Overall, the number of *distinct* ε-normal cones (and consequently the number of distinct ε-polar cones) is finite; see Proposition 2.3.

PROPOSITION 2.2. *If $x \in \Omega$, and $v \in T(x,\varepsilon)$ satisfies $\|v\| \leq \varepsilon$, then $x + v \in \Omega$.*

Proof. Let $x \in \Omega$, and $v \in T(x,\varepsilon)$ with $\|v\| \leq \varepsilon$. Since $v \in T(x,\varepsilon) = (N(x,\varepsilon))^\circ$, $a_i^T v \leq 0$ for all $i \in I(x,\varepsilon)$. Thus, $x+v$ satisfies all constraints with $i \in I(x,\varepsilon)$ because

$$a_i^T(x+v) = a_i^T x + a_i^T v \leq b + 0 = b.$$

Meanwhile, if $i \notin I(x,\varepsilon)$, the face $\mathcal{C}_i$ where the ith constraint is binding is more than distance ε away from x. Thus, $x + v \in \Omega$. ∎

PROPOSITION 2.3. *For all $x \in \Omega$ and $\varepsilon > 0$, there are at most 2^m distinct sets $I(x,\varepsilon)$. Consequently, there are at most 2^m distinct cones $N(x,\varepsilon)$ and at most 2^m distinct cones $T(x,\varepsilon)$.*

Proof. Each $I(x_k,\varepsilon_k)$ is a subset of $\{1,\ldots,m\}$, of which there are exactly 2^m possible subsets, including the empty set. The remainder of the proof follows directly from the definitions of $N(x,\varepsilon)$ and $T(x,\varepsilon)$. ∎

2.3. Conditions on the search directions. We now state the conditions on the sets of search directions for GSS for linearly constrained optimization.

At each iteration, a linearly constrained GSS method assembles $\mathcal{D}_k$, the set of search directions. We partition $\mathcal{D}_k$ into two subsets that play different roles in the analysis:

$$\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k.$$

The set $\mathcal{G}_k$ is required to generate $T(x_k,\varepsilon_k)$ and is called the set of *core directions*. The requirement that the set of search directions contain a set of generators for $T(x_k,\varepsilon_k)$ (which is always $\mathbb{R}^n$ in the unconstrained case) is what led to the name *generating set search* [15].

The (possibly empty) set $\mathcal{H}_k$ accommodates any remaining directions in $\mathcal{D}_k$, the presence of which may prove instrumental in efforts to accelerate the overall progress of the search. For instance, using $\mathcal{H}_k = \{a_i : i \in I(x_k,\varepsilon_k)\}$ can be advantageous computationally [17].

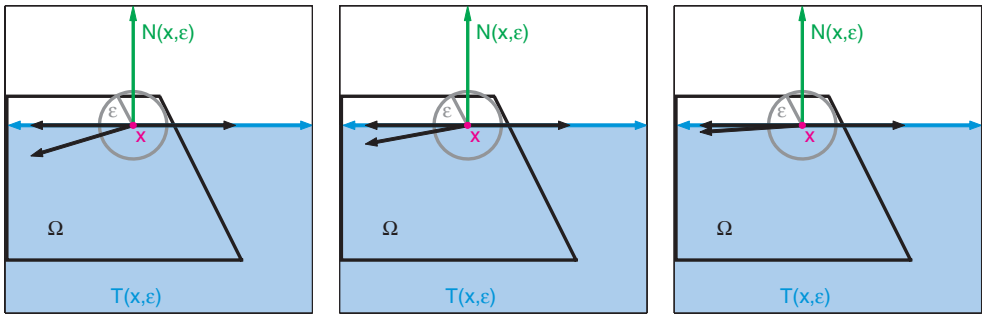


FIG. 2.2. *Condition* 1 *is needed to avoid a sequence of* $\mathcal{G}_k$*'s for which* $\kappa(\mathcal{G}_k) \to 0$.

Our focus here is on the conditions on $\mathcal{G}_k$. The set $\mathcal{H}_k$ accommodates additional directions suggested by heuristics to improve the progress of the search, but has little effect on the analysis. The generating set for $T(x_k, \varepsilon_k)$ contained in $\mathcal{G}_k$ is crucial.

CONDITION 1. There exists a constant $\kappa_{\min} > 0$, independent of k, such that for every k for which $T(x_k, \varepsilon_k) \neq \{\mathbf{0}\}$, the set $\mathcal{G}_k$ generates $T(x_k, \varepsilon_k)$ and satisfies $\kappa(\mathcal{G}_k) \geq \kappa_{\min}$.

Even though there are only finitely many ε-tangent cones $T(x, \varepsilon)$, the set of possible generators for each cone is not necessarily unique, as seen in Figure 2.2. The lower bound $\kappa_{\min}$ from Condition 1 precludes a sequence of $\mathcal{G}_k$'s for which $\kappa(\mathcal{G}_k) \to 0$. Such a situation is depicted in Figure 2.2 for

$$\mathcal{G} = \left\{ \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -\eta \end{pmatrix} \right\}$$

with three choices of $\eta > 0$. If $-\nabla f(x) = (0, \ -1)^T$, neither of the first two elements of $\mathcal{G}$ are descent directions. Furthermore, since

$$\kappa(\mathcal{G}) \leq \max_{d \in \mathcal{G}} \frac{-\nabla f(x)^T d}{\| \nabla f(x) \| \, \| d \|} = \frac{\eta}{\sqrt{1+\eta^2}} < \eta,$$

the remaining element in $\mathcal{G}$ will be an increasingly poor descent direction if $\eta \to 0$. A nonzero lower bound on $\kappa(\mathcal{G})$, as in Condition 1, will keep the angle between v and at least one generator bounded away from 90°; see [15, sections 2.2 and 3.4.1] for further discussion.

A simple technique to ensure Condition 1 is satisfied is as follows. Let $k_2 > k_1$. If $I(x_{k_2}, \varepsilon_{k_2}) = I(x_{k_1}, \varepsilon_{k_1})$, use the same generators for $T(x_{k_2}, \varepsilon_{k_2})$ as were used for $T(x_{k_1}, \varepsilon_{k_1})$. Recall from Proposition 2.3 that there at most 2^m distinct index sets $I(x, \varepsilon)$ and their corresponding ε-tangent cones $T(x, \varepsilon)$. It then follows that there are at most 2^m distinct sets $\mathcal{G}$ if the same set of generators is always used to generate a particular ε-tangent cone. Since by Proposition 2.1 each $\mathcal{G} \neq \{\mathbf{0}\}$ has a strictly positive value for $\kappa(\mathcal{G})$, and since this technique ensures there are only finitely many $\mathcal{G}_k$'s, we can set $\kappa_{\min} = \min\{\kappa(\mathcal{G}_k) : T(x_k, \varepsilon_k) \neq \{\mathbf{0}\}\}$. Thus, Condition 1 is satisfied.

We have not yet indicated how to compute the generators for a given $T(x_k, \varepsilon_k)$ so as to assemble $\mathcal{G}_k$. If the working set $\{ a_i \mid i \in I(x_k, \varepsilon_k) \}$ is linearly independent, then it is straightforward to calculate the generators of $T(x_k, \varepsilon_k)$ as described in

[25, 19, 17]. If the set $\{ a_i \mid i \in I(x_k, \varepsilon_k) \}$ is linearly dependent (e.g., in the degenerate case), then it is also possible to calculate the generators as described in [17]. In the latter case, the experience reported in [17] suggests that the worst-case computational complexity bounds do not indicate expected performance. For example, for one problem illustrated in [17], the worst-case estimate indicates that more than 4×10^{17} vectors need to be considered when, in fact, only one vector was needed and this one vector was easily identified in less than one-seventh of a second on a conventional workstation using Fukuda's `cddlib` package [9].

Finally, all the core directions must be uniformly bounded; see Condition 2.

CONDITION 2. There exist $\beta_{\max} \geq \beta_{\min} > 0$, independent of k, such that for every k for which $T(x_k, \varepsilon_k) \neq \{\mathbf{0}\}$, the following holds:

$$\beta_{\min} \leq \|d\| \leq \beta_{\max} \qquad \text{for all } d \in \mathcal{G}_k.$$

Condition 2 is easy to satisfy, say, by normalizing all search directions so that $\beta_{\min} = \beta_{\max} = 1$. However, there may be situations where it makes sense to allow the directions in $\mathcal{G}_k$ to accommodate scaling information. This poses no difficulties for the analysis, so long as there are lower and upper bounds, independent of k, on the norm of each $d \in \mathcal{G}_k$.

