ACCELERATIONS FOR A VARIETY OF GLOBAL OPTIMIZATION METHODS

by

W.P. Baritompa

University of Canterbury, Christchurch, New Zealand

No. 81   March, 1993

Abstract – Optimization methods for a given class are easily modified to utilize additional information and work faster on a more restricted class. In particular algorithms that use only the Lipschitz constant (e.g. Mladineo, Piyavskii, Shubert and Wood) can be modified to use second derivative bounds or gradient calculations. The algorithm of Breiman & Cutler can be modified to use Lipschitz bounds. Test cases illustrating accelerations to various algorithms are provided.

Keywords – Multidimensional bisection, deterministic, global optimization, mathematical programming, search

Accelerations for a Variety of Global Optimization Methods

W. Baritompa

Department of Mathematics
University of Canterbury
Christchurch, New Zealand

Optimization methods for a given class are easily modified to utilize additional information and work faster on a more restricted class. In particular algorithms that use only the Lipschitz constant (e.g. Mladineo, Piyavskii, Shubert and Wood) can be modified to use second derivative bounds or gradient calculations. The algorithm of Breiman & Cutler can be modified to use Lipschitz bounds. Test cases illustrating accelerations to various algorithms are provided.

1. Preliminaries

Introduction

As discussed in [1], algorithms can be customized for special classes of functions. This implies the simple expediency of raising function values appropriately, accelerate many existing implementations such as those of Breiman & Cutler, Mladineo, Piyavskii, Shubert and Wood. These methods can be easily modified to use more information and work faster on a more restricted class. This section continues presenting some basic terminology and includes as background a general description of the basic algorithms. Section 2 gives the details of the accelerations. Section 3 looks at some comparison tests.

Notation and basic problem

The basic problem is to find the global minimum α and its location $E = f^{-1}(\alpha) \cap K$ of a function $f : K \to \mathsf{R}$ where $K \subset \mathsf{R}^n$. K is usually the closure of a bounded open set and often a convex body.

An $M-cone$ is any translate of $\{(x,y)|y \le -M\|x\|\}$ in R^{n+1}. A $B-paraboloid$ is any translate of $\left\{(x,y)\middle|y \le -\frac{1}{2}B\|x\|^2\right\}$. An $MB-parabolically\ capped\ cone$ is any translate of the union of the $B-$paraboloid and $\left\{(x,y)\middle|y \le \min\left\{-\frac{M^2}{2B}, -M\|x\| + \frac{M^2}{2B}\right\}\right\}$, the part below the level of tangency of the circumscribed $M-$cone.

Let $L(M)$ be the class of Lipschitz continuous functions with constant M. Let ZG be the set of all differentiable functions with global minimum having zero gradient. Let $C_u^2(B)$ be the class of all twice differentiable functions such that $h(x_0 + \Delta x) = f(x_0) + \nabla f(x_0)\Delta x + \frac{1}{2}B\|\Delta x\|^2$ is an upper bound at each point of the domain x_0. Similarly let $C_l^2(B)$ have $h(x_0 + \Delta x) = f(x_0) + \nabla f(x_0)\Delta x - \frac{1}{2}B\|\Delta x\|^2$ as a lower bound at each point of the domain. For a given function the best bounds of the above, respectively, are the maximum and negative of the minimum of the eigenvalues of the Hessian.

The algorithms of Mladineo, Wood, etc. work on $L(M)$. Breiman & Cutler (used to minimize) deal with $C_l^2(B)$.

Background Description of Algorithms

For the reader not familiar with the details of the algorithms of Breiman & Cutler, Mladineo, Piyavskii, Shubert and Wood, the following general description due to Piyavskii [4] is useful.

- *Initialization:*

  $\alpha_{-1} = \infty$

  $i = -1$

  Take a user specified x_0 from the domain K
- *Evaluation Step:*

  Increment i

  Compute $f(x_i)$

  $g_i = \nabla f(x_i)$ (if required)

  $\alpha_i = \min\{\alpha_{i-1}, f(x_i)\}$
- *Update Envelope Function Step:*

  $F_i(x) = \max\limits_{k=0,\ldots,i} h_k(x)$ where $h_k(x)$ depends on $(x_k, f(x_k))$ and perhaps the gradient vector g_i.
- *Get Next Sample Point Step:*

  $x_{i+1} = \arg \min\limits_{x \in K} F_i(x)$
- *Termination Test:*

  If $\min\limits_{x \in K} F_i(x)$ is close to α_i stop, otherwise go back to the evaluation step.

