# OUTER APPROXIMATION ALGORITHM FOR NON-DIFFERENTIABLE OPTIMIZATION PROBLEMS

by

D. Q. Mayne and E. Polak

Memorandum No. UCB/ERL M83/40

12 July 1983

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# OUTER APPROXIMATION ALGORITHM FOR NON-DIFFERENTIABLE OPTIMIZATION PROBLEMS[1]

D Q Mayne[2] and E Polak[3]

ABSTRACT

It is known that the problem of minimizing a convex function $f(x)$ over a compact subset $X$ of $\mathbb{R}^n$ can be expressed as minimizing $\max \{g(x, y) \mid y \in X\}$ where $g$ is a "support" function for $f$ ($f(x) \geq g(x, y)$ for all $y \in X$ and $f(x) = g(x, x)$). Standard outer approximation theory can then be employed to obtain outer approximation algorithms with procedures for dropping previous cuts. It is shown here how this methodology can be extended to non-convex non-differentiable functions.

Publication no: EE.CON.82.14
Department of Electrical Engineering
Imperial College
London SW7 2BT

---

1. Research supported by Science and Engineering Research Council, UK and the National Science Foundation under Grant No. ECS-79-13148.

2. D Q Mayne is with the Department of Electrical Engineering, Imperial College of Science and Technology, London SW7 2BT, England.

3. E Polak is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720, USA

Original June/82 Revision 1: 9/8/82

## 1. INTRODUCTION

Consider the problem of minimizing $f(x)$ over a compact subset $X$ of $\mathbb{R}^n$. If $f$ is non-differentiable but locally Lipschitz continuous, a necessary condition of optimality is that $0 \in \partial f(x)$, where $\partial f$ is the generalized gradient of $f$. The point-to-set map $x \mapsto \partial f(x)$ is upper-semi-continuous but the obvious extension of the classical steepest descent algorithm, using $s(x) = -g(x)$, $g(x) \triangleq \arg\min \{\|g\| \mid g \in \partial f(x)\}$, as the search direction ($g(x) = \nabla f(x)$ if $f$ is differentiable at $x$), does not work since local uniform upper-semi-continuity is required for convergence. In order to achieve convergence, using analogs of conventional algorithms, $\partial f(x)$ in the above expression for $g(x)$ is usually replaced by a suitably chosen set $G$ which includes $\partial f(x)$ as a subset. Thus if $f$ is convex $G$ is constructed as a bundle of current and previous subgradients [1]. If $f$ is merely locally Lipschitz continuous $\partial f$ is replaced by $\partial_\varepsilon f(x)$, a bundle of generalized gradients obtained by exploring completely an $\varepsilon$-neighbourhood of $x$, together with a procedure for reducing $\varepsilon$ to zero [2,3,4]. This is computationally prohibitive so implementable algorithms require that $f$ be semi-smooth [5,6] in which case a suitable approximation to $\partial_\varepsilon f$ can be obtained by exploration of a finite number of points on a finite set of search directions. Hence, from a practical point of view, current algorithms for non-differentiable optimization are restricted to the case where $f$ is convex or $f$ is semi-smooth. It is the main purpose of this paper to extend the range of problems for which implementable algorithms are available. Specifically, we present in this paper an algorithm for minimizing $f(x)$ over $X$ when $f$ is globally Lipschitz continuous (but not necessarily convex or semi-smooth).

The approach adopted in this paper can best be appreciated by considering the case when f is convex. In this case, existing algorithms exploit the fact that, by duality, the minimization problem can be expressed as a min-max problem, which requires, at each x, the solution of $\max \{f(x) + \langle g, x - y\rangle | y \in X, g \in \partial f(y)\}$ where $\partial f$ is the subgradient of f. The cutting plane or outer approximation [7,8] algorithms replace X at iteration i by $X_i$ which is equal to or is a subset of the set $\{x_0, x_1, \ldots, x_{i-1}\}$ of previously generated points. Constraint dropping schemes, such as those described in [9], can be employed to reduce $X_i$ to a small subset of $\{x_0, x_1, \ldots, x_{i-1}\}$ thus substantially reducing the complexity of the maximization stage, without destroying the convergence properties of the algorithm.

In this paper we exploit an extension [10] of the concept of duality to non-convex functions. If we assume that f is globally Lipschitz continuous with Lipschitz constant K, then there exists a "support" function $g_k$, parametrized by k, such that f(x) is identical to $\max \{g_k(x, y) | y \in X\}$, provided that $k \geq K$. Hence the minimization problem may again be expressed as a min-max problem which can be solved using a standard outer approximation algorithm [8] with constraint dropping [9].

The paper is organized as follows. In Section 2 we define a suitable "support" function $g_k$ so that the minimization problem may be redefined as a min-max problem and present an outer-approximation algorithm which solves the global minimization problem. In Section 3 we show how previous elements of $X_i$ may be discarded (thus reducing the complexity of the maximization stage) without damaging the convergence properties of the algorithm. In Section 4 we show how the

algorithm may be modified to compute a local rather than a global minimum, thus reducing computational effort. In all the above we assume that the global Lipschitz constant K is known. In Section 5 we consider briefly the problem of selecting a suitable approximation to K when it is not known.