3. Choosing the step lengths. Given a set of search directions, the length of the step along each direction is dictated by the step-length control parameter Δ_k. In the unconstrained case, the set of trial points at iteration k would be

$$\left\{ x_k + \Delta_k d_k^{(i)} \mid i = 1, \ldots, p_k \right\},$$

where

$$\mathcal{D}_k = \left\{ d_k^{(1)}, d_k^{(2)}, \ldots, d_k^{(p_k)} \right\}.$$

In the constrained case, however, some of those trial points may be infeasible. Thus, the trial points are instead defined by

$$\left\{ x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \mid i = 1, \ldots, p_k \right\},$$

where

$$\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$$

is chosen so that $x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \in \Omega$. The main requirement on choosing $\tilde{\Delta}_k^{(i)}$ is that a full step is used if possible, as formally stated in the following condition.

CONDITION 3. If $x_k + \Delta_k d_k^{(i)} \in \Omega$, then $\tilde{\Delta}_k^{(i)} = \Delta_k$.

The simplest formula for choosing $\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$ that satisfies Condition 3 is

$$\tilde{\Delta}_k^{(i)} = \begin{cases} \Delta_k & \text{if } x_k + \Delta_k d_k^{(i)} \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

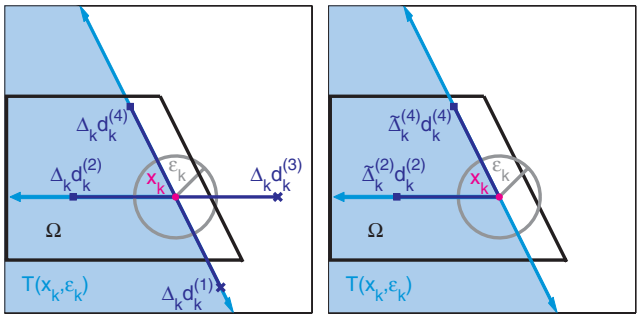


FIG. 3.1. *The step-length control parameter Δ_k may lead to infeasible trial points. The effect of using* (3.1) *is that infeasible points simply are not considered as candidates to replace x_k.*

This corresponds to a form of exact penalization (see [15, section 8.1]) since the effect of (3.1) is to reject (by setting $\tilde{\Delta}_k^{(i)} = 0$) any step $\Delta_k d_k^{(i)}$ that would generate an infeasible trial point. Since the constraints are assumed to be explicit (i.e., A and b are known), verifying the feasibility of a trial point is straightforward. This strategy is illustrated in Figure 3.1.

More sophisticated strategies can be employed for choosing $\tilde{\Delta}_k^{(i)}$ when $x_k + \Delta_k d_k^{(i)}$ is infeasible. Since alternatives for choosing $\tilde{\Delta}_k^{(i)}$ depend on the globalization strategy, we defer the discussion of further examples to section 4.

4. Globalization. Globalization of GSS refers to the conditions that are enforced to ensure that

$$\liminf_{k\to\infty} \Delta_k = 0. \tag{4.1}$$

These conditions affect the decision of whether or not to accept a trial point as the next iterate and how to update Δ_k. Globalization strategies for GSS are discussed in detail in [15, section 3.7]. Here we review those features that are relevant to our analysis of algorithms for the linearly constrained case.

In any GSS algorithm, x_k is always the best feasible point discovered thus far; i.e., $f(x_k) \le f(x_j)$ for all $j \le k$. However, different conditions are imposed on *how much better* a trial point must be to be accepted as the next iterate.

In general, for an iteration to be considered successful we require that

$$\begin{aligned} &x_k + \tilde{\Delta}_k d_k \in \Omega \quad \text{and} \quad f(x_k + \tilde{\Delta}_k d_k) < f(x_k) - \rho(\Delta_k) \\ &\text{for some } d_k \in \mathcal{D}_k \quad \text{and} \quad \tilde{\Delta}_k \in [0, \Delta_k]. \end{aligned} \tag{4.2}$$

The function $\rho(\cdot)$ is called the *forcing function* and must satisfy Condition 4.

CONDITION 4 (general requirements on the forcing function).
1. The function $\rho(\cdot)$ is a nonnegative continuous function on $[0, +\infty)$.
2. The function $\rho(\cdot)$ is $o(t)$ as $t \downarrow 0$; i.e., $\lim_{t\downarrow 0} \rho(t)/t = 0$.
3. The function $\rho(\cdot)$ is nondecreasing; i.e., $\rho(t_1) \le \rho(t_2)$ if $t_1 \le t_2$.

Both $\rho(\Delta) \equiv 0$ and $\rho(\Delta) = \alpha\Delta^p$, where $\alpha > 0$ and $p > 1$, satisfy Condition 4. The first choice also requires globalization via a rational lattice, which is discussed in

section 4.2. The second choice can be used with globalization via a sufficient decrease condition, which is discussed in section 4.1.

In the case of a successful iteration (i.e., one that satisfies (4.2)), the next iterate is defined by

$$x_{k+1} = x_k + \tilde{\Delta}_k d_k \quad \text{for } k \in \mathcal{S}.$$

(Recall from section 1.1 that the set of indices of all successful iterations is denoted by $\mathcal{S}$.) In addition, Δ_k is updated according to

$$\Delta_{k+1} = \phi_k \Delta_k, \ \phi_k \geq 1 \quad \text{for } k \in \mathcal{S}.$$

The parameter ϕ_k is called the *expansion parameter*.

For the kth iteration to be unsuccessful, it must be the case that

$$x_k + \Delta_k d \notin \Omega \ \text{ or } \ f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k) \text{ for every } d \in \mathcal{G}_k. \tag{4.3}$$

When the iteration is unsuccessful, the best point is unchanged:

$$x_{k+1} = x_k \quad \text{for } k \in \mathcal{U}.$$

(Recall from section 1.1 that the set of indices of all unsuccessful iterations is denoted by $\mathcal{U}$.) In addition, the step-length control parameter is reduced:

$$\Delta_{k+1} = \theta_k \Delta_k, \ \theta_k \in (0, 1) \quad \text{for } k \in \mathcal{U}.$$

The parameter θ_k is called the *contraction parameter*.

There are intimate connections between choosing the ϕ_k or θ_k in the update for Δ_k and guaranteeing that (4.1) holds. Further requirements depend on the particular choice of globalization strategy, and so are given in sections 4.1 and 4.2.

4.1. Globalization via a sufficient decrease condition. In the context of gradient-based nonlinear programming algorithms, the enforcement of a sufficient decrease condition on the step is well established (e.g., [10, 28, 29], or see the discussion in [15, section 2.2]). In the context of gradient-based methods, enforcing a sufficient decrease condition ties the choice of the step-length control parameter to the expected decrease, as estimated by the initial rate of decrease $-\nabla f(x_k)^T d_k$. In the context of GSS methods, the underlying assumption is that the value of $\nabla f(x_k)$ is unavailable—which means that the types of sufficient decrease conditions often used with gradient-based methods cannot be enforced. However, in [11] an alternative that uses the step-length control parameter, rather than $\nabla f(x_k)$, was introduced and analyzed in the context of linesearch methods for unconstrained minimization. In [21, 22, 23, 24], this basic concept was then extended to both unconstrained and constrained versions of what we here refer to as GSS methods. We now review the essential features of this approach.

Within the context of GSS methods for linearly constrained optimization, a sufficient decrease globalization strategy requires the following of the forcing function $\rho(\cdot)$ and the choice of the contraction parameter θ_k.

> CONDITION 5 (the forcing function for sufficient decrease).
> The forcing function $\rho(\cdot)$ is such that $\rho(t) > 0$ for $t > 0$.