The functions $h_k(x)$ in the above description determine the specific algorithms: Breiman & Cutler use $h_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) - \frac{1}{2}B\|x - x_k\|^2$; Mladineo, Piyavskii and Shubert use $h_k(x) = f(x_k) - M\|x - x_k\|$; and Wood uses $h_k(x) = f(x_k) + \min\limits_{i=0,\ldots,n} u_i \cdot (x - x_k)$ where n is the dimension of the domain and the vectors u_i are the vertices of the standard n–simplex in $\mathbb{R}^n$.

All these satisfy the two conditions required by Piyavskii, namely $h_k(x) \leq f(x)$ for all x in the domain and $h_k(x_k) = f(x_k)$, and thus the functions $F_i(x)$ are "lower envelopes."

So for these algorithms the global minimum is always between lowest value of the envelope, $\min_{x \in K} F_i(x)$, and the lowest known function evaluation, α_i.

Taking a geometric viewpoint, the set of points above or on the graph of $F_i(x)$ and below or on the hyperplane at height α_i form a bracket of the point(s) on the graph of f corresponding to the global minimum. These nest down as the algorithm proceeds. As pointed out in [1], even if the conditions on $h_k(x)$ are not satified, the resultant algorithms may still produce brackets of the global minimum.

2. Acceleration Of Existing Methods

The geometric viewpoint developed in [1] is the key behind the acceleration ideas presented in this paper. The viewpoint is that the bracket found by the algorithm occurs by removal of certain regions at each step. The special cases of removing paraboloids and cones are the implementations by Breiman & Cutler, Mladineo, Piyavskii, Shubert and Wood as outlined in the previous section. However removing complicated geometric shapes from a bracket (i.e. finding the global minimum of an complex envelope) is more difficult in practice.

An algorithm designed to work with certain removal regions can be improved to work on a class of functions where "better" uniformly bigger removal regions are appropriate. Often the better removal region contains one of the algorithm's removal regions at a higher point, so the later can be removed to achieve an acceleration. This idea was first used by Wood [7 pg. 166–168] for "spherical" acceleration of multidimensional bisection. He observed that simplicial cones are strictly inside better spherical cones and hence sometimes can be raised.

Parabolically Capped Cone Acceleration

All the accelerations discussed in this section concern the way the next point is used by the algorithm. The notation presented in the background description is used. In particular the sample sequence of points where the function has been evaluated is denoted (x_i), and the lowest known height is $\alpha_i = \min_i f(x_i)$. During the *Update Envelope Function* step, the function $h_i(x)$ depends on the evaluated point $(x_i, f(x_i))$ and possibly the gradient g_i (i.e. $h_i(x) = h(x; x_i, f(x_i), g_i)$). If the evaluated point is sufficiently high, the following propositions show an even higher replacement point $(x_i^a, f^a(x_i))$ can be used. Letting $h_i(x) = h(x; x_i^a, f^a(x_i), g_i)$ will produce faster convergence and the extra computation is minimal and should not affect the overheads of the algorithm.

The following propositions consider situations where the MB−parabolically capped cone can be cut away. Since M−cones, simplicial M−cones and B−paraboloids fit inside, any methods that use these as cutters can be accelerated. All proofs refer to the following

two elementary lemmas which express basic geometric facts relating M-cones and B-paraboloids. It suffices only to examine two-dimensional cross sections to verify them.

Lemma 1 *Given an intersecting M-cone and B-paraboloid (with common vertical axis). Let r be the radius of their sphere of intersection, d_p and d_c be the distance from the apex of the paraboloid and cone to the hyperplane containing the sphere. The following holds:* $r = \dfrac{d_c}{M} = \dfrac{\sqrt{2d_p}}{\sqrt{B}}.$

Lemma 2 *Consider the circumscribed M-cone and inscribed B-paraboloid to a MB-parabolically capped cone. Using the notation of the previous lemma:* $r = \dfrac{M}{B}$, $d_c = \dfrac{M^2}{B}$ *and* $d_p = \dfrac{M^2}{2B}.$

Figure 1 M-cone and B-paraboloid (cross section)

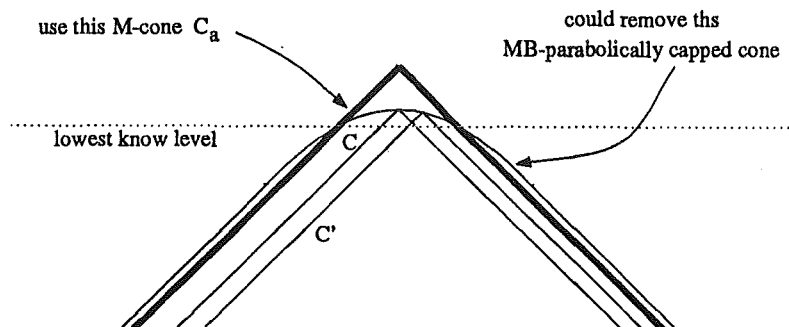The next two results refer to raising an M-cone and refer to the following figure.

