



ELSEVIER

Journal of Computational and Applied Mathematics 124 (2000) 191–207

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Direct search methods: then and now

Robert Michael Lewis^{a,1}, Virginia Torczon^{a,b,*,2}, Michael W. Trosset^{c,a}

^aICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA

^bDepartment of Computer Science, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187-8795, USA

^cDepartment of Mathematics, College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187-8795, USA

Received 1 July 1999; received in revised form 23 February 2000

Abstract

We discuss direct search methods for unconstrained optimization. We give a modern perspective on this classical family of derivative-free algorithms, focusing on the development of direct search methods during their golden age from 1960 to 1971. We discuss how direct search methods are characterized by the absence of the construction of a model of the objective. We then consider a number of the classical direct search methods and discuss what research in the intervening years has uncovered about these algorithms. In particular, while the original direct search methods were consciously based on straightforward heuristics, more recent analysis has shown that in most — but not all — cases these heuristics actually suffice to ensure global convergence of at least one subsequence of the sequence of iterates to a first-order stationary point of the objective function. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Derivative-free optimization; Direct search methods; Pattern search methods

1. Introduction

Robert Hooke and T.A. Jeeves coined the phrase “direct search” in a paper that appeared in 1961 in the *Journal of the Association of Computing Machinery* [12]. They provided the following description of direct search in the introduction to their paper:

* Corresponding author.

E-mail addresses: buckaroo@icase.edu (R.M. Lewis), va@cs.wm.edu (V. Torczon), trosset@math.wm.edu (M.W. Trosset).

¹This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046.

²This research was supported by the National Science Foundation under Grant CCR-9734044 and by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046, while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE).

0377-0427/00/\$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.

PII: S0377-0427(00)00423-4

We use the phrase “direct search” to describe sequential examination of trial solutions involving comparison of each trial solution with the “best” obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be. The phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques of classical analysis except where there is a demonstrable advantage in doing so.

To a modern reader, this preference for avoiding techniques of classical analysis “except where there is a demonstrable advantage in doing so” quite likely sounds odd. After all, the success of quasi-Newton methods, when applicable, is now undisputed. But consider the historical context of the remark by Hooke and Jeeves. Hooke and Jeeves’ paper appeared five years before what are now referred to as the Armijo–Goldstein–Wolfe conditions were introduced and used to show how the method of steepest descent could be modified to ensure global convergence [1,11,29]. Their paper appeared only two years after Davidon’s unpublished report on using secant updates to derive quasi-Newton methods [8], and two years before Fletcher and Powell published a similar idea in *The Computer Journal* [10]. So in 1961, this preference on the part of Hooke and Jeeves was not without justification.

Forty years later, the question we now ask is: why are direct search methods still in use? Surely, this seemingly hodge-podge collection of methods based on heuristics, which generally appeared without any attempt at a theoretical justification, should have been superseded by more “modern” approaches to numerical optimization.

To a large extent direct search methods *have* been replaced by more sophisticated techniques. As the field of numerical optimization has matured, and software has appeared which eases the ability of consumers to make use of these more sophisticated numerical techniques, many users now routinely rely on some variant of a globalized quasi-Newton method.

Yet direct search methods persist for several good reasons. First and foremost, direct search methods have remained popular because they work well in practice. In fact, many of the direct search methods are based on surprisingly sound heuristics that fairly recent analysis demonstrates guarantee global convergence behavior analogous to the results known for globalized quasi-Newton techniques. Direct search methods succeed because many of them — including the direct search method of Hooke and Jeeves — can be shown to rely on techniques of classical analysis in ways that are not readily apparent from their original specifications.

Second, quasi-Newton methods are not applicable to all nonlinear optimization problems. Direct search methods have succeeded when more elaborate approaches failed. Features unique to direct search methods often avoid the pitfalls that can plague more sophisticated approaches.

Third, direct search methods can be the method of first recourse, even among well-informed users. The reason is simple enough: direct search methods are reasonably straightforward to implement and can be applied almost immediately to many nonlinear optimization problems. The requirements from a user are minimal and the algorithms themselves require the setting of few parameters. It is not unusual for complex optimization problems to require further software development before quasi-Newton methods can be applied (e.g., the development of procedures to compute derivatives or the proper choice of perturbation for finite-difference approximations to gradients). For such problems, it can make sense to begin the search for a minimizer using a direct search method with known global convergence properties, while undertaking the preparations for the quasi-Newton method. When the preparations for the quasi-Newton method have

been completed, the best known result from the direct search calculation can be used as a “hot start” for one of the quasi-Newton approaches, which enjoy superior local convergence properties. Such hybrid optimization strategies are as old as the direct search methods themselves [21].

We have three goals in this review. First, we want to outline the features of direct search that distinguish these methods from other approaches to nonlinear optimization. Understanding these features will go a long way toward explaining their continued success. Second, as part of our categorization of direct search, we suggest three basic approaches to devising direct search methods and explain how the better known classical techniques fit into one of these three camps. Finally, we review what is now known about the convergence properties of direct search methods. The heuristics that first motivated the development of these techniques have proven, with time, to embody enough structure to allow — in most instances — analysis based on now standard techniques. We are never quite sure if the original authors appreciated just how reliable their techniques would prove to be; we would like to believe they did. Nevertheless, we are always impressed by new insights to be gleaned from the discussions to be found in the original papers. We enjoy the perspective of forty intervening years of optimization research. Our intent is to use this hindsight to place direct search methods on a firm standing as one of many useful classes of techniques available for solving nonlinear optimization problems.

Our discussion of direct search algorithms is by no means exhaustive, focusing on those developed during the dozen years from 1960 to 1971. Space also does not permit an exhaustive bibliography. Consequently, we apologize in advance for omitting reference to a great deal of interesting work.

2. What is “direct search”?

For simplicity, we restrict our attention in the paper to unconstrained minimization:

$$\text{minimize } f(x), \tag{2.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$. We assume that f is continuously differentiable, but that information about the gradient of f is either unavailable or unreliable.