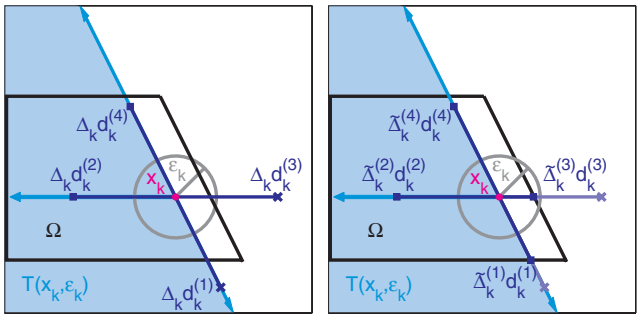


FIG. 4.1. *Observe in the second illustration that globalization via a sufficient decrease condition makes it possible to avoid infeasible trial points by simply stopping at the boundary of* Ω.

CONDITION 6 (contracting Δ_k for sufficient decrease).
A constant $\theta_{\max} < 1$ exists such that $\theta_k \leq \theta_{\max}$ for all k.

Full details are discussed in [15, section 3.7.1], but we include a few salient observations here. The requirements of Condition 5 are easily satisfied by choosing, say, $\rho(\Delta) = 10^{-4}\Delta^2$, while the requirements of Condition 6 are easily satisfied by choosing, say, $\theta_k = \frac{1}{2}$ for all k. The upper bound on the contraction factor θ_k ensures a predictable fraction of reduction on Δ_k at the conclusion of an unsuccessful iteration.

If a sufficient decrease condition is being employed, then we can use an alternative to the exact penalization strategy, given in (3.1), for choosing $\tilde{\Delta}_k^{(i)}$ when $x_k + \Delta_k d_k^{(i)} \notin \Omega$: simply find the step to the nearest constraint from x_k along $d_k^{(i)}$. This is a well-known technique in nonlinear programming (see, for instance, [10, section 5.2] or [28, section 15.4]). In other words, compute $\tilde{\Delta}_k^{(i)}$ as the maximum nonnegative feasible step along $d_k^{(i)}$. This option is illustrated in Figure 4.1.

4.2. Globalization via a rational lattice. Traditionally, direct search methods have relied on simple, as opposed to sufficient, decrease when accepting a step [33]. In other words, it is enough for the step $\tilde{\Delta}_k^{(i)} d_k^{(i)}$ to satisfy $f(x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)}) < f(x_k)$. The trade-off is that when the condition for accepting a step is relaxed to admit simple decrease, further restrictions are required on the types of steps that are allowed. These restrictions are detailed in Conditions 7, 8, and 9.

CONDITION 7 (choosing the directions for the rational lattice).
Let $\mathbf{G} = \cup_{k=0}^{\infty} \mathcal{G}_k$.
1. The set $\mathbf{G}$ is finite and so can be written as $\mathbf{G} = \{g^{(1)}, \ldots, g^{(p)}\}$.
2. Every vector $g \in \mathbf{G}$ is of the form $g \in \mathbb{Z}^n$, where $\mathbb{Z}$ is the set of integers.
3. Every vector $h \in \mathcal{H}_k$ is of the form $h \in \mathbb{Z}^n$.

CONDITION 8 (expanding or contracting Δ_k for the rational lattice).
1. The scalar τ is a fixed rational number strictly greater than 1.
2. For all $k \in \mathcal{S}$, ϕ_k is of the form $\phi_k = \tau^{\ell_k}$, where $\ell_k \in \{0, \ldots, L\}$, $L \geq 0$.
3. For all $k \in \mathcal{U}$, θ_k is of the form $\theta_k = \tau^{m_k}$, where $m_k \in \{M, \ldots, -1\}$, $M \leq -1$.

CONDITION 9 (choosing the steps for the rational lattice).
$\tilde{\Delta}_k^{(i)}$ satisfies either $\tilde{\Delta}_k^{(i)} = 0$ or $\tilde{\Delta}_k^{(i)} = \tau^{\tilde{m}_k^{(i)}} \Delta_k$, where $\tilde{m}_k^{(i)} \in \{\tilde{M}, \ldots, 0\}$, $\tilde{M} \leq 0$.

While the list of requirements in Conditions 7, 8, and 9 looks onerous, they can be satisfied in a straightforward fashion. A discussion of the reasons for these conditions can be found in [19, sections 3.4, 4, and 5]. (A detailed discussion of the rational lattice globalization strategy for the unconstrained case can be found in [15, section 3.7.2].) Here we make only a few pertinent observations.

First, a critical consequence of Conditions 7 and 8 is that when these two conditions are enforced, along with the exact penalization strategy in (3.1), Theorem 5.1 in [19] ensures that all iterates lie on a rational lattice. This fact plays a crucial role in guaranteeing (4.1) when only simple decrease is enforced. Condition 9 is a straightforward extension that preserves the fact that all the iterates lie on a rational lattice while relaxing the exact penalization strategy in (3.1) (an example is shown in Figure 4.2).

Obtaining a finite $\mathbf{G}$ to satisfy part 1 of Condition 7 can be done by following the procedure outlined in section 2.3 (i.e., if $I(x_{k_2}, \varepsilon_{k_2}) = I(x_{k_1}, \varepsilon_{k_1})$ for $k_2 > k_1$, then use the same generators for $T(x_{k_2}, \varepsilon_{k_2})$ as were used for $T(x_{k_1}, \varepsilon_{k_1})$). To satisfy part 2, a standard assumption in the context of simple decrease is that the linear constraints are rational, i.e., $A \in \mathbb{Q}^{m \times n}$, where $\mathbb{Q}$ denotes the set of rational numbers. By clearing denominators, it is then possible—with some care—to obtain a set of integral vectors to generate all possible ε-tangent cones; see [19, section 8] for further discussion. Part 3 is enforced directly.

In Condition 8, the usual choice of τ is 2. The parameter ϕ_k typically is chosen to be 1 so that $\ell_k = 0$ for all k, satisfying the requirement placed on ϕ_k in Condition 8. Usually θ_k is chosen to be $\frac{1}{2}$ so that $m_k = -1$ for all k, satisfying the requirement placed on θ_k in Condition 8. The fact that τ^{-1} is the largest possible choice of θ_k obviates the need to explicitly bound θ_k from above, as was required in Condition 6 for sufficient decrease.

Condition 9 says that it is possible to choose a partial step along a given direction so long as the trial point remains on a rational lattice. One strategy is illustrated in Figure 4.2. Starting with the situation illustrated on the left, along direction $d^{(1)}$, $\tilde{\Delta}_k^{(1)} = 0.5\Delta_k$ yields the feasible trial step $\tilde{\Delta}_k^{(1)} d^{(1)}$ while along direction $d^{(3)}$, $\tilde{\Delta}_k^{(3)} = 0.25\Delta_k$ yields the feasible trial step $\tilde{\Delta}_k^{(3)} d^{(3)}$, as illustrated on the right. These choices for $\tilde{\Delta}_k^{(1)}$ and $\tilde{\Delta}_k^{(3)}$ correspond to choosing $m_k^{(1)} = -1$ and $m_k^{(3)} = -2$, with $\tau = 2$ and $\tilde{M} = -2$.

The general strategy is to find the largest $\tilde{\Delta}_k^{(i)}$ (by finding the largest $m_k^{(i)}$) such that $x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \in \Omega$ while satisfying Condition 9. To do so, either reduce Δ_k by a

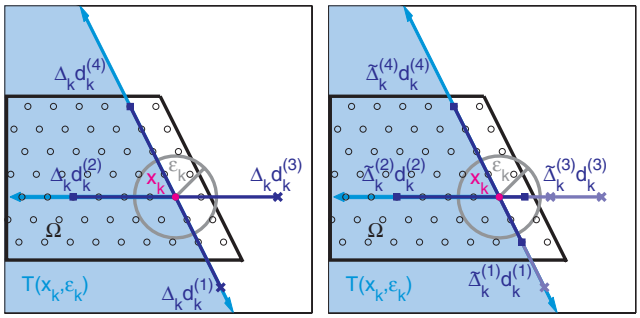


FIG. 4.2. *Globalization via a rational lattice means that the trial points lie on the rational lattice that exists as a consequence of Conditions 7–9. For this example note that while the two reduced steps are near the boundary, the requirement that they remain on the rational lattice means that they may not be on the boundary.*

factor of $1/\tau$ until

$$x_k + \tau^{m_k^{(i)}} \Delta_k d_k^{(i)} \in \Omega \quad \text{with } m_k^{(i)} \geq \tilde{M} \tag{4.4}$$

or set $\tilde{\Delta}_k^{(i)} = 0$ if it is not possible to satisfy (4.4) (for instance, when x_k is on the boundary of the feasible region then *any* step along $d_k^{(i)}$ would be infeasible).