## 2. AN OUTER APPROXIMATION ALGORITHM

We consider initially the problem

$$P : \min \{f(x) \mid x \in X\} \tag{2.1}$$

where X is a compact subset of $\mathbb{R}^n$ and $f : X \to \mathbb{R}$ is assumed to be globally Lipschitz continuous with Lipschitz constant K, so that

$$|f(x) - f(y)| \leq K\| x - y\|_\infty \tag{2.2}$$

for all $x, y \in X$. For all $k \in \mathbb{R}^+$ let $g_k : X \times X \to \mathbb{R}$ be defined by:

$$g_k(x, y) \triangleq f(y) - k\| x - y\|_\infty . \tag{2.3}$$

It follows from (2.3) that $g_k(x, y) \leq f(y)$ and that $g_k(x, x) = f(x)$ for all $x, y$ in X and all $k \geq K$. Hence we have

Proposition 1

For all $k \geq K$, for all $x \in X$

$$f(x) = \max \{g_k(x, y) \mid y \in X\} . \tag{2.4}$$

Moreover

$$\arg\max \{g_k(x, y) \mid y \in X\} = x. \tag{2.5}$$

□

Hence, provided that k is greater than or equal to K, problem P is equivalent to problem

$$P^k = \min_{x \in X} \max_{y \in X} \{g_k(x, y)\} \tag{2.6}$$

We can now state our first algorithm for solving $\min \{f(x) \mid x \in X\}$.

Algorithm 1

Data: $x_0 \in X$; $X_0$, a discrete subset of X; $k \geq K$.

Step 0: Set $i = 0$.

Step 1: Compute $x_i$, a solution of

$$\min_{x} \max_{y} \{g_k(x, y) \mid x \in X, y \in X_i\}.$$

Step 2: Set $X_{i+1} = X_i \cup \{x_i\}$.

Step 3: Set $i = i + 1$. Go to Step 1. □

The convergence properties of this algorithm are easily established. Let $f^0$ denote $\min \{f(x) \mid x \in X\}$.

Theorem 1

Suppose f is globally Lipschitz continuous with Lipschitz constant K and that $k \geq K$. Then any accumulation point $x^*$ of an infinite sequence $\{x_i\}$ generated by Algorithm 1 is a global solution of $\min \{f(x) | x \in X\}$ so that $f(x^*) = f^O$. Moreover $\max_y \{g_k(x_i, y) | y \in X_i\} \nearrow f^O$ as $i \to \infty$. □

Proof

For all $Y \subset X$ let $\psi_Y : X \to \mathbb{R}$ and $\psi_Y^O \in \mathbb{R}$ be defined by:

$$\psi_Y(x) \triangleq \max_y \{g_k(x, y) | y \in Y\}, \tag{2.7}$$

and

$$\psi_Y^O \triangleq \min_x \{\psi_Y(x) | x \in X\}. \tag{2.8}$$

Clearly $f(x_i) \geq f^O$ for all i and $f(x_i) \xrightarrow{I} f(x^*) \geq f^{O'}$ for some subsequence I of {0, 1, 2... }. Since $X_i \subset X_{i+1}$ it follows that $\psi_{X_i}(x_i) = \psi_{X_i}^O \leq \psi_{X_{i+1}}(x_{i+1}) = \psi_{X_{i+1}}^O$ for all i so that $\psi_{X_i}(x_i) \nearrow w^*$, say, as $i \to \infty$. Also $\psi_{X_i}(x_i) = \psi_{X_i}^O \leq f^O$ for all i so that $w^* \leq f^O$. But $\psi_{X_i}(x_i) \geq f(x_j) - k\| x_i - x_j \|_\infty$ for all $i, j \in I$, $i > j$ so that $w^* \geq f(x^*) \geq f^O$. Hence $w^* = f(x^*) = f^O$. □

If X is defined by affine inequalities (i.e. $X \triangleq \{x | Cx + d \leq 0\}$)

then the optimization problem in Step 1 may be expressed (for some A, b, c) in the form:

$$\min_{x,w} \{w | g_k(x, y) \leq w \text{ for all } y \in X_i;\ Cx + d \leq 0\}$$

$$= \min_{x,w} \{w | Ax + bw + c \leq 0,\ Cx + d \leq 0\}, \tag{2.9}$$

i.e. as a linear program.

Since in Algorithm 1 the cardinality of the set $X_i$ increases monotonically, thus increasing the complexity of the optimization problem in Step 1, we examine next the possibility of discarding element from $X_i$.

## 3. REDUCING $X_i$

In this section we show how the growth of the set $X_i$ may be moderated by discarding at each iteration those elements which are estimated to be irrelevant. At iteration i only those elements $x_j$ in $X_i$ for which $f(x_j) - \psi_{X_j}(x_j)$ is 'sufficiently large' are retained. To quantify 'sufficiently large' we introduce a double indexed sequence $\{\varepsilon_{i,j}\}$ having the following properties:

i) $\varepsilon_{i,i} = 0$ and $\varepsilon_{i,j} > 0$ for all i, j, i > j,

ii) $\varepsilon_{i,j} \nearrow \hat{\varepsilon}_j$, uniformly in j, as $i \to \infty$,

iii) $\hat{\varepsilon}_j \searrow 0$ as $j \to \infty$.