Figure 2 Raising an M-cone (cross section)

Proposition 3 *Let f be a function in the class $L(M) \cap C_u(B) \cap ZG$. When using the algorithm of Mladineo or Wood, an acceleration is obtained by using the replacement sample point $(x_i^a, f^a(x_i))$ during the Update Envelope Function Step. Its coordinates are given by $x_i^a = x_i$ and $f^a(x_i) = \begin{cases} \alpha_i + \frac{M}{\sqrt{B}}\sqrt{2d_i} & d_i < \frac{M^2}{2B} \\ f(x_i) + \frac{M^2}{2B} & d_i \geq \frac{M^2}{2B} \end{cases}$ where $d_i = f(x_i) - \alpha_i$.*

Proof: Let C_a be the M-cone intersecting the hyperplane at height α_i in the same sphere as the MB-parabolically capped cone with apex at $(x_i, f(x_i))$ does. The part of C_a below α_i is contained in the parabolically capped cone which can be removed [1]. Since $C_a \supseteq C$, the M-cone used by the algorithm, C_a can be

4

used to produce a bigger reduction in the bracket. Now $d_i = f(x_i) - \alpha_i$ is the distance from the apex of the $B-$paraboloid to the hyperplane. When $d_i < \frac{M^2}{2B}$, Lemma 1 gives the top of C_a above α_i. In the other case, lemma 2 gives the top of C_a above the top of the paraboloid. ∎

If the next point produces a "record" value (i.e. has the lowest value so far), it is used in the usual way. If the next point's value is above the lowest known value, the replacement point is higher. The amount the accelerated value exceeds the unaccelerated value depends on the drop from the current point to the lowest known value. Once the drop exceeds a certain amount, the accelerated height is a fixed distance above the evaluated height.

Proposition 4 *Given a function f in the class $C_f^2(B) \cap L(M)$. When using the algorithm of Mladineo, an acceleration using gradient information is obtained by using the replacement sample point $(x_i^a, f^a(x_i))$ during the Update Envelope Function Step. Its coordinates are given by $x_i^a = x_i + \frac{1}{B}\nabla f(x_i)$ and*

$$f^a(x_i) = \begin{cases} \alpha_i + \frac{M}{\sqrt{B}}\sqrt{2d_i} & d_i < \frac{M^2}{2B} \\ f(x_i) + \frac{M^2 + \|\nabla f(x_i)\|^2}{2B} & d_i \geq \frac{M^2}{2B} \end{cases} \text{ where } d_i = f(x_i) - \alpha_i + \frac{\|\nabla f(x_i)\|^2}{2B}.$$

Proof: The $MB-$parabolically capped cone that is tangent to the graph of f at $(x_i, f(x_i))$ can be removed. Its apex is $\left(x_i + \frac{1}{B}\nabla f(x_i), f(x_i) + \frac{\|\nabla f(x_i)\|^2}{2B}\right)$. Hence the $M-$cone C with apex at this point could be used. Raising C to C_a as in the previous proof gives $f^a(x_i)$. Since C_a contains the apex of C', the cone used by the algorithm, then $C_a \supseteq C'$ and using C_a gives a bigger bracket reduction. ∎

Here the accelerated value is always greater than the function evaluation unless it is a record with zero gradient.

Note the above result holds only for Mladineo's algorithm which uses $M-$cones. With simplicial cones used by Wood's method, $C_a \supseteq C'$ often does not hold. To guarantee an acceleration, Wood's algorithm needs to be modified to remove both C_a and C'. It is worth noting that spherical acceleration discussed by Wood is compatible with the above two accelerations for $M-$cones. It can be implemented using the accelerated point as the function evaluation.

