On Descent from Local Minima

By A. A. Goldstein and J. F. Price

Abstract. When a local minimum of a function of several variables has been found by use of an algorithm for finding such minima numerically, one often runs the same algorithm many times with different starting values in the hopes of finding a lower minimum. Here, under the assumption that a local minimum is known, a process with analytical criteria is described which sometimes finds smaller local minima in an algorithmic manner.

Methods of descent are useful for minimizing functions of several variables. Generally, one can always obtain points (if such exist) for which the gradient vanishes, and moreover, points which are local minima. At saddle points one can continue descent with second derivative information. A point which is a local minimum for a function may or may not be a global minimum. At this juncture one resorts to search techniques to attempt to further decrease the function. The process to be described sometimes finds smaller local minima in an algorithmic manner with analytical criteria. One has no general test, of course, for a global minimum.

Consider first the problem of finding the global minimum for a $2n$th degree polynomial $P_1(x)$ in one variable. The coefficient of x^{2n} will be positive. Let x_1 be a local minimizer of P_1. Then one may write

$$P_1(x) = P_1(x_1)$$
$$+ \frac{P_1''(x_1)}{2!}(x - x_1)^2 + \frac{P_1'''(x_1)}{3!}(x - x_1)^3 + \cdots + \frac{P_1^{(2n)}(x_1)}{(2n)!}(x - x_1)^{2n}.$$

If it is further assumed that $P_1''(x_1) \neq 0$, we may define the polynomial

$$P_2(x_1, x) = \frac{2(P_1(x) - P_1(x_1))}{P_1''(x_1)(x - x_1)^2}$$
$$\equiv 1 + \frac{P_1'''(x_1)(x - x_1)}{3P_1''(x_1)} + \cdots + \frac{2P_1^{(2n)}(x_1)(x - x_1)^{2n-2}}{(2n)!\ P_1''(x_1)}.$$

Since $P_1''(x_1) > 0$ and since

$$P_1(x) = P_1(x_1) + \frac{P_1''(x_1)}{2}(x - x_1)^2 P_2(x_1, x),$$

if x_1 is not a global minimum, then for some point $\xi \neq x_1$, $P_2(x_1, \xi) < 0$, so that $P_1(\xi) < P_1(x_1)$. We seek such a point by trying to find a local minimum of $P_2(x_1, x)$. Suppose that at $x = x_2$, $P_2(x_1, x)$ has a local minimum. If $P_2(x_1, x_2) < 0$, x_2 may be used as a new starting value to find a lower minimum of $P_1(x)$. On the other hand, if

$P_2(x_1, x_2) \geqq 0$, the process is extended. Assuming that $P_2''(x_1, x_2) \neq 0$, we define

$$P_3(x_1, x_2, x) = \frac{2(P_2(x_1, x) - P_2(x_1, x_2))}{P_2''(x_1, x_2)(x - x_2)^2}$$

$$\equiv 1 + \frac{P_2'''(x_1, x_2)(x - x_2)}{3 P_2''(x_1, x_2)} + \cdots + \frac{2 P_2^{(2n-2)}(x, x_2)(x - x_2)^{2n-4}}{(2n - 2)! \, P_2''(x_1, x_2)} ,$$

and seek a negative value for this function. Assume that at $x = x_3$, $P_3(x_1, x_2, x)$ has a minimum. If $P_3(x_1, x_2, x_3) < 0$, we have a new starting value for finding a lower minimum for $P_2(x_1, x)$. If $P_3(x_1, x_2, x_3) \geqq 0$ (and assuming again that $P_3''(x_1, x_2, x_3) \neq 0$), we define $P_4(x_1, x_2, x_3, x)$ in an analogous manner and continue the process.

Since

$$(\text{degree of } P_k) = (\text{degree of } P_{k-1}) - 2,$$

only a finite number of polynomials are involved and P_{n+1} is a constant. Since P_{n+1} cannot be decreased, it follows that P_n cannot be decreased; hence P_{n-1} cannot be decreased, and finally it is seen that P_1 cannot be decreased. Hence x_1 is a global minimum.