Because direct search methods neither compute nor approximate derivatives, they are often described as “derivative-free”. However, as argued in [27], this description does not fully characterize what constitutes “direct search”.

Historically, most approaches to optimization have appealed to a familiar “technique of classical analysis”, the Taylor’s series expansion of the objective function. In fact, one can classify most methods for numerical optimization according to how many terms of the expansion are exploited. Newton’s method, which assumes the availability of first and second derivatives and uses the second-order Taylor polynomial to construct local quadratic approximations of f , is a second-order method. Steepest descent, which assumes the availability of first derivatives and uses the first-order Taylor polynomial to construct local linear approximations of f , is a first-order method. In this taxonomy, “zero-order methods” do not require derivative information and do not construct approximations of f . They are direct search methods, which indeed are often called zero-order methods in the engineering optimization community.

Direct search methods rely exclusively on values of the objective function, but even this property is not enough to distinguish them from other optimization methods. For example, suppose that one would like to use steepest descent, but that gradients are not available. In this case, it is customary to replace the actual gradient with an estimated gradient. If it is possible to observe exact values of the objective function, then the gradient is usually estimated by finite differencing. This is the case of numerical optimization, with which we are concerned herein. If function evaluation is uncertain, then the gradient is usually estimated by designing an appropriate experiment and performing a regression analysis. This occurs, for instance, in *response surface methodology* in stochastic optimization. Response surface methodology played a crucial role in the pre-history of direct search methods, a point to which we return shortly. Both approaches rely exclusively on values of the objective function, yet each is properly classified as a first-order method. What, then, is a direct search method? What exactly does it mean to say that direct search methods neither compute nor approximate derivatives?

Although instructive, we believe that a taxonomy based on Taylor expansions diverts attention from the basic issue. As in [27], we prefer here to emphasize the construction of approximations, not the mechanism by which they are constructed. The optimization literature contains numerous examples of methods that do not require derivative information and approximate the objective function without recourse to Taylor expansions. Such methods are “derivative-free”, but they are not direct searches. What is the distinction?

Hooke and Jeeves considered that direct search involves the comparison of each trial solution with the best previous solution. Thus, a distinguishing characterization of direct search methods (at least in the case of unconstrained optimization) is that they do not require numerical function values: the relative rank of objective values is sufficient. That is, direct search methods for unconstrained optimization depend on the objective function only through the relative ranks of a countable set of function values. This means that direct search methods can accept new iterates that produce simple decrease in the objective. This is in contrast to the Armijo–Goldstein–Wolfe conditions for quasi-Newton line search algorithms, which require that a sufficient decrease condition be satisfied. Another consequence of this characterization of direct search is that it precludes the usual ways of approximating f , since access to numerical function values is not presumed.

There are other reasons to distinguish direct search methods within the larger class of derivative-free methods. We have already remarked that response surface methodology constructs local approximations of f by regression. Response surface methodology was proposed in 1951, in the seminal paper [4], as a variant of steepest descent (actually steepest ascent, since the authors were maximizing). In 1957, concerned with the problem of improving industrial processes and the shortage of technical personnel, Box [3] outlined a less sophisticated procedure called *evolutionary operation*. Response surface methodology relied on esoteric experimental designs, regression, and steepest ascent; evolutionary operation relied on simple designs and the direct comparison of observed function values. Spendley et al. [21] subsequently observed that the designs in [3] could be replaced with simplex designs and suggested that evolutionary operation could be automated and used for numerical optimization. As discussed in Section 3.2, their algorithm is still in use and is the progenitor of the simplex algorithm of Nelder and Mead [17], the most famous of all direct search methods. Thus, the distinction that G.E.P. Box drew in the 1950s, between response surface methodology and evolutionary operation, between approximating f and comparing values of f , played a crucial role in the development of direct search methods.

3. Classical direct search methods

We organize the popular direct search methods for unconstrained minimization into three basic categories. For a variety of reasons, we focus on the classical direct search methods, those developed during the period 1960–1971. The restriction is part practical, part historical.

On the practical side, we will make the distinction between *pattern search methods*, *simplex methods* (and here we do *not* mean the simplex method for linear programming), and *methods with adaptive sets of search directions*. The direct search methods that one finds described most often in texts can be partitioned relatively neatly into these three categories. Furthermore, the early developments in direct search methods more or less set the stage for subsequent algorithmic developments. While a wealth of variants on these three basic approaches to designing direct search methods have appeared in subsequent years — largely in the applications literature — these newer methods are modifications of the basic themes that had already been established by 1971. Once we understand the motivating principles behind each of the three approaches, it is a relatively straightforward matter to devise variations on these three themes.

There are also historical reasons for restricting our attention to the algorithmic developments in the 1960s. Throughout those years, direct search methods enjoyed attention in the numerical optimization community. The algorithms proposed were then (and are now) of considerable practical importance. As their discipline matured, however, numerical optimizers became less interested in heuristics and more interested in formal theories of convergence. At a joint IMA/NPL conference that took place at the National Physics Laboratory in England in January 1971, Swann [23] surveyed the status of direct search methods and concluded with this apology:

Although the methods described above have been developed heuristically and no proofs of convergence have been derived for them, in practice they have generally proved to be robust and reliable in that only rarely do they fail to locate at least a local minimum of a given function, although sometimes the rate of convergence can be very slow.

Swann's remarks address an unfortunate perception that would dominate the research community for years to come: that whatever successes they enjoy in practice, direct search methods are theoretically suspect. Ironically, in the same year as Swann's survey, convergence results for direct search methods began to appear, though they seem not to have been widely known, as we discuss shortly. Only recently, in the late 1990s, as computational experience has evolved and further analysis has been developed, has this perception changed [30].