5. GSS algorithms for linearly constrained problems. We now formally state two GSS algorithms for solving linearly constrained optimization problems. The fundamental requirement for both algorithms is that at every iteration k, the set of search directions $\mathcal{D}_k$ must include a set of generators $\mathcal{G}_k$ for the ε-normal cone $T(x_k, \varepsilon_k)$—hence the name *generating set search* methods. The primary requirements on the GSS methods presented here are that they satisfy Conditions 1, 2, 3, and 4. The differences in the two versions given depend on the type of globalization that is used: sufficient decrease in Algorithm 5.1 versus simple decrease in Algorithm 5.2. Sufficient decrease requires Conditions 5 and 6. Simple decrease admits the choice $\rho(\cdot) \equiv 0$, but requires Conditions 7, 8, and 9 in lieu of Conditions 5 and 6.

New in the statements of Algorithms 5.1 and 5.2, and to the analysis that follows, is the way in which ε_k is defined, which has bearing on the construction of the critical set $\mathcal{G}_k \subseteq \mathcal{D}_k$. Here we set $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$. This selection of ε_k differs from that used in either [19] or [24]. Specifically, in [19] and Algorithm 2 of [24]—as well as earlier in [25], in a slightly restricted form—$\mathcal{G}_k$ is required to contain generators for $T(x_k, \varepsilon)$ for all ε in the interval $[0, \varepsilon_{\max}]$, with $\varepsilon_{\max} > 0$. This means that $\mathcal{G}_k$ may need to contain generators for multiple cones rather than a single cone. Since Δ_k can be reduced only at the conclusion of an unsuccessful iteration, and an unsuccessful iteration requires the verification of (4.3), there is practical incentive to try and keep the cardinality of $\mathcal{G}_k$ manageable when the cost of computing $f(x)$ for $x \in \Omega$ is appreciable. Thus, Algorithm 1 in [24] first introduced the potential for a smaller set of search directions: the set of search directions must exactly generate $T(x_k, \varepsilon_k)$—and *only* $T(x_k, \varepsilon_k)$. Using our notation, this means that $\mathcal{H}_k = \emptyset$ for all k. Furthermore, for Algorithm 1 in [24], ε_k is simply a parameter decreased at unsuccessful iterations as opposed to the particular choice of ε_k given here.

Our requirement that the search directions include generators for $T(x_k, \varepsilon_k)$, with $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$, is a compromise. On the one hand, it may significantly

ALGORITHM 5.1. LINEARLY CONSTRAINED GSS USING A SUFFICIENT DECREASE GLOBALIZATION STRATEGY

INITIALIZATION.

Let $x_0 \in \Omega$ be the initial guess.

Let $\Delta_{\text{tol}} > 0$ be the tolerance used to test for convergence.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\varepsilon_{\max} > \beta_{\max}\Delta_{\text{tol}}$ be the maximum distance used to identify nearby constraints ($\varepsilon_{\max} = +\infty$ is permissible).

Let $\rho(\cdot)$ be a forcing function satisfying Conditions 4 and 5.

ALGORITHM. For each iteration $k = 0, 1, 2, \ldots$

STEP 1. Let $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$. Choose a set of search directions $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$ satisfying Conditions 1 and 2.

STEP 2. If there exists $d_k \in \mathcal{D}_k$ and a corresponding $\tilde{\Delta}_k \in [0, \Delta_k]$ satisfying Condition 3 such that $x_k + \tilde{\Delta}_k d_k \in \Omega$ and

$$f(x_k + \tilde{\Delta}_k d_k) < f(x_k) - \rho(\Delta_k),$$

then:

- Set $x_{k+1} = x_k + \tilde{\Delta}_k d_k$.
- Set $\Delta_{k+1} = \phi_k \Delta_k$ for any choice of $\phi_k \geq 1$.

STEP 3. Otherwise, for every $d \in \mathcal{G}_k$, either $x_k + \Delta_k d \notin \Omega$ or

$$f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k).$$

In this case:

- Set $x_{k+1} = x_k$ (no change).
- Set $\Delta_{k+1} = \theta_k \Delta_k$ for some choice $\theta_k \in (0, 1)$ satisfying Condition 6.

If $\Delta_{k+1} < \Delta_{\text{tol}}$, then terminate.

FIG. 5.1. *Linearly constrained GSS using a sufficient decrease globalization strategy.*

decrease the number of directions in $\mathcal{G}_k$ over that needed when $\mathcal{G}_k$ is required to contain generators for $T(x_k, \varepsilon)$ for all ε in the interval $[0, \varepsilon_{\max}]$. On the other hand, it allows $\mathcal{H}_k \neq \emptyset$—the set of search directions can be augmented in an effort to accelerate the search—without adversely affecting the convergence guarantees for the algorithm.

Yoking the value of ε_k to the value of Δ_k has geometrical motivations. Once Δ_k is small enough, so that $\varepsilon_k = \beta_{\max}\Delta_k$, full steps along directions in $\mathcal{G}_k$ will be feasible, as Figure 2.1 demonstrates.

There is an intuitive practical appeal to allowing—while not requiring—$\mathcal{D}_k$ to

ALGORITHM 5.2. LINEARLY CONSTRAINED GSS USING A RATIONAL LATTICE GLOBALIZATION STRATEGY

INITIALIZATION.

Let $x_0 \in \Omega$ be the initial guess.

Let $\Delta_{\text{tol}} > 0$ be the tolerance used to test for convergence.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\varepsilon_{\max} > \beta_{\max}\Delta_{\text{tol}}$ be the maximum distance used to identify nearby constraints ($\varepsilon_{\max} = +\infty$ is permissible).

Let $\rho(\cdot)$ be a forcing function satisfying Condition 4, e.g., $\rho(\cdot) \equiv 0$ is typical.

ALGORITHM. For each iteration $k = 0, 1, 2, \ldots$

STEP 1. Let $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$. Choose a set of search directions $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$ satisfying Conditions 1, 2, and 7.

STEP 2. If there exists $d_k \in \mathcal{D}_k$ and a corresponding $\tilde{\Delta}_k \in [0, \Delta_k]$ satisfying Conditions 3 and 9 such that $x_k + \tilde{\Delta}_k d_k \in \Omega$ and

$$f(x_k + \tilde{\Delta}_k d_k) < f(x_k) - \rho(\Delta_k),$$

then:

- Set $x_{k+1} = x_k + \tilde{\Delta}_k d_k$.
- Set $\Delta_{k+1} = \phi_k \Delta_k$ for a choice of $\phi_k \geq 1$ satisfying Condition 8.

STEP 3. Otherwise, for every $d \in \mathcal{G}_k$, either $x_k + \Delta_k d \notin \Omega$ or

$$f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k).$$

In this case:

- Set $x_{k+1} = x_k$ (no change).
- Set $\Delta_{k+1} = \theta_k \Delta_k$ for some choice $\theta_k \in (0, 1)$ satisfying Condition 8.

If $\Delta_{k+1} < \Delta_{\text{tol}}$, then terminate.

FIG. 5.2. *Linearly constrained GSS using a rational lattice globalization strategy.*

include more search directions. Note that if $T(x_k, \varepsilon_k) \neq \{\mathbf{0}\}$, then the directions in $\mathcal{G}_k$ will move the search along directions that are in some sense "parallel" (the situation is more complicated for $n > 2$) to the faces of the polyhedron that have been identified by the working set. This is best seen in the illustration on the left in Figure 2.1. Intuitively, it makes sense to also allow the search to move *toward* the faces of the polyhedron that have been identified by the working set—particularly

when the solution lies on the boundary of the feasible region. Such intuition is borne out by the numerical results reported in [17].

Before proceeding, we note a technical difference between the presentation of the algorithms in Algorithms 5.1 and 5.2 and what is assumed for the analysis in section 6. In practice, GSS algorithms terminate when the step-length control parameter Δ_k falls below a given threshold $\Delta_{\text{tol}} > 0$. Because this is important to any implementation, we have included it in the statement of the algorithm. In Theorems 6.3, 6.4, and 6.5, however, we assume that the iterations continue ad infinitum (i.e., in the context of the analysis, the reader should assume $\Delta_{\text{tol}} = 0$).