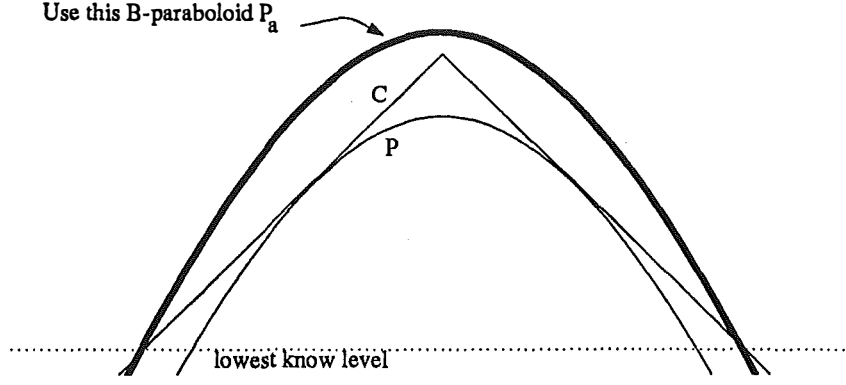The next two results concern raising paraboloids and refer to this figure.

Figure 3 Raising a B-paraboloid (cross section)

As mentioned in [1] Breiman & Cutler's method could be used to remove B-paraboloids by artificially taking the gradient to be zero. If a Lipschitz bound is available an acceleration is possible.

Proposition 5 *Given a function f in the class $L(M) \cap C_u(B) \cap ZG$. When using the method of Breiman & Cutler to remove B-paraboloids, an acceleration is obtained by using the replacement sample point $(x_i^a, f^a(x_i))$ during the Update Envelope Function Step. Its coordinates are given by $x_i^a = x_i$ and*

$$f^a(x_i) = \begin{cases} f(x_i) + \frac{B}{2M^2}(d_i - l)^2 & d_i > l \\ f(x_i) & d_i \le l \end{cases} \text{ where } d_i = f(x_i) - \alpha_i \text{ and } l = \frac{M^2}{2B}.$$

Proof: The B-paraboloid P with apex at $(x_i, f(x_i))$ is used by the algorithm. If $d_i = f(x_i) - \alpha_i > \frac{M^2}{2B}$, the paraboloid P_a can be used since the part below α_i is in the M-cone C which can be removed. Its apex is at $d_i + \frac{M^2}{2B}$ by lemma 2. Lemma 1 provides the height of P_a above C. Since $P_a \supset P$, its use provides an acceleration. ∎

Breiman & Cutler's method can be modified to use a Lipschitz bound.

Proposition 6 *Given a function f in the class $C_l^2(B) \cap L(M)$. When using the algorithm of Breiman & Cutler, an acceleration is obtained by using the replacement sample point $(x_i^a, f^a(x_i))$ during the Update Envelope Function Step. Its coordinates are given by $x_i^a = x_i$ and $f^a(x_i) = \begin{cases} f(x_i) + \frac{B}{2M^2}(d_i - l_i)^2 & d_i > l_i \\ f(x_i) & d_i \le l_i \end{cases}$ where $d_i = f(x_i) - \alpha_i$ and $l_i = \frac{M^2 - \|\nabla f(x_i)\|^2}{2B}$.*

Proof: The paraboloid used by the algorithm is tangent to the graph of f at $(x_i, f(x_i))$. Its apex is $\left(x_i + \frac{1}{B}\nabla f(x_i), f(x_i) + \frac{\|\nabla f(x_i)\|^2}{2B}\right)$. The amount this can be raised follows from the previous argument using $f(x_i) + \frac{\|\nabla f(x_i)\|^2}{2B}$ in place of $f(x_i)$. ∎

6

3. Examples

Testing

A discrete testing program discussed in [1] is used to compare different methods. A grid of 101×101 points was used. Although machine accuracy is attained when evaluating function and envelope values at grid points, the algorithms simulated are not exactly deepest point but deepest **grid** point methods. Similarly the difference between lowest known value and envelope minimum is only on the grid. For this reason the tests were stopped when the bracket consisted of exactly the global minimum or an upper limit on evaluations was reached. The main use here is to see the effect of the various accelerations. It is noted however the results in column two of Table 2 are comparable with the results reported in Breiman & Cutler[2].

Tests on five functions are summarized in the tables below. The domains and starting points are as in Breiman & Cutler[2] (except for one test on the camelback function). Bounds were taken to be the best seen on grid. Tables 1 and 2 give the number of iterations until stopping. For those not having stopped by 750 iterations, some indication of the estimated total number (based on extrapolating the error curves produced by the program) is given.