3.1. Pattern search

In his belated preface for ANL 5990 [8], Davidon described one of the most basic of pattern search algorithms, one so simple that it goes without attribution:

Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter

further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure,

Pattern search methods are characterized by a series of *exploratory moves* that consider the behavior of the objective function at a pattern of points, all of which lie on a rational lattice. In the example described above, the unit coordinate vectors form a basis for the lattice and the current magnitude of the steps (it is convenient to refer to this quantity as Δ_k) dictates the resolution of the lattice. The exploratory moves consist of a systematic strategy for visiting the points in the lattice in the immediate vicinity of the current iterate.

It is instructive to note several features of the procedure used by Fermi and Metropolis. First, it does not model the underlying objective function. Each time that a parameter was varied, the scientists asked: was there improvement in the fit to the experimental data. A simple “yes” or “no” answer determined which move would be made. Thus, the procedure is a direct search. Second, the parameters were varied by steps of predetermined magnitude. When the step size was reduced, it was multiplied by one-half, thereby ensuring that all iterates remained on a rational lattice. This is the key feature that makes the direct search a pattern search. Third, the step size was reduced *only* when no increase or decrease in any one parameter further improved the fit, thus ensuring that the step sizes were not decreased prematurely. This feature is another part of the formal definition of pattern search in [26] and is crucial to the convergence analysis presented therein.

3.1.1. Early analysis

By 1971, a proof of global convergence for this simple algorithm existed in the optimization text [18], where the technique goes by the name *method of local variations*. Specifically, Polak proved the following result:

Theorem 3.1. *If $\{x_k\}$ is a sequence constructed by the method of local variations, then any accumulation point x' of $\{x_k\}$ satisfies $\nabla f(x') = 0$. (By assumption, $f(x)$ is at least once continuously differentiable.)*

Polak’s result is as strong as any of the contemporaneous global convergence results for either steepest descent or a globalized quasi-Newton method. However, to establish global convergence for these latter methods, one must enforce either sufficient decrease conditions (the Armijo–Goldstein–Wolfe conditions) or a fraction of Cauchy decrease condition — all of which rely on explicit numerical function values, as well as explicit approximations to the directional derivative at the current iterate. What is remarkable is that we have neither for direct search methods, yet can prove convergence.

What Polak clearly realized, though his proof does not make explicit use of this fact, is that all of the iterates for the method of local variations lie on a rational lattice (one glance at the figure on p. 43 of his text confirms his insight). The effect, as he notes, is that the method can construct only a *finite* number of intermediate points before reducing the step size by one-half. Thus the algorithm “cannot jam up at a point” — precisely, the pathology of premature convergence that the Armijo–Goldstein–Wolfe conditions are designed to preclude.

Polak was not alone in recognizing that pattern search methods contain sufficient structure to support a global convergence result. In the same year, C ea also published an optimization text [7] in which he provided a proof of global convergence for the pattern search algorithm of Hooke and Jeeves [12]. The assumptions used to establish convergence were stronger (in addition to the assumption that $f \in C^1$, it is assumed that f is strictly convex and that $f(x) \rightarrow +\infty$ as $\|x\| \rightarrow +\infty$). Nevertheless, it is established that the sequence of iterates produced by the method of Hooke and Jeeves converges to the unique minimizer of f — again with an algorithm that has no explicit recourse to the directional derivative and for which ranking information is sufficient.

Both Polak’s and C ea’s results rely on the fact that when either of these two algorithms reach the stage where the decision is made to reduce Δ_k , which controls the length of the steps, sufficient information about the local behavior of the objective has been acquired to ensure that the reduction is not premature. Specifically, neither the method of local variations nor the pattern search algorithm of Hooke and Jeeves allow Δ_k to be reduced until it has been verified that

$$f(x_k) \leq f(x_k \pm \Delta_k e_i), \quad i = \{1, \dots, n\},$$

where e_i denotes the i th unit coordinate vector. This plays a critical role in both analyses. As long as x_k is not a stationary point of f , then at least one of the $2n$ directions defined by $\pm e_i$, $i \in \{1, \dots, n\}$ must be a direction of descent. Thus, once Δ_k is sufficiently small, we are guaranteed that either $f(x_k + \Delta_k e_i) < f(x_k)$ or $f(x_k - \Delta_k e_i) < f(x_k)$ for at least one $i \in \{1, \dots, n\}$.

The other early analysis worth noting is that of Berman [2]. In light of later developments, Berman’s work is interesting precisely because he realized that if he made explicit use of a rational lattice structure, he could construct algorithms that produce minimizers to continuous nonlinear functions that might not be differentiable. For example, if f is continuous and strongly unimodal, he argues that convergence to a minimizer is guaranteed.

In the algorithms formulated and analyzed by Berman, the rational lattice plays an explicit role. The lattice L determined by x_0 (the initial iterate) and Δ_0 (the initial resolution of the lattice) is defined by $L(x_0, \Delta_0) = \{x \mid x = x_0 + \Delta_0 \lambda, \lambda \in A\}$, where A is the lattice of integral points of \mathbb{R}^n . Particularly important is the fact that the lattices used successively to approximate the minimizer have the following property: if $L_k = L(x_k, \Delta_k)$, where $\Delta_k = \Delta_0 / \tau^k$ and $\tau > 1$ denotes a positive integer, then $L_k \subset L_{k+1}$. The important ramification of this fact is that $\{x_0, x_1, x_2, \dots, x_k\} \subset L_{k+1}$, for any choice of k , thus ensuring the finiteness property to which Polak alludes, and which also plays an important role in the more recent analysis for pattern search.

Before moving on to the more recent results, however, we close with some observations about this early work. First, it is with no small degree of irony that we note that all three results [2,7,18] are contemporaneous with Swann’s remark that no proofs of convergence had been derived for direct search methods. However, each of these results was developed in isolation. None of the three authors appears to have been aware of the work of the others; none of the works contains citations of the other two and there is nothing in the discussion surrounding each result to suggest that any one of the authors was aware of the more-or-less simultaneous developments by the other two. Furthermore, these results have passed largely unknown and unreferenced in the nonlinear optimization literature. They have not been part of the “common wisdom” and so it was not unusual, until quite recently, to still hear claims that direct search methods had “been developed heuristically and no proofs of convergence have been derived for them”.