5.1. GSS using a sufficient decrease condition. A linearly constrained GSS algorithm based on a sufficient decrease globalization strategy is presented in Algorithm 5.1. Using a sufficient decrease globalization strategy, as outlined in section 4.1, requires that we enforce two particular conditions. Condition 5 ensures that $\rho(\Delta_k) = 0$ only when $\Delta_k = 0$. Condition 6 ensures that there is sufficient reduction on Δ_k at unsuccessful iterations.

The only assumption on f necessary to show that some subsequence of $\{\Delta_k\}$ converges to zero is that f be bounded below in the feasible region.

THEOREM 5.1 (see Theorem 3.4 of [15]). *Suppose f is bounded below on Ω. Then for a linearly constrained GSS method using a sufficient decrease globalization strategy satisfying Conditions* 4, 5, *and* 6 (*as outlined in Algorithm* 5.1), $\liminf_{k\to\infty} \Delta_k = 0$.

5.2. GSS using a rational lattice. A linearly constrained GSS algorithm based on a rational lattice globalization strategy is presented in Algorithm 5.2. The choice $\rho(\cdot) \equiv 0$ is standard for the rational lattice globalization strategy, which means only simple decrease, i.e., $f(x_k + \tilde{\Delta}_k d_k) < f(x_k)$, is required. We note, however, that a sufficient decrease condition may be employed in conjunction with a rational lattice globalization strategy; see [15, section 3.7.2]. The choice $\rho(\cdot) \equiv 0$ also means that Condition 4 is satisfied automatically. The trade-off for using simple decrease is that additional conditions must be imposed on the choice of admissible $\mathcal{D}_k$ (Condition 7), ϕ_k and θ_k (Condition 8), and $\tilde{\Delta}_k$ (Condition 9).

Using a rational lattice globalization strategy, to show that some subsequence of the step-length control parameters goes to zero, the only assumption placed on f is that the set $\mathcal{F} = \{\, x \in \Omega \mid f(x) \le f(x_0) \,\}$ be bounded. This is a stronger condition on f than is needed when using a sufficient decrease globalization strategy, where all that is required is that f be bounded below. The analysis for the rational lattice globalization strategy requires the sequence $\{x_k\}$ to remain in a bounded set so as to ensure that there is a finite number of lattice points to consider. We could adopt this weaker assumption, though it is not clear how it would be enforced in practice. Instead, assuming that $\mathcal{F}$ is bounded guarantees this requirement.

THEOREM 5.2 (see Theorem 6.5 of [19]). *Assume that $\mathcal{F} = \{x \in \Omega \mid f(x) \le f(x_0)\}$ is bounded and that $A \in \mathbb{Q}^{m\times n}$, where $\mathbb{Q}$ denotes the set of rational numbers. Then for a linearly constrained GSS method using a rational lattice globalization strategy satisfying Conditions* 4, 7, 8, *and* 9 (*as outlined in Algorithm* 5.2), $\liminf_{k\to\infty} \Delta_k = 0$.

6. Stationarity results. At *unsuccessful* iterations of the linearly constrained GSS methods outlined in Algorithms 5.1 and 5.2, we can bound the measure of stationarity $\chi(x_k)$ in terms of Δ_k. To do so, we make the following assumptions.

ASSUMPTION 6.1. *The set* $\mathcal{F} = \{ x \in \Omega \mid f(x) \leq f(x_0) \}$ *is bounded.*

ASSUMPTION 6.2. *The gradient of* f *is Lipschitz continuous with constant* M *on* Ω.

If Assumptions 6.1 and 6.2 hold, then there exists $\gamma > 0$ such that for all $x \in \mathcal{F}$,

$$\| \nabla f(x) \| < \gamma. \tag{6.1}$$

We then have the following results for the algorithms in Algorithms 5.1 and 5.2. Recall from section 2.1 that given a convex cone K and any vector v, we denote the projection of v onto K by v_K.

THEOREM 6.3. *Suppose that Assumption* 6.2 *holds. Consider the linearly constrained GSS algorithms given in Algorithms* 5.1 *and* 5.2*, both of which satisfy Conditions* 1*,* 2*, and* 3*. If* $k \in \mathcal{U}$ *and* ε_k *satisfies* $\varepsilon_k = \beta_{\max}\Delta_k$*, then*

$$\| [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \| \leq \frac{1}{\kappa_{\min}} \left(M\Delta_k\beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k\beta_{\min}} \right). \tag{6.2}$$

Here, $\kappa_{\min}$ *is from Condition* 1*,* M *is from Assumption* 6.2*, and* $\beta_{\max}$ *and* $\beta_{\min}$ *are from Condition* 2*.*

Proof. Clearly, we need only consider the case when $[-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \neq 0$. Condition 1 guarantees a set $\mathcal{G}_k$ that generates $T(x_k, \varepsilon_k)$. By (2.1) (with $K = T(x_k, \varepsilon_k)$ and $v = -\nabla f(x_k)$) there exists some $\hat{d} \in \mathcal{G}_k$ such that

$$\kappa(\mathcal{G}_k) \, \| [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \| \, \| \hat{d} \| \leq -\nabla f(x_k)^T \hat{d}. \tag{6.3}$$

Condition 3 and the fact that iteration k is unsuccessful tell us that

$$f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k) \quad \text{for all } d \in \mathcal{G}_k \quad \text{for which } x_k + \Delta_k d \in \Omega.$$

Condition 2 ensures that for all $d \in \mathcal{G}_k$, $\| \Delta_k d \| \leq \Delta_k \beta_{\max}$ and, by assumption, $\Delta_k\beta_{\max} = \varepsilon_k$, so we have $\| \Delta_k d \| \leq \varepsilon_k$ for all $d \in \mathcal{G}_k$. Proposition 2.2 then assures us that $x_k + \Delta_k d \in \Omega$ for all $d \in \mathcal{G}_k$. Thus,

$$f(x_k + \Delta_k d) - f(x_k) + \rho(\Delta_k) \geq 0 \quad \text{for all } d \in \mathcal{G}_k. \tag{6.4}$$

Meanwhile, since the gradient of f is assumed to be continuous (Assumption 6.2), we can apply the mean value theorem to obtain, for some $\alpha_k \in (0, 1)$,

$$f(x_k + \Delta_k d) - f(x_k) = \Delta_k \nabla f(x_k + \alpha_k \Delta_k d)^T d \quad \text{for all } d \in \mathcal{G}_k.$$

Putting this together with (6.4),

$$0 \leq \Delta_k \nabla f(x_k + \alpha_k \Delta_k d)^T d + \rho(\Delta_k) \quad \text{for all } d \in \mathcal{G}_k.$$

Dividing through by Δ_k and subtracting $\nabla f(x_k)^T d$ from both sides yields

$$-\nabla f(x_k)^T d \leq \left(\nabla f(x_k + \alpha_k \Delta_k d) - \nabla f(x_k)\right)^T d + \rho(\Delta_k)/\Delta_k \quad \text{for all } d \in \mathcal{G}_k.$$

Since $\nabla f(x)$ is Lipschitz continuous (Assumption 6.2) and $0 < \alpha_k < 1$, we obtain

$$-\nabla f(x_k)^T d \le M\Delta_k \| d \|^2 + \rho(\Delta_k)/\Delta_k \quad \text{for all } d \in \mathcal{G}_k. \tag{6.5}$$

Since (6.5) holds for all $d \in \mathcal{G}_k$, (6.3) tells us that for some $\hat{d} \in \mathcal{G}_k$,

$$\kappa(\mathcal{G}_k) \, \| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| \le M\Delta_k \| \, \hat{d} \, \| + \frac{\rho(\Delta_k)}{\Delta_k \, \| \, \hat{d} \, \|}.$$

Using the bounds on $\| \, \hat{d} \, \|$ in Condition 2,

$$\| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| \le \frac{1}{\kappa(\mathcal{G}_k)} \left(M\Delta_k \beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k \beta_{\min}} \right).$$

The theorem then follows from the fact that $\kappa(\mathcal{G}_k) \ge \kappa_{\min}$ (Condition 1). ∎

Theorem 6.4 relates the measure of stationarity $\chi(x_k)$ to the step-length control parameter Δ_k. Before we proceed, we define the following constant (recall that $\kappa(\cdot)$ is defined in (2.1)):

$$\nu_{\min} = \min \left\{ \kappa(\mathcal{A}) : \mathcal{A} = \cup_{i \in I(x,\varepsilon)} \{a_i\}, \; x \in \Omega, \; \varepsilon \ge 0, \; I(x,\varepsilon) \ne \emptyset \right\} > 0. \tag{6.6}$$

We know that $\nu_{\min} > 0$ because there are no more than 2^m possibilities for $\mathcal{A}$.