On the other hand, if, at $x = x_1$, $P_1(x)$ was a local (but not a global) minimum, then since all P_k's, $k = 2, 3 \cdots, n$, could have at most a finite number of minima, it follows that the process described would yield a point ξ (inducing a lower value of P_1) in a finite number of steps. Clearly, the process terminates in a global minimum for $P_1(x)$.

As a simple example, consider the function

$$F_1(x) = x^6 - 15x^4 + 27x^2 + 250$$

with minima $(0, 250)$, $(\pm 3, 7)$. If one finds $x_1 = 0$ as a local minimizer, then $F_1(x)$ is written

$$F_1(x) = 250 + x^2(x^4 - 15x^2 + 27)$$

and the minima of $F_2(0, x)$ are both negative. The algorithm would then yield one of the other minima of $F_1(x)$. If one finds $x_1 = 3$ as a local minimizer of $F_1(x)$, then $F_1(x)$ is written

$$F_1(x) = 7 + (x - 3)^2 \{ x^4 + 6x^3 + 12x^2 + 18x + 27 \}.$$

The only minimum of $F_2(3, x)$ is at $x = -3$, and $F_2(3, -3) = 0$. Thus

$$F_1(x) = 7 + (x - 3)^2 \{ 0 + (x + 3)^2 (3 + x^2) \}.$$

It is now seen that $F_3(3, -3, x)$ has its only minimum at $x = 0$. Since this minimum is positive and since $F_4(3, -3, 0, x)$ is a constant, it follows that $F_1(x)$ cannot be decreased further.

If one had started with $x_1 = -3$, the results would have been essentially the same. The final decomposition of $F_1(x)$ would have been

$$F_1(x) = 7 + (x + 3)^2 \{ 0 + (x - 3)^2 (3 + x^2) \}.$$

We consider now an analogous process for the case of analytic functions of several variables. Conditions for the success of this algorithm are unknown to us. However, we include some examples of the application of the process.

Given an analytic function F_1 of n variables with local minimum at x_1, let $H_1(x_1)$ be the Hessian of $F_1(x)$ at x_1. Assume $H_1(x_1)$ is nonsingular. We define $F_2(x_1, x)$ by the equation:

$$F_1(x) = F_1(x_1) + \frac{[(x - x_1), H_1(x_1)(x - x_1)]}{2!} F_2(x_1, x),$$

where $[\cdot, \cdot]$ is the inner product in E_n. If it is possible to find a value of ξ such that $F_2(x_1, \xi) < 0$, this value of x will be a good starting value for finding a lower minimum of F_1. The new problem is then to minimize $F_2(x_1, x)$. If the process continues, we may write:

(1)
$$\begin{aligned}
F_{k-1}&(x_1, x_2, \cdots, x_{k-2}, x) \\
&= F_{k-1}(x_1, \cdots, x_{k-2}, x_{k-1}) \\
&\quad + \frac{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}{2!} F_k(x_1, \cdots, x_{k-1}, x).
\end{aligned}$$

We conjecture that if the function F_1 is analytic, and if the functions F_1, F_2, F_3, etc., have a finite number of stationary points all of which lie in a bounded set, then we can find a point lower than any given purely local minimum in a finite number of steps. On the other hand, starting at a global minimum x the algorithm will generally not terminate, except for the case of a polynomial in one variable. On the other hand, if some auxiliary function F_k is nonnegative everywhere, then x is a global minimum.