Yet all the critical pieces needed for a more general convergence theory of pattern search had been identified by 1971. The work of Polak and C ea was more modest in scope in that each was proving convergence for a single, extant algorithm, already widely in use. Berman’s work was more ambitious in that he was defining a general principle with the intent of deriving any number of new algorithms tailored to particular assumptions about the problem to be solved. What remained to be realized was that all this work could be unified under one analysis — and generalized even further to allow more algorithmic perturbations.

3.1.2. *Recent analysis*

Recently, a general theory for pattern search [26] extended a global convergence analysis [25] of the multidirectional search algorithm [24]. Like the simplex algorithms of Section 3.2, multidirectional search proceeds by reflecting a simplex ($n + 1$ points in \mathbb{R}^n) through the centroid of one of the faces. However, unlike the simplex methods discussed in Section 3.2, multidirectional search is also a pattern search.

In fact, the essential ingredients of the general theory has already been identified in [2,7,18]. First, the pattern of points from which one selects trial points at which to evaluate the objective function must be sufficiently rich to ensure at least one direction of descent if x_k is not a stationary point of f . For C ea and Polak, this meant a pattern that included points of the form $x'_k = x_k \pm \Delta_k e_i$, $i \in \{1, \dots, n\}$, where the e_i are the unit coordinate vectors. For Berman, it meant requiring A to be the lattice integral points of \mathbb{R}^n , i.e., requiring that the basis for the lattice be the identity matrix $I \in \mathbb{R}^{n \times n}$.

In [26], these conditions were relaxed to allow any nonsingular matrix $B \in \mathbb{R}^{n \times n}$ to be the basis for the lattice. In fact, we can allow patterns of the form $x'_k = x_k + \Delta_k B \gamma'_k$, where γ'_k is an integral vector, so that the direction of the step is determined by forming an integral combination of the columns of B . The special cases studied by C ea and Polak are easily recovered by choosing $B \equiv I$ and $\gamma'_k = \pm e_i$, $i \in \{1, \dots, n\}$.

Second, an essential ingredient of each of the analyses is the requirement that Δ_k not be reduced if the objective function can be decreased by moving to one of the x'_k . Generalizations of this requirement were considered in [26,15]. This restriction acts to prevent premature convergence to a nonstationary point.

Finally, we restrict the manner by which Δ_k is rescaled. The conventional choice, used by both C ea and Polak, is to divide Δ_k by two, so that $\Delta_k = \Delta_0/2^k$. Somewhat more generally, Berman allowed dividing by any integer $\tau > 1$, so that (for example) one could have $\Delta_k = \Delta_0/3^k$. In fact, even greater generality is possible. For $\tau > 1$, we allow $\Delta_{k+1} = \tau^w \Delta_k$, where w is any integer in a designated finite set. Then there are three possibilities:

1. $w < 0$. This decreases Δ_k , which is only permitted under certain conditions (see above). When it is permitted, then $L_k \subset L_{k+1}$, the relation considered by Berman.
2. $w = 0$. This leaves Δ_k unchanged, so that $L_k = L_{k+1}$.
3. $w > 0$. This increases Δ_k , so that $L_{k+1} \subset L_k$.

It turns out that what matters is not the relation of L_k to L_{k+1} , but the assurance that there exists a single lattice $L_i \in \{L_0, L_1, \dots, L_k, L_{k+1}\}$, for which $L_j \subseteq L_i$ for all $j = 0, \dots, k + 1$. This implies that $\{x_0, \dots, x_k\} \subset L_i$, which in turn plays a crucial role in the convergence analysis.

Exploiting the essential ingredients that we have identified, one can derive a general theory of global convergence. The following result says that at least one subsequence of iterates converges to a stationary point of the objective function.

Theorem 3.2. *Assume that $L(x_0) = \{x \mid f(x) \leq f(x_0)\}$ is compact and that f is continuously differentiable on a neighborhood of $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by a generalized pattern search algorithm,*

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

Under only slightly stronger hypotheses, one can show that every limit point of $\{x_k\}$ is a stationary point of f , generalizing Polak's convergence result. Details of the analysis can be found in [26,15]; [14] provides an expository discussion of the basic argument.

3.2. Simplex search

Simplex search methods are characterized by the simple device that they use to guide the search.

The first of the simplex methods is due to Spendley et al. [21] in a paper that appeared in 1962. They were motivated by the fact that earlier direct search methods required anywhere from $2n$ to 2^n objective evaluations to complete the search for improvement on the iterate. Their observation was that it should take no more than $n + 1$ values of the objective to identify a downhill (or uphill) direction. This makes sense, since $n + 1$ points in the graph of $f(x)$ determine a plane, and $n + 1$ values of $f(x)$ would be needed to estimate $\nabla f(x)$ via finite differences. At the same time, $n + 1$ points determine a simplex. This leads to the basic idea of simplex search: construct a nondegenerate simplex in \mathbb{R}^n and use the simplex to drive the search.

A simplex is a set of $n + 1$ points in \mathbb{R}^n . Thus one has a triangle in \mathbb{R}^2 , a tetrahedron in \mathbb{R}^3 , etc. A nondegenerate simplex is one for which the set of edges adjacent to any vertex in the simplex forms a basis for the space. In other words, we want to be sure that any point in the domain of the search can be constructed by taking linear combinations of the edges adjacent to any given vertex.

Not only does the simplex provide a frugal design for sampling the space, it has the added feature that if one replaces a vertex by reflecting it through the centroid of the opposite face, then the result is also a simplex, as shown in Fig. 1. This, too, is a frugal feature because it means that one can proceed parsimoniously, reflecting one vertex at a time, in the search for an optimizer.