THEOREM 6.4. *Suppose that Assumptions* 6.1 *and* 6.2 *hold. Consider the linearly constrained GSS algorithms given in Algorithms* 5.1 *and* 5.2*, both of which satisfy Conditions* 1*,* 2*, and* 3*. If $k \in \mathcal{U}$ and $\varepsilon_k = \beta_{\max}\Delta_k$, then*

$$\chi(x_k) \le \left(\frac{M}{\kappa_{\min}} + \frac{\gamma}{\nu_{\min}} \right) \Delta_k \, \beta_{\max} + \frac{1}{\kappa_{\min} \, \beta_{\min}} \frac{\rho(\Delta_k)}{\Delta_k}. \tag{6.7}$$

Here, $\kappa_{\min}$ is from Condition 1*, $\nu_{\min}$ is from* (6.6)*, M is from Assumption* 6.2*, γ is from* (6.1)*, and $\beta_{\max}$ and $\beta_{\min}$ are from Condition* 2*.*

Proof. Since $\varepsilon_k = \Delta_k \beta_{\max}$, Proposition B.2 tells us that

$$\chi(x_k) \le \| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| + \frac{\Delta_k \beta_{\max}}{\nu_{\min}} \, \| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \|.$$

Furthermore, the bound on $\| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \|$ from Theorem 6.3 holds. The projection onto convex sets is contractive, so $\| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \| \le \| \, \nabla f(x_k) \, \|$. Under Assumptions 6.1 and 6.2, (6.1) holds, so $\| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \| \le \gamma$. The result follows. ∎

If we choose either $\rho(\Delta) \equiv 0$ or $\rho(\Delta) = \alpha\Delta^p$ with $\alpha > 0$ and $p \ge 2$, then we obtain an estimate of the form $\chi(x_k) = O(\Delta_k)$.

The constants M, γ, and $\nu_{\min}$ in (6.7) are properties of the linearly constrained optimization problem. The remaining quantities—the bounds on the lengths of the search directions $\beta_{\min}$ and $\beta_{\max}$, as well as $\kappa_{\min}$—are under the control of the algorithm.

Before continuing, we observe that the Lipschitz assumption (Assumption 6.2) can be relaxed. A similar bound can be obtained assuming only continuous differentiability of f. Let ω denote the following modulus of continuity of $\nabla f(x)$: given $x \in \Omega$ and $r > 0$,

$$\omega(x,r) = \max \{ \| \, \nabla f(y) - \nabla f(x) \, \| \;\mid\; y \in \Omega, \; \| \, y - x \, \| \le r \}.$$

Then the proof of Theorem 6.4 yields the bound

$$\chi(x_k) \leq \frac{1}{\kappa_{\min}} \omega(x_k, \Delta_k \beta_{\max}) + \frac{\gamma}{\nu_{\min}} \Delta_k \beta_{\max} + \frac{1}{\kappa_{\min} \beta_{\min}} \frac{\rho(\Delta_k)}{\Delta_k}.$$

Returning to Theorem 6.4, if we recall from Theorems 5.1 and 5.2 that the step-length control parameter Δ_k is manipulated explicitly by GSS methods in a way that ensures $\liminf_{k\to\infty} \Delta_k = 0$, then an immediate corollary is the following first-order convergence result.

THEOREM 6.5. *Suppose that Assumptions* 6.1 *and* 6.2 *hold. Consider either*

(i) *the linearly constrained GSS algorithm in Algorithm* 5.1*, which satisfies Conditions* 1*,* 2*,* 3*,* 4*,* 5*, and* 6*, or*

(ii) *the linearly constrained GSS algorithm in Algorithm* 5.2*, which satisfies Conditions* 1*,* 2*,* 3*,* 4*,* 7*,* 8*, and* 9*, with the additional assumption that A is rational.*

For both algorithms we have $\liminf_{k\to+\infty} \chi(x_k) = 0$.

7. Using Δ_k to terminate GSS methods after unsuccessful iterations. We now present some numerical illustrations of the practical implications of Theorem 6.4. We show that Δ_k can be used as a reasonable measure of stationarity when implementing GSS methods to solve linearly constrained minimization problems. The results in section 6 serve as a justification for terminating the search when $\Delta_k < \Delta_{\text{tol}}$.

To demonstrate that Δ_k is a reasonable measure of stationarity, we show the following results from experiments using an implementation of a GSS method for solving linearly constrained optimization problems (a thorough discussion of the implementation, as well as further numerical results, can be found in [17]).

The first test problem is the following quadratic program (QP) for $n = 8$:

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{j=1}^{n} j^2 x_j^2 \\ \text{subject to} & 0 \leq x \leq 1, \\ & \sum_{j=1}^{n} x_j \geq 1, \end{array} \tag{7.1}$$

where x_j is the jth component of the vector x. The last constraint is binding at the solution. The second test problem is posed on a pyramid in $\mathbb{R}^3$:

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{j=1}^{3} [(4-j)^2 (x_j - c_j)^2 - x_j] \\ \text{subject to} & x_3 \geq 0, \\ & x_1 + x_2 + x_3 \leq 1, \\ & x_1 - x_2 + x_3 \leq 1, \\ & -x_1 + x_2 + x_3 \leq 1, \\ & -x_1 - x_2 + x_3 \leq 1, \end{array} \tag{7.2}$$

with $c = (0.01, 0.01, 0.98)^T$. Again, x_j and c_j are the jth components of the vectors x and j, respectively. The solution is at c, which is near the apex of the pyramid. The algorithm actually visits the apex, which is a degenerate vertex insofar as there are four constraints in three variables that meet there.

These two problems were solved using the implementation of Algorithm 5.1 reported in [17]. The forcing function was $\rho(\Delta) = 10^{-4}\Delta^2$. The set of search directions $\mathcal{D}_k$ contained both the set $\mathcal{G}_k$, the generators for the ε-tangent cone $T(x_k, \varepsilon_k)$, as well as the set $\mathcal{H}_k$, which contained the nonzero generators for the ε-normal cone $N(x_k, \varepsilon_k)$.

All search directions were normalized, so $\beta_{\min} = \beta_{\max} = 1$. For these choices, Theorem 6.4 says that $\chi(x_k) = O(\Delta_k)$ at unsuccessful iterations when $\Delta_k \leq \varepsilon_{\max}$.

We used $\theta_k = \frac{1}{2}$ and $\phi_k = 1$ for all k. After any unsuccessful iteration, we recorded the value of Δ_k and computed the value of $\chi(x_k)$. These values are reported in Table 7.1 for unsuccessful iterations with $\varepsilon_k = \Delta_k \beta_{\max}$.

TABLE 7.1
GSS *runs showing decrease in* Δ_k *versus the value of* $\chi(x_k)$ *at unsuccessful iterations.*

Δ_k	$\chi(x_k)$
0.100000000000	0.762038045731
0.050000000000	0.719781449029
0.025000000000	0.683858024464
0.012500000000	0.522963684221
0.006250000000	0.147769116216
0.003125000000	0.009094010555
0.001562500000	0.009042346694
0.000781250000	0.005424114678
0.000390625000	0.002291442563
0.000195312500	0.000803137090
0.000097656250	0.000616656194
0.000048828125	0.000583197890
0.000024414063	0.000134935864
0.000012207031	0.000214535279
0.000006103516	0.000122058457
0.000003051758	0.000033834262
0.000001525879	0.000014798430
0.000000762939	0.000002976275
0.000000381470	0.000003506102
0.000000190735	0.000001047463

(a) The QP in (7.1).

Δ_k	$\chi(x_k)$
0.100000000000	0.009296268053
0.050000000000	0.009296268053
0.025000000000	0.068321041838
0.012500000000	0.001889009252
0.006250000000	0.000193017831
0.003125000000	0.000193017831
0.001562500000	0.003786874320
0.000781250000	0.003080612089
0.000390625000	0.000016499610
0.000195312500	0.000016499610
0.000097656250	0.000004481178
0.000048828125	0.000004481178
0.000024414063	0.000001550420
0.000012207031	0.000007616742
0.000006103516	0.000007616742
0.000003051758	0.000001501552
0.000001525879	0.000000807763
0.000000762939	0.000000008203
0.000000381470	0.000000008203
0.000000190735	0.000000008203

(b) The QP in (7.2).