Test	Mladineo	Mladineo + prop 3	Parabolic cutters	Parabolic cutters + prop 5	MB-parabolic cones
EXP2	267	12	10	8	8
COS2	238	64	57	54	54
RCOS	>> 750	212	193	193	192
GW	> 750	474	701	443	441
C6	>>>> 750	>>> 750	>> 750	>> 750	>> 750

Table 1 Algorithms not using the gradient (iteration = function evaluation)

Test	Mladineo + prop 4	Breiman & Cutler	B&C + prop 6	Tangent MB parabolic cones
EXP2	19	27	25	22
COS2	61	70	55	56
RCOS	176	176	174	176
GW	434	705	458	435
C6	52	92	92	92
C6 [start (-5,-5)]	55	58	58	58

Table 2 Algorithms using the gradient (iteration = function and gradient evaluation)

Test	Domain	Initial Point	Lipshitz constant M	B_u (B-par)	B_l (with ∇f)
EXP2	$(-1,1) \times (-1,1)$	$(0.2, 0.2)$	0.61	1	0.37
COS2	$(-1,1) \times (-1,1)$	$(0.5, 0.5)$	4.8	26.7	22.7
RCOS	$(-5,10) \times (0,15)$	$(0,5)$	113.6	29.2	16.8
GW	$(-100,100) \times (-100,100)$	$(25,25)$	2.15	1.01	0.99
C6	$(-5,5) \times (-5,5)$	$(0,0)$	5601	5628	8.93

Table 3 Bounds used

Comments

All these accelerations rely on the fact that $MB-$parabolically capped cones could have been removed. So the algorithm that removes them should bound the possible improvement these accelerations give. Examining the right-most column of the tables shows this is (almost) correct. The main exception is the acceleration of Mladineo's algorithm using both second derivative bound and gradient calculation on the six-hump camel back function (Table 2, fifth row, first column). For the same sample sequence of points, removing parabolically capped cones will always produce a smaller bracket than using the accelerated cones. However in this case the deepest point sequences are quite different. The accelerated $M-$cone algorithm has points going interior to the domain quite early on. When another starting point is used this effect disappears.

Methods using only the $M-$cone produce lower bounding functions which are very poor local approximations to the graph (if the minimum is smooth). This is quite apparent in column one of Table 1. Of course an algorithm that works on the whole class of Lipschitz functions must allow for quite jagged possibilities. Utilizing second derivative bounds and gradient information via propositions 3 and 4 make up for this deficiency. Note the accelerations work nearly as well as the parabolically capped cone removing algorithms shown in the right-most columns. Propositions 3 and 4 provide useful improvements even if the drop in function value is very small. In fact compared to the magnitude of the drop, the acceleration goes to infinity as the drop gets small, hence in the later stages of these algorithms the accelerations are affecting each iteration.

In contrast, methods using paraboloids have envelopes which are good local approximations near evaluated points, but move quickly away from them. The accelerations of propositions 5 and 6 take effect only if there is a large drop in value. These help only in the early stages of an algorithm. For RCOS this minimal drop is nearly the overall distance from minimum to maximum, so acceleration hardly occurred. For C6, the minimal drop is so large it never occurred, in fact the paraboloid part of the parabolically capped

cone is so big that columns 2 to 4 produced identical output. For GW the minimal drop is quite small and the improvement is quite marked.

Matlab

The discrete testing program was written for the **matlab** package. It is available upon request (e-mail wpb@math.canterbury.ac.nz).

References

[1] W. Baritompa, Customizing methods for global optimization — a geometric viewpoint, *J. Global Optimization* To appear

[2] Breiman & Cutler(1989), A Deterministic Algorithm for Global Optimization, Technical Report #224, Dept. of Stats., Univ. of California, Berkeley

[3] R. H. Mladineo (1986), An algorithm for finding the global maximum of a multimodal, multivariate function, *Mathematical Programming* 34 188–200.

[4] S.A. Piyavskii (1972), An algorithm for finding the absolute extremum of a function, *USSR Comp. Math. and Math. Phys.* **12**, 57–67

[5] Bruno O. Shubert (1972), A sequential method seeking the global maximum of a function, *SIAM J. Numer. Anal.* **9**, 379–388

[6] G. R. Wood (1992), The bisection method in higher dimensions, *Math. Program.* **55** 319–337

[7] G. R. Wood (1991), Multidimensional bisection and global optimization, *Computers and Math. Applic.* **21** 161–172