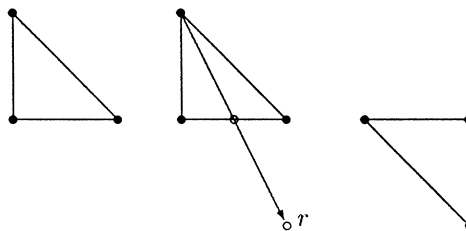


Fig. 1. The original simplex, the reflection of one vertex through the centroid of the opposite face, and the resulting reflection simplex.

Once an initial simplex is constructed, the single move specified in the original Spendley et al. simplex algorithm is that of reflection. This move first identifies the “worst” vertex in the simplex (i.e., the one with the least desirable objective value) and then reflects the worst simplex through the centroid of the opposite face. If the reflected vertex is still the worst vertex, then next choose the “next worst” vertex and repeat the process. (A quick review of Fig. 1 should confirm that if the reflected vertex is not better than the next worst vertex, then if the “worst” vertex is once again chosen for reflection, it will simply be reflected back to where it started, thus creating an infinite cycle.)

The ultimate goals are either to replace the “best” vertex (i.e., the one with the most desirable objective value) or to ascertain that the best vertex is a candidate for a minimizer. Until then, the algorithm keeps moving the simplex by flipping some vertex (other than the best vertex) through the centroid of the opposite face.

The basic heuristic is straightforward in the extreme: we move a “worse” vertex in the general direction of the remaining vertices (as represented by the centroid of the remaining vertices), with the expectation of eventual improvement in the value of the objective at the best vertex. The questions then become: do we have a new candidate for a minimizer and are we at or near a minimizer?

The first question is easy to answer. When a reflected vertex produces strict decrease on the value of the objective at the best vertex, we have a new candidate for a minimizer; once again the simple decrease rule is in effect.

The answer to the second question is decidedly more ambiguous. In the original paper, Spendley, Hext, and Himsworth illustrate — in two dimensions — a circling sequence of simplices that could be interpreted as indicating that the neighborhood of a minimizer has been identified. We see a similar example in Fig. 2, where a sequence of five reflections brings the search back to where it started, without replacing x_k , thus suggesting that x_k may be in the neighborhood of a stationary point.

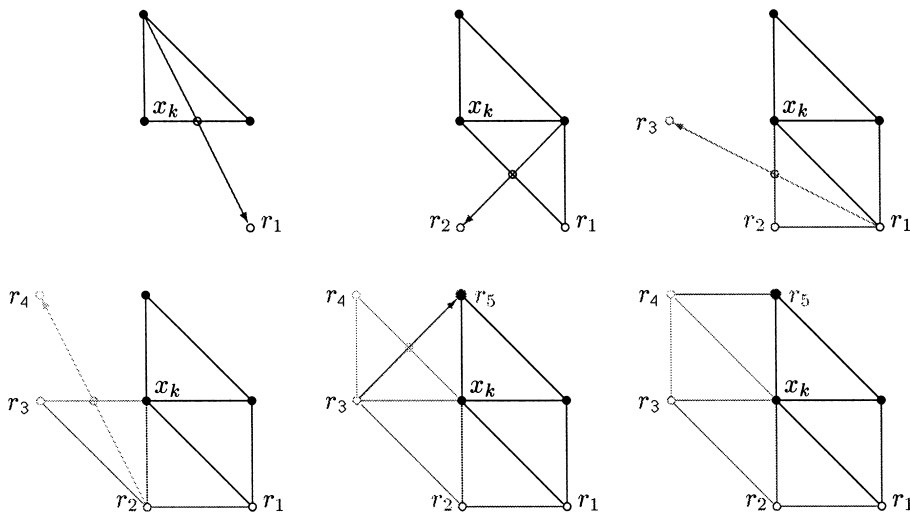


Fig. 2. A sequence of reflections $\{r_1, r_2, r_3, r_4, r_5\}$, each of which fails to replace the best vertex x_k , which brings the search back to the simplex from which this sequence started.

The picture in two dimensions is somewhat misleading since the fifth reflection maps back onto the worst vertex in the original simplex — a situation that only occurs in either one or two dimensions. So Spendley, Hext, and Himesworth give a heuristic formula for when the simplex has flipped around the current best vertex long enough to conclude that the neighborhood of a minimizer has been identified. When this situation has been detected, they suggest two alternatives: either reduce the lengths of the edges adjacent to the “best” vertex and resume the search or resort to a higher-order method to obtain faster local convergence.

The contribution of Nelder and Mead [17] was to turn simplex search into an optimization algorithm with additional moves designed to accelerate the search. In particular, it was already well-understood that the reflection move preserved the original shape of the simplex — regardless of the dimension. What Nelder and Mead proposed was to supplement the basic reflection move with additional options designed to accelerate the search by deforming the simplex in a way that they suggested would better adapt to the features of the objective function. To this end, they added what are known as expansion and contraction moves, as shown in Fig. 3.

We leave the full details of the logic of the algorithm to others; a particularly clear and careful description, using modern algorithmic notation, can be found in [13]. For our purposes, what is important to note is that the expansion step allows for a more aggressive move by doubling the length of the step from the centroid to the reflection point, whereas the contraction steps allow for more conservative moves by halving the length of the step from the centroid to either the reflection point or the worst vertex. Furthermore, in addition to allowing these adaptations within a single iteration, these new possibilities have repercussions for future iterations as they deform (or, as the rationale goes, adapt) the shape of the original simplex.

Nelder and Mead also resolved the question of what to do if none of the steps tried bring acceptable improvement by adding a shrink step: when all else fails, reduce the lengths of the edges adjacent to the current best vertex by half, as is also illustrated in Fig. 3.

The Nelder–Mead simplex algorithm has enjoyed enduring popularity. Of all the direct search methods, the Nelder–Mead simplex algorithm is the one most often found in numerical software packages. The original paper by Nelder and Mead is a Science Citation Index classic, with several thousand references across the scientific literature in journals ranging from *Acta Anaesthesiologica*

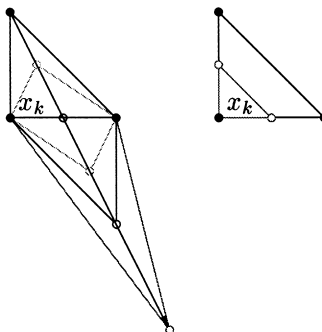


Fig. 3. The original simplex, with the reflection, expansion, and two possible contraction simplices, along with the shrink step toward the best vertex x_k , when all else fails.