The point of the results reported in Table 7.1 is not to demand close scrutiny of each entry but rather to demonstrate the trend in the quantities measured. We clearly see the linear relationship between Δ_k and $\chi(x_k)$ that Theorem 6.4 tells us to expect. These results are consistent with findings for the unconstrained case [8] as well as with a long-standing recommendation for using Δ_k as a stopping criterion for direct search methods (see [14, 3, 32]).

One practical benefit of using Δ_k as a measure of stationarity is that it is already present in GSS algorithms; no additional computation is required.

We close with the observation that the effectiveness of Δ_k as a measure of stationarity clearly depends on the value of the constants in the bound in (6.7). For instance, if f is highly nonlinear, so that the Lipschitz constant M is large, then using Δ_k to estimate $\chi(x_k)$ might be misleading. While GSS methods cannot control M, γ, or $\nu_{\min}$, which depend on the linearly constrained optimization problem, a careful implementation of GSS methods for solving linearly constrained optimization problems can control the remaining constants in (6.7). Thus a careful implementation can ensure that Δ_k is a useful measure of stationarity except when f is highly nonlinear (i.e., M

is large with respect to $\| \nabla f \|$) or A is ill-conditioned.

8. Conclusions. The results we have presented are useful in several ways. First, we present a new prescription for how the search directions should conform to the boundary near an iterate x_k. Theorems 6.3 and 6.4 bring out many of the elements common to the approaches described in [18, 19] and [23, 24]. Although the globalization approaches that ensure $\liminf_{k\to\infty} \Delta_k = 0$ differ, the same analysis shows that for both classes of algorithms,

$$\chi(x_k) = O(\Delta_k).$$

This result does not depend on the method of globalization.

Second, the results presented here give theoretical support for terminating GSS methods for linearly constrained optimization when Δ_k falls below some tolerance. Under the assumptions of Theorem 6.4, at the subsequence of unsuccessful iterations ($k \in \mathcal{U}$) we have $\chi(x_k) = O(\Delta_k)$ as $\Delta_k \to 0$. At the same time, Theorem 6.4 also suggests that this stopping criterion may be unsuitable if the objective is highly nonlinear, making clear the need for direct search methods, like all optimization algorithms, to account for scaling.

Theorem 6.3 underlies the use of linearly constrained GSS methods in the augmented Lagrangian framework given in [5]. The latter proceeds by successive approximate minimization of an augmented Lagrangian. The stopping criterion in the subproblems involves the norm of the projection onto $T(x_k, \omega_k)$ of the negative gradient of the augmented Lagrangian, for a parameter $\omega_k \downarrow 0$. In the direct search setting the gradient is unavailable. However, Theorem 6.3 enables us to use Δ_k as an alternative measure of stationarity in the subproblems. Details appear in [16].

Appendix A. Criticality measure for first-order constrained stationarity. Here we discuss $\chi(x)$ and $\| q(x) \|$ in more detail. Because these measures are not novel, we have relegated their discussion to an appendix.

For $x \in \Omega$, progress toward a KKT point of (1.1) is measured by

$$\chi(x) \equiv \max_{\substack{x+w\in\Omega \\ \| w \| \le 1}} -\nabla f(x)^T w. \tag{A.1}$$

This measure was originally proposed in [5] and is discussed at length in section 12.1.4 of [6], where the following properties are noted:

1. $\chi(x)$ is continuous,
2. $\chi(x) \ge 0$, and
3. $\chi(x) = 0$ if and only if x is a KKT point for (1.1).

Showing that $\chi(x_k) \to 0$ as $k \to \infty$ for a subsequence of iterates k constitutes a global first-order stationarity result.

To help better understand this measure, the w's that define $\chi(x)$ in (A.1) are illustrated in Figure A.1 for several choices of $-\nabla f(x)$. Conn, Gould, and Toint [6] observe that $\chi(x)$ can be interpreted as the progress that can be made on a first-order model at x in a ball of radius unity with the constraint of preserving feasibility. They go on to observe that $\chi(x)$ is a direct generalization of $\| \nabla f(x) \|$; in fact, $\chi(x) = \| \nabla f(x) \|$ whenever $\Omega = \mathbb{R}^n$ or $x - \nabla f(x) \in \Omega$.

The work in [19, 20] used the measure $q(x)$ defined in (1.5) (this quantity appears in [6] as equation (12.1.19)), but the resulting stationarity result is unsatisfying in the case of general linear constraints. The quantity $\chi(x)$ turns out to be easier to work with than $q(x)$. The latter involves a projection onto the feasible polyhedron, and if

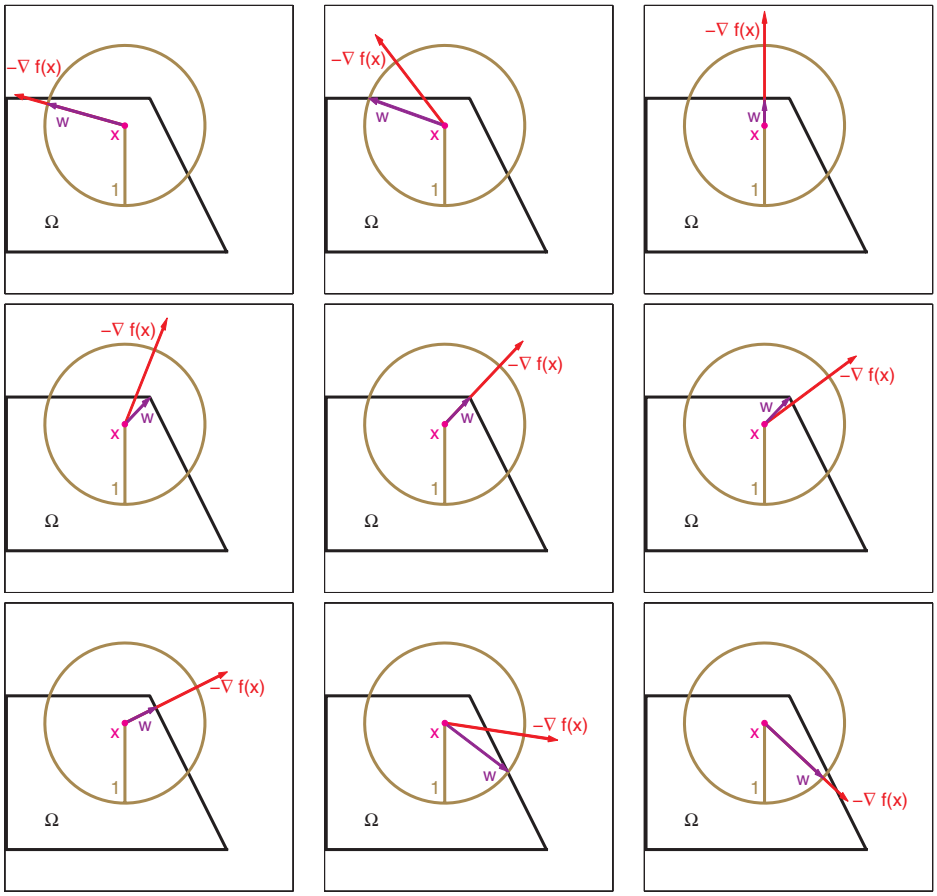


FIG. A.1. *How the* w *in* (A.1) *varies with* $-\nabla f(x)$ *when* $x - \nabla f(x) \notin \Omega$.

the constraints binding at the projection do not correspond to the constraints near x, technical difficulties ensue in relating $q(x)$ to the geometry of the feasible region near x. This is not the case with $\chi(x)$.

Appendix B. Geometric results on cones and polyhedra. Here we present geometrical results having to do with our use of $\chi(\cdot)$ as a measure of stationarity.