The elements of the Hessian matrix H_k may be calculated by use of the gradients and Hessians of F_{k-1}. By differentiation of the above formula, one can obtain the result

$$\nabla F_k(x_1, \cdots, x_{k-1}, x)$$
$$= \frac{2\{\nabla F_{k-1}(x_1, \cdots, x_{k-2}, x) - F_k(x_1, \cdots, x_{k-1}, x)H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})\}}{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}.$$

Differentiation of Eq. (1) a second time leads to the result

$$H_k(x_1, \cdots, x_{k-1}, x)$$
$$= \frac{2\{H_{k-1}(x_1, \cdots, x_{k-2}, x) - \nabla F_k(x_1, \cdots, x_{k-1}, x)\{H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})\}^T\}}{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}$$
$$- \frac{2\{H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})\{\nabla F_k(x_1, \cdots, x_{k-1}, x)\}^T\}}{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}$$
$$- \frac{2F_k(x_1, \cdots, x_{k-1}, x)H_{k-1}(x_1, \cdots, x_{k-1})}{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}.$$

In particular, since x_k is defined as a minimum of $F_k(x_1, \cdots, x_{k-1}, x)$, it follows that

$$H_k(x_1, \cdots, x_k) = \frac{2\{H_{k-1}(x_1, \cdots, x_{k-2}, x_k) - F_k(x_1, \cdots, x_k)H_{k-1}(x_1, \cdots, x_{k-1})\}}{[(x - x_{k-1}), H_{k-1}(x_1, \cdots, x_{k-1})(x - x_{k-1})]}.$$

We have tried the process on several test problems. As a practical limit, our computer routines were such that they stopped rather than consider F_5; this, of

course, restricts the success of the process. The basic local minimization algorithm used was that of the reference.

First, we consider an 8th degree polynomial in two variables, namely:

$$F_1 = \{1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)\}$$
$$\cdot \{30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)\}$$

which has the local minimum points (1.2, 0.8, 840), (1.8, 0.2, 84), (−0.6, −0.4, 30), and (0, −1, 3). When the starting value was chosen near the second named minimum, the process gave the following minima for the functions as defined in the algorithm:

$$F_1 = 84.00000 \quad \text{at} \quad (1.80000, 0.20000)$$

$$F_2 = 0.07407 \quad \text{at} \quad (1.13333, 0.86667)$$

$$F_3 = -0.18729 \quad \text{at} \quad (-0.44558, -0.20925)$$

$$F_2 = -0.18519 \quad \text{at} \quad (0.20000, -0.86667)$$

$$F_1 = 3.00000 \quad \text{at} \quad (0.00000, -1.00000) \quad \text{(Global minimum)}$$

$$F_2 = 0.07407 \quad \text{at} \quad (2.00000, 0.33333)$$

$$F_3 = 0.05334 \quad \text{at} \quad (1.27987, 0.98687).$$

Our local minimization routine then failed to find a true minimum for F_4, and thus the program concluded that the point (0, −1, 3) was possibly a global minimum. Near the line $x + y + 1 = 0$, F_4 and the components of ∇F_4 approach zero as x and y approach infinity, and the minimization algorithm headed out along this trough. Ordinarily, one might expect to have to go to F_6 before finding such behavior at ∞ because F_1 is of 8th degree. However, along the line $x + y + 1 = 0$ (or along the line $2x - 3y = 0$), F_1 is only of 4th degree.

When a starting value was chosen near the largest local minimum of F_1, the process gave the following minima for the functions defined by the algorithm:

$$F_1 = 840.00000 \quad \text{at} \quad (1.20000, 0.80000)$$

$$F_2 = -0.18519 \quad \text{at} \quad (-0.40000, -0.26667)$$

$$F_1 = 30.00000 \quad \text{at} \quad (-0.60000, -0.40000)$$

$$F_2 = 0.07407 \quad \text{at} \quad (1.40000, 0.93333)$$

$$F_3 = -0.18611 \quad \text{at} \quad (1.92089, 0.39312)$$

$$F_2 = 0.00503 \quad \text{at} \quad (1.94567, 0.29703)$$

$$F_3 = 0.07440 \quad \text{at} \quad (1.33389, 1.00067).$$

Our local minimization routine again failed to find a true minimum for F_4. When it stopped looking, it had been going very slowly out towards infinity along a trough near the line $2x - 3y = 0$.