Scandinavica to *Zhurnal Fizicheskio Khimii*. In fact, there is an entire book from the chemical engineering community devoted to simplex search for optimization [28].

So why bother with looking any further? Why not rely exclusively on the Nelder–Mead simplex method if one is going to employ a direct search method? The answer: there is the outstanding question regarding the robustness of the Nelder–Mead simplex method that has long troubled numerical optimizers. When the method works, it can work very well indeed, often finding a solution in far fewer evaluations of the objective function than other direct search methods. But it can also fail. One can see this in the applications literature, fairly early on, frequently reported as no more than “slow” convergence. A systematic study of Nelder–Mead, when applied to a suite of standard optimization test problems, also reported occasional convergence to a nonstationary point of the function [24]; the one consistent observation to be made was that in these instances the deformation of the simplex meant that the search direction (i.e., the direction defined along the worst vertex toward the centroid of the remaining vertices) became numerically orthogonal to the gradient.

These observations about the behavior of Nelder–Mead in practice led to two, relatively recent, investigations. The first [13], strives to investigate what *can* be proven about the asymptotic behavior of Nelder–Mead. The results show that in \mathbb{R}^1 , the algorithm is robust; under standard assumptions, convergence to a stationary point is guaranteed. Some general properties in higher dimensions can also be proven, but none that guarantee global convergence for problems in higher dimensions.

This is not surprising in light of a second recent result by McKinnon [16]. He shows with several examples that limits exist on proving global convergence for Nelder–Mead: to wit, the algorithm can fail on smooth (C^2) convex objectives in two dimensions.

This leaves us in the unsatisfactory situation of reporting that no general convergence results exist for the simplex methods of either Spendley et al. or Nelder and Mead — despite the fact that they are two of the most popular and widely used of the direct search methods. Further, McKinnon’s examples indicate that it will not be possible to prove global convergence for the Nelder–Mead simplex algorithm in higher dimensions. On the other hand, the mechanism that leads to failure in McKinnon’s counterexample does not seem to be the mechanism by which Nelder–Mead typically fails in practice. This leaves the question of why Nelder–Mead fails in practice unresolved.

3.3. Methods with adaptive sets of search directions

The last family of classical methods we consider includes Rosenbrock’s and Powell’s methods. These algorithms attempt to accelerate the search by constructing directions designed to use information about the curvature of the objective obtained during the course of the search.

3.3.1. Rosenbrock’s method

Of these methods, the first was due to Rosenbrock [20]. Rosenbrock’s method was quite consciously derived to cope with the peculiar features of Rosenbrock’s famous “banana function”, the minimizer of which lies inside a narrow, curved valley. Rosenbrock’s method proceeds by a series of stages, each of which consists of a number of exploratory searches along a set of directions that are fixed for the given stage, but which are updated from stage to stage to make use of information acquired about the objective.

The initial stage of Rosenbrock’s method begins with the coordinate directions as the search directions. It then conducts searches along these directions, cycling over each in turn, moving to new

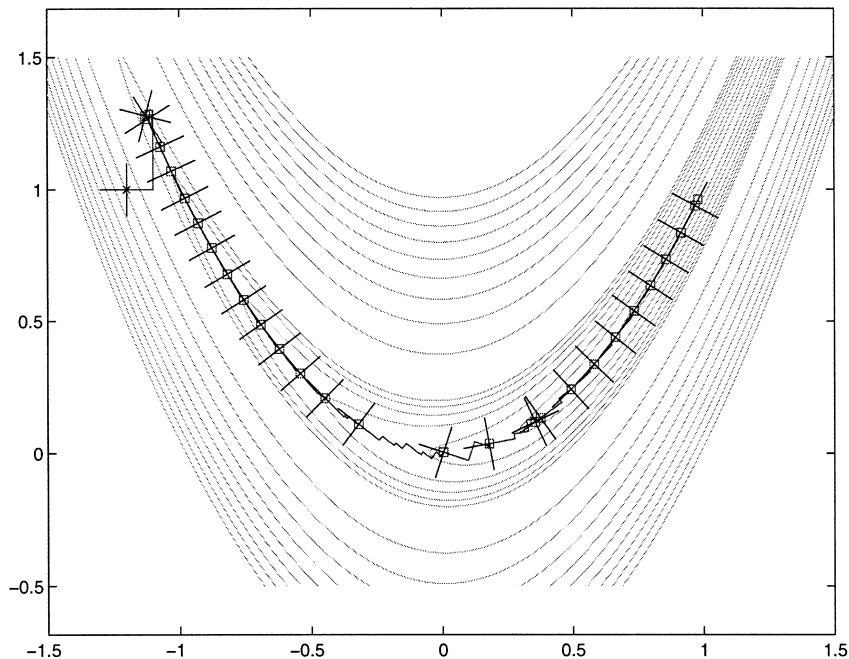


Fig. 4. Rosenbrock's algorithm in action.

iterates that yield successful steps (an unsuccessful step being one that leads to a less desirable value of the objective). This continues until there has been at least one successful and one unsuccessful step in each search direction. Once this occurs, the current stage terminates. As is the case for direct search methods, numerical values of the objective are not necessary in this process. If the objective at any of these steps is perceived as being an improvement over the objective at the current best point, we move to the new point.