The first proposition says that if one can move from x to $x+v$ and remain feasible, then v cannot be too outward-pointing with respect to the constraints near x. Recall from section 2.1 that given a convex cone K and any vector v, there is a unique closest point of K to v, the *projection* of v onto K, which we denote by v_K. Thus $v_{N(x,\varepsilon)}$ is the projection of v onto the ε-normal cone $N(x,\varepsilon)$ while $v_{T(x,\varepsilon)}$ is the projection of v onto the ε-tangent cone $T(x,\varepsilon)$.

PROPOSITION B.1. *If* $x \in \Omega$ *and* $x + v \in \Omega$, *then for any* $\varepsilon \geq 0$, $\| v_{N(x,\varepsilon)} \| \leq \varepsilon/\nu_{\min}$, *where* $\nu_{\min}$ *is the constant from* (6.6).

Proof. Let $N = N(x,\varepsilon)$. The result is immediate if $v_N = 0$, so we need only consider the case when $v_N \neq 0$. Recall that N is generated by the outward-pointing normals to the binding constraints within distance ε of x; thus, the set $\mathcal{A} = \{\, a_i \mid i \in I(x,\varepsilon) \,\}$ generates N. A simple calculation shows that the distance

from x to $\{\ y\ \mid\ a_i^T y = b_i\ \}$ is $(b_i - a_i^T x)/\| a_i \|$, so it follows that

$$\frac{b_i - a_i^T x}{\| a_i \|} \leq \varepsilon \quad \text{for all } i \in I(x, \varepsilon).$$

Meanwhile, since $x + v \in \Omega$, we have

$$a_i^T x + a_i^T v \leq b_i \quad \text{for all } i.$$

The preceding two relations then lead to

$$a_i^T v \leq b_i - a_i^T x \leq \varepsilon \| a_i \| \quad \text{for all } i \in I(x, \varepsilon).$$

Since N is generated by $\mathcal{A} \subseteq \mathbf{A} = \{a_1, \ldots, a_m\}$ and $v_N \neq 0$, by (2.1) and (6.6),

$$\nu_{\min} \| v_N \| \leq \max_{i \in I(x,\varepsilon)} \frac{v^T a_i}{\| a_i \|} \leq \max_{i \in I(x,\varepsilon)} \frac{\varepsilon \| a_i \|}{\| a_i \|} = \varepsilon. \qquad \square$$

For $x \in \Omega$ and $v \in \mathbb{R}^n$, define

$$\hat{\chi}(x; v) = \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v. \tag{B.1}$$

Note from (A.1) that $\chi(x) = \hat{\chi}(x; -\nabla f(x))$. We use v in (B.1) to emphasize that the following results are purely geometric facts about cones and polyhedra.

The following proposition relates $\hat{\chi}(x; v)$ to the projection of v onto the cones $T(x, \varepsilon)$ and $N(x, \varepsilon)$. Roughly speaking, it says that if $\varepsilon > 0$ is small, so that we are only looking at a portion of the boundary very near x, then the projection of v onto $T(x, \varepsilon)$ (i.e., the portion of v pointing into the interior of the feasible region) cannot be small unless $\hat{\chi}(x; v)$ is also small.

PROPOSITION B.2. *If $x \in \Omega$, then for all $\varepsilon \geq 0$,*

$$\hat{\chi}(x; v) \leq \| v_{T(x,\varepsilon)} \| + \frac{\varepsilon}{\nu_{\min}} \| v_{N(x,\varepsilon)} \|,$$

where $\nu_{\min}$ is the constant from (6.6).

Proof. Let $N = N(x, \varepsilon)$ and $T = T(x, \varepsilon)$. Writing v in terms of its polar decomposition, $v = v_N + v_T$, we obtain

$$\hat{\chi}(x; v) = \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v \leq \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_T + \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_N.$$

For the first term on the right-hand side we have

$$\max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_T \leq \| v_T \|.$$

Meanwhile, for any w we have

$$w^T v_N = (w_T + w_N)^T v_N \leq w_N^T v_N$$

since $w_T^T v_N \leq 0$. Thus,

$$\max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_N \leq \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} \| w_N \| \, \| v_N \|.$$

However, since $x + w \in \Omega$, Proposition B.1 tells us that

$$\| w_N \| \leq \frac{\varepsilon}{\nu_{\min}}.$$

Therefore,

$$\hat{\chi}(x; v) \leq \| v_T \| + \frac{\varepsilon}{\nu_{\min}} \| v_N \|. \qquad \square$$

Acknowledgments. We thank Margaret Wright, the associate editor who handled this paper, along with two anonymous referees, for the many useful comments that led to a significant improvement in this presentation.

REFERENCES

[1] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.
[2] A. Ben-Israel, A. Ben-Tal, and S. Zlobec, *Optimality in Nonlinear Programming: A Feasible Directions Approach*, John Wiley & Sons, New York, 1981.
[3] M. J. Box, D. Davies, and W. H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph 5, Oliver & Boyd, Edinburgh, UK, 1969.
[4] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19 (1965), pp. 577–593.
[5] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, *Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints*, SIAM J. Optim., 6 (1996), pp. 674–703.
[6] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MPS/SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.
[7] C. Davis, *Theory of positive linear dependence*, Amer. J. Math., 76 (1954), pp. 733–746.
[8] E. D. Dolan, R. M. Lewis, and V. J. Torczon, *On the local convergence properties of pattern search*, SIAM J. Optim., 14 (2003), pp. 567–583.
[9] K. Fukuda, *cdd and cddplus homepage*. From McGill University, Montreal, Canada, http://www.cs.mcgill.ca/∼fukuda/soft/cdd_home/cdd.html, 2005.
[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
[11] L. Grippo, F. Lampariello, and S. Lucidi, *Global convergence and stabilization of unconstrained minimization methods without derivatives*, J. Optim. Theory Appl., 56 (1988), pp. 385–406.
[12] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms,* I, Springer-Verlag, Berlin, 1993.
[13] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, II, Springer-Verlag, Berlin, 1993.
[14] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8 (1961), pp. 212–229.
[15] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.
[16] T. G. Kolda, R. M. Lewis, and V. Torczon, *A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints*, Technical report SAND2006-5315, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006.
[17] R. M. Lewis, A. Shepherd, and V. Torczon, *Implementing Generating Set Search Methods for Linearly Constrained Minimization*, Technical report WM–CS–2005–01, Department of Computer Science, College of William & Mary, Williamsburg, VA, 2005.
[18] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.
[19] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.
[20] R. M. Lewis and V. Torczon, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM J. Optim., 12 (2002), pp. 1075–1089.

[21] S. Lucidi and M. Sciandrone, *Numerical results for unconstrained optimization without derivatives*, in Nonlinear Optimization and Applications (Proceedings of the International School of Mathematics "G. Stampacchia" 21st Workshop, Erice, Italy, 1995), G. Di Pillo and F. Giannessi, eds., Kluwer Academic/Plenum Publishers, New York, 1996, pp. 261–270.

[22] S. Lucidi and M. Sciandrone, *A derivative-free algorithm for bound constrained optimization*, Comput. Optim. Appl., 21 (2002), pp. 119–142.

[23] S. Lucidi and M. Sciandrone, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM J. Optim., 13 (2002), pp. 97–116.

[24] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Math. Program., 92 (2002), pp. 37–59.

[25] J. H. May, *Linearly Constrained Nonlinear Programming: A Solution Method That Does Not Require Analytic Derivatives*, Ph.D. thesis, Yale University, New Haven, CT, 1974.

[26] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[27] J.-J. Moreau, *Décomposition orthgonale d'un espace hilbertien selon deux cônes mutuellement polaires*, C. R. Acad. Sci. Paris, 255 (1962), pp. 238–240.

[28] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw–Hill, New York, 1996.

[29] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, 1999.

[30] C. J. Price and I. D. Coope, *Frames and grids in unconstrained and linearly constrained optimization: A nonsmooth approach*, SIAM J. Optim., 14 (2003), pp. 415–438.

[31] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[32] W. H. Swann, *Direct search methods*, in Numerical Methods for Unconstrained Optimization, W. Murray, ed., Academic Press, London, New York, 1972, pp. 13–28.

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

[34] W. Yu and Y. Li, *A direct search method by the local positive basis for linearly constrained optimization*, Chinese Ann. Math., 2 (1981), pp. 139–146.

[35] W. I. Zangwill, *Nonlinear Programming; A Unified Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[36] G. Zoutendijk, *Mathematical Programming Methods*, North–Holland, Amsterdam, 1976.