Consider now the function of two variables

$$F_1 = e^{u^2} + \sin^4(4x - 3y) + \tfrac{1}{2}(2x + y - 10)^2$$

where

$$u = \tfrac{1}{2}(x^2 + y^2 - 25).$$

This function has many local minima. Its global minimum of 1 is attained at the point (3, 4).

With a starting guess of (4.3, 2.5), the process gave the following minima for the functions as defined in the algorithm:

$F_1 = 1.59610$ at (4.23366, 2.57739)

$F_2 = 0.31912$ at (4.28207, 2.49359)

$F_3 = -0.01400$ at (4.52803, 2.02104)

$F_2 = 0.00115$ at (4.59177, 1.85735)

$F_3 = -0.03217$ at (4.79969, 1.26915)

$F_2 = -0.01801$ at (4.83117, 1.14328)

$F_1 = 1.35089$ at (4.83509, 1.10764)

$F_2 = 0.95340$ at (4.89062, 1.16513)

$F_3 = -0.17817$ at (4.96210, 0.64176)

$F_2 = -0.08634$ at (4.96255, 0.44300)

$F_1 = 1.03481$ at (4.97860, 0.26959)

$F_2 = 0.38552$ at (4.96093, 0.41790)

$F_3 = -0.02164$ at (4.85379, 1.04191)

$F_2 = 0.07052$ at (4.79111, 1.25705)

$F_3 = -0.00950$ at (3.99411, 2.98862)

$F_2 = 0.00417$ at (3.84188, 3.14352)

$F_3 = 0.97058$ at (3.86674, 3.19619)

$F_4 = -0.13176$ at (3.65358, 3.44897)

$F_3 = -0.07390$ at (3.52206, 3.51881)

$F_2 = 0.00064$ at (3.42575, 3.61546)

$F_3 = -0.03809$ at (3.04941, 3.96126)

$F_2 = -0.00017$ at (3.01190, 3.99057)

$F_1 = 1.00000$ at (3.00000, 4.00000) (Global minimum)

$F_2 = 0.99895$ at (2.99059, 3.99678)

$F_3 = 0.40716$ at (3.00446, 4.13102)

$F_4 = -4.37602$ at (3.00048, 3.99221)

$F_3 = -0.02424$ at (2.99984, 3.99096)

$$F_2 = 0.99884 \quad \text{at} \quad (2.99981, 3.99024)$$

$$F_3 = 0.29377 \quad \text{at} \quad (3.14976, 4.04084)$$

$$F_4 = -0.39210 \quad \text{at} \quad (2.63994, 3.99252)$$

$$F_3 = -.00500 \quad \text{at} \quad (1.92550, 4.52410).$$

At this point, our program suggested that $(3, 4, 1)$ might be a global minimum because the local minimization routine failed to find a local minimum for F_2.

When one looks at the minima of the various auxiliary functions as the algorithm proceeds, it is noted that in many cases the process could have been speeded up by making the following change in the algorithm: Whenever a local minimum point is found for any function, the coordinates of x are saved in a list. Then, when one wishes to find a local minimum of F_k, a starting value for x is chosen so that F_k has its smallest value obtainable for any point in the list. We have not tried this modified algorithm on any test problems. If difficulty occurs because of stationary points at infinity, the addition of a penalty function would be a possibility. For example, if the domain under consideration is a sphere of radius ρ with center x_0, we could add the function $[||x - x_0||^2/\rho^2]^q$ to F_1, where q is a large integer. For large q, the minima of the sum function will be close to the minima of F_1, and the stationary points of F_k when k is not too large will lie in the sphere.

University of Washington
Seattle, Washington 98105

Boeing Scientific Research Laboratories
Seattle, Washington 98105

1. A. A. GOLDSTEIN & J. F. PRICE, "An effective algorithm for minimization," *Numer. Math.*, v. 10, 1967, pp. 184–189. MR 36 #1084.