At the next stage, rather than repeating the search process with the *same* set of orthogonal vectors, as is done for the method of local variations, Rosenbrock *rotates* the set of directions to capture information about the objective ascertained during the course of the earlier moves. Specifically, he takes advantage of the fact that a nonzero step from the iterate at the beginning of the previous stage to the iterate at the start of the new stage suggests a good direction of descent — or, at the very least, a promising direction — so in the new stage, he makes sure that this particular direction is included in the set of directions along which the search will be conducted. (This heuristic is particularly apt for following the bottom of the valley that leads to the minimizer of the banana function.) Rosenbrock imposes the condition that the set of search directions always be an orthogonal set of n vectors so that the set of vectors remains nicely linearly independent. The new set of orthonormal vectors is generated using the Gram-Schmidt orthonormalization procedure, with the “promising” direction from the just-completed stage used as the first vector in the orthonormalization process.

Rosenbrock's method as applied to his banana function is depicted in Fig. 4. The iterate at the beginning of each stage is indicated with a square. Superimposed on these iterates are the search directions for the new stage. Note how quickly the search adapts to the narrow valley; within three

stages the search directions reflect this feature. Also notice how the search directions change to allow the algorithm to turn the corner in the valley and continue to the solution.

Updating the set of search directions for Rosenbrock's method entails slightly more complexity than that which appears in any of the other two families of direct search methods we have surveyed. On the other hand, the example of the banana function makes the motivation for this additional work clear: adapting the entire set of search directions takes advantage of what has been learned about the objective during the course of the search.

3.3.2. *The variant of Davies, Swann, and Campey*

A refinement to Rosenbrock's algorithm was proposed in [22].³ Davies et al. noted that there was merit to carrying out a sequence of more sophisticated one-dimensional searches along each of the search directions than those performed in Rosenbrock's original algorithm.

As described in [23], the more elaborate line search of Davies et al. first takes steps of increasing multiples of some fixed value Δ along a direction from the prescribed set until a bracket for the (one-dimensional) minimizer is obtained. This still corresponds to our definition of a direct search method.

However, once a bracket for the one-dimensional minimizer has been found, a "single quadratic interpolation is made to predict the position of the minimum more closely" [23]. This is the construction of a model of the objective, and to do this, numerical values for the objective must be in hand. Thus, this final move within an iteration disqualifies the method of Davies, Swann, and Campey as a direct search method by our characterization. Nonetheless, this strategy is undeniably appealing, and its authors aver that this variant of Rosenbrock's method is more generally efficient than the original [6].

3.3.3. *Powell's method*

In a paper appearing the same year as the report in [22], Powell [19] outlined a method for finding minimizers without calculating derivatives. By our definition, it is a derivative-free, rather than a direct search method, for modeling is at the heart of the approach. The explicit goal is to ensure that if the method is applied to a convex quadratic function, conjugate directions are chosen with the goal of accelerating convergence. In this sense, Powell's algorithm may be viewed as a derivative-free version of nonlinear conjugate gradients.

Like Rosenbrock's method, Powell's method proceeds in stages. Each stage consists of a sequence of $n + 1$ one-dimensional searches. The one-dimensional searches are conducted by finding the exact minimizer of a quadratic interpolant computed for each direction (hence our classification of the method as a derivative-free, but not direct search, method). The first n searches are along each of a set of linearly independent directions. The last search is along the direction connecting the point obtained at the end of the first n searches with the starting point of the stage. At the end of the stage, one of the first n search directions is replaced by the last search direction. The process then repeats at the next stage.

Powell showed that if the objective is a convex quadratic, then the set of directions added at the last step of each stage forms a set of conjugate directions (provided they remain linearly

³ A paper the authors have been unable to locate. The authors would be very much obliged to any reader who has a copy of the original report and would forward a photocopy to us.

independent). Powell used this, in turn, to show that his method possessed what was known then as the “Q-property”. An algorithm has the Q-property if it will find the minimizer of a convex quadratic in a finite number of iterations. That is, the Q-property is the finite termination property for convex quadratics such as that exhibited by the conjugate gradient algorithm. In the case of Powell’s method, one obtains finite termination in n stages for convex quadratics.

Zangwill [31] gave a modification of Powell’s method that avoids the possibility of linearly dependent search directions. Zangwill further proved convergence to minimizers of strictly convex functions (though not in a finite number of steps).

To the best of our knowledge, Powell’s method marks the first time that either a direct search or a derivative-free method appeared with any attendant convergence analysis. The appeal of the *explicit* modeling of the objective such as that used in the line-searches in Powell’s method is clear: it makes possible strong statements about the behavior of the optimization method. We can expect the algorithm to quickly converge to a minimizer once in a neighborhood of a solution on which the objective is essentially quadratic.

Finite termination on quadratic objectives was a frequently expressed concern within the optimization community during the 1960s and 1970s. The contemporary numerical results produced by the optimization community (for analytical, closed-form objective functions, it should be noted) evidence this concern. Most reports of the time [5,9] confirm the supposed superiority of the modeling-based approach, with guaranteed finite termination as embodied in Powell’s derivative-free conjugate directions algorithm.

Yet forty years later, direct search methods, “which employ no techniques of analysis except where there is a demonstrable advantage in doing so”, remain popular, as indicated by any number of measures: satisfied users, literature citations, and available software. What explains this apparently contradictory historical development?

4. Conclusion

Direct search methods remain popular because of their simplicity, flexibility, and reliability. Looking back at the initial development of direct search methods from a remove of forty years, we can firmly place what is now known and understood about these algorithms in a broader context.

With the exception of the simplex-based methods specifically discussed in Section 3.2, direct search methods are robust. Analytical results now exist to demonstrate that under assumptions comparable to those commonly used to analyze the global behavior of algorithms for solving unconstrained nonlinear optimization problems, direct search methods can be shown to satisfy the first-order necessary conditions for a minimizer (i.e., convergence to a stationary point). This seems remarkable given that direct search methods neither require nor explicitly estimate derivative information; in fact, one obtains these guarantees even when using only ranking information. The fact that most of the direct search methods require a set of directions that span the search space is enough to guarantee that sufficient information about the local behavior of the function exists to safely reduce the step length after the full set of directions has been queried.

Following the lead of Spendley et al. [21], we like to think of direct search methods as “methods of steep descent”. These authors made it quite clear that their algorithm was designed to be related to the method of steepest descent (actually steepest ascent, since the authors were maximizing). Although

no explicit representation of the gradient is formed, enough local information is obtained by sampling to ensure that a downhill direction (though not necessarily the steepest downhill direction) can be identified. Spendley et al. also intuited that steep descent would be needed to ensure what we now call global convergence; furthermore, they recognized the need to switch to higher-order methods to obtain fast local convergence.

This brings us to the second point to be made about the classical direct search methods. They do not enjoy finite termination on quadratic objectives or rapid local convergence. For this, one needs to capture the local curvature of the objective, and this necessarily requires some manner of modeling — hence, the undeniable appeal of modeling-based approaches. However, modeling introduces additional restrictions that may not always be appropriate in the settings in which direct search methods are used: specifically, the need to have explicit numerical function values of sufficient reliability to allow interpolation or some other form of approximation. In truth, the jury is still out on the effectiveness of adding this additional layer of information to devise derivative-free methods that also approximate curvature (second-order) information. Several groups of researchers are currently looking for a derivative-free analog of the elegant trust region globalization techniques for quasi-Newton methods that switch seamlessly between favoring the Cauchy (steepest-descent) direction to ensure global convergence and the Newton direction to ensure fast local convergence.

We close with the observation that, since nonlinear optimization problems come in all forms, there is no “one-size-fits-all” algorithm that can successfully solve all problems. Direct search methods are sometimes used — inappropriately — as the method of first recourse when other optimization techniques would be more suitable. But direct search methods are also used — appropriately — as the methods of last recourse, when other approaches have been tried and failed. Any practical optimizer would be well-advised to include direct search methods among their many tools of the trade. Analysis now confirms what practitioners in many different fields have long recognized: a carefully chosen, carefully implemented direct search method can be an effective tool for solving many nonlinear optimization problems.

References

- [1] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific J. Math.* 16 (1966) 1–3.
- [2] G. Berman, Lattice approximations to the minima of functions of several variables, *J. Assoc. Comput. Mach.* 16 (1969) 286–294.
- [3] G.E.P. Box, Evolutionary operation: a method for increasing industrial productivity, *Appl. Statist.* 6 (1957) 81–101.
- [4] G.E.P. Box, K.B. Wilson, On the experimental attainment of optimum conditions, *J. Roy. Statist. Soc. Ser. B XIII* (1951) 1–45.
- [5] M.J. Box, A comparison of several current optimization methods, and the use of transformations in constrained problems, *Comput. J.* 9 (1966) 67–77.
- [6] M.J. Box, D. Davies, W.H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph, No. 5, Oliver & Boyd, Edinburgh, 1969.
- [7] J. C ea, *Optimisation: Th orie et Algorithmes*, Dunod, Paris, 1971.
- [8] W.C. Davidon, Variable metric method for minimization, *SIAM J. Optim.* 1 (1991) 1–17 (the article was originally published as Argonne National Laboratory Research and Development Report 5990, May 1959 (revised November 1959)).
- [9] R. Fletcher, Function minimization without evaluating derivatives — a review, *Comput. J.* 8 (1965) 33–41.
- [10] R. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization, *Comput. J.* 6 (1963) 163–168.

- [11] A.A. Goldstein, *Constructive Real Analysis*, Harper & Row, New York, 1967.
- [12] R. Hooke, T.A. Jeeves, Direct search solution of numerical and statistical problems, *J. Assoc. Comput. Mach.* 8 (1961) 212–229.
- [13] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM J. Optim.* 9 (1998) 112–147.
- [14] R.M. Lewis, V. Torczon, M.W. Trosset, Why pattern search works, *Optima* (1998) 1–7.
- [15] R.M. Lewis, V.J. Torczon, Rank ordering and positive bases in pattern search algorithms, Technical Report 96-71, Institute for Computer Applications in Science and Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681–2199, 1996.
- [16] K.I.M. McKinnon, Convergence of the Nelder–Mead simplex method to a nonstationary point, *SIAM J. Optim.* 9 (1998) 148–158.
- [17] J.A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.* 7 (1965) 308–313.
- [18] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.
- [19] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Comput. J.* 7 (1964) 155–162.
- [20] H.H. Rosenbrock, An automatic method for finding the greatest or least value of a function, *Comput. J.* 3 (1960) 175–184.
- [21] W. Spendley, G.R. Hext, F.R. Himsforth, Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics* 4 (1962) 441–461.
- [22] W.H. Swann, Report on the development of a new direct search method of optimization, Research Note 64/3, I.C.I. Central Instrument Lab, 1964.
- [23] W.H. Swann, Direct search methods, in: W. Murray (Ed.), *Numerical Methods for Unconstrained Optimization*, Academic Press, New York, 1972, pp. 13–28.
- [24] V. Torczon, Multi-directional search: a direct search algorithm for parallel machines, Ph.D. Thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, 1989.
- [25] V. Torczon, On the convergence of the multidirectional search algorithm, *SIAM J. Optim.* 1 (1991) 123–145.
- [26] V. Torczon, On the convergence of pattern search algorithms, *SIAM J. Optim.* 7 (1997) 1–25.
- [27] M.W. Trosset, I know it when I see it: toward a definition of direct search methods, *SIAG/OPT Views-and-News* 9 (1997) 7–10.
- [28] F.H. Walters, L.R. Parker Jr., S.L. Morgan, S.N. Deming, *Sequential Simplex Optimization*, Chemometrics Series, CRC Press, Boca Raton, FL, 1991.
- [29] P. Wolfe, Convergence conditions for ascent methods, *SIAM Rev.* 11 (1969) 226–235.
- [30] M.H. Wright, Direct search methods: once scorned, now respectable, in: D.F. Griffiths, G.A. Watson (Eds.), *Numerical Analysis 1995*, Addison-Wesley Longman, Reading, MA, 1996, pp. 191–208.
- [31] W.I. Zangwill, Minimizing a function without calculating derivatives, *Comput. J.* 10 (1967) 293–296.