An example of such a sequence is $\varepsilon_{i,j} \triangleq \delta^j - \delta^i$ where $\delta \in (0, 1)$; in this case $\hat{\varepsilon}_j = \delta^j$. At iteration i, the element $x_j$ of $X_i$ is retained in $X_{i+1}$ if

$$f(x_j) - \psi_{X_j}(x_j) > \varepsilon_{i,j}. \tag{3.1}$$

Hence we obtain

Algorithm 2

Data: $x_0 \in X$; $X_0$, a discrete subset of X;
$k \geq K$; $\{\varepsilon_{i,j}\}$.

Step 0: Set i = 0.

Step 1: Compute $x_i$, the solution of

$$\min_x \max_y \{g_k(x, y) \mid x \in X, y \in X_i\}.$$

Step 2: Set $X_{i+1} = \{x_i\} \cup \{x_j \in X_i \mid f(x_j) - \psi_{X_j}(x_j) > \varepsilon_{i,j}\}$.

Step 3: Set i = i+1. Go to Step 1. □

To establish that the algorithm generates convergent subsequences we require the following results.

Proposition 1

If $\{X_i\}$ is a sequence of discrete subsets of X and $\{x_i\}$ a sequence of

points in X satisfying

(i) $x_i \to x^*$,

(ii) $\psi_{X_i}(x_i) \to \psi_X(x^*) = f(x^*)$,

(iii) $\psi_{X_i}(x_i) - \psi^0_{X_i} \to 0$

as $i \to \infty$, then

$$f(x^*) = f^0.$$

Proof

From (ii) and (iii), $\psi^0_{X_i} \to f(x^*)$ as $i \to \infty$. Since $\psi^0_{X_i} \leq f^0$ for all $i$ it follows that $f(x^*) \leq f^0$. Because $f^0 \triangleq \min \{f(x) \mid x \in X\} \leq f(x^*)$, it follows that $f(x^*) = f^0$. □

Our main result is

Theorem 2

Suppose f is globally Lipschitz continuous with Lipschitz constant K and that $k \geq K$. Then any accumulation point $x^*$ of an infinite sequence $\{x_i\}$ generated by Algorithm 2 is a global solution of $\min \{f(x) \mid x \in X\}$ .

Proof

Since $\{x_i\}$ is compact so is $\{\psi_{X_i}(x_i)\}$. Consider therefore a subsequence I of $\{0, 1, 2, \ldots\}$ such that $x_i \xrightarrow{I} x^*$ and $\psi_{X_i}(x_i) \xrightarrow{I} w^*$. From Step 2, $\psi_{X_i}(x_i) = \psi^0_{X_i}$ for all i. If we can show that $w^* = f(x^*)$, it follows from Proposition 1 that $f^0 = f(x^*)$.

Assume, therefore, contrary to what is to be proven, that $w^* < \psi_X(x^*) = f(x^*)$. It follows that there exists a $j_0 \in I$ such that $\psi_X(x_j) - \psi_{X_j}(x_j) > \hat{\varepsilon}^j > \varepsilon_{i,j}$ for all $i, j \in I$ such that $i > j \geq j_0$. Hence $x_j \in X_i$ so that

$$\psi_{X_i}(x_i) \geq f(x_j) - k\| x_i - x_j \|_\infty$$

for all $i, j \in I$, $i > j \geq j_0$. Consequently $w^* \geq f(x^*)$, contradicting our assumption that $w^* < f(x^*)$. Hence $w^* = f(x^*)$, i.e. $\psi_{X_i}(x_i) \xrightarrow{I} f(x^*)$. By Proposition 1, $f(x^*) = f^0$. □

Hence Algorithm 2 solves the global minimization problem $\min \{f(x) \mid x \in X\}$. Of course, considerable computational expense is involved. In effect $\psi_{X_i}$ must become an increasingly good approximation to f so that the cardinality of $X_i$ eventually becomes very high. Hence we investigate next an algorithm for determining a local rather than a global minimum.

## 4. DETERMINATION OF A LOCAL MINIMUM

We wish to compute an $x^*$ such that $x^*$ minimizes f over a $\delta$-neighbourhood of $x^*$. Thus now we merely require that $\psi_{X_i}$ becomes an increasingly good

approximation to f over this neighbourhood, a much less stringent condition than that required for global minimization. We assume again that $k \geq K$.

For all $x \in X$, $\delta > 0$, $Y \subset X$ let $N_\delta(x)$ denote the set $\{x' \in X \,|\, \|x' - x\|_\infty \leq \delta\}$ and let $f^0_\delta : X \to \mathbb{R}$ and $\psi^0_{Y,\delta} : X \to \mathbb{R}$ be defined by

$$f^0_\delta(x) \triangleq \min_{x'} \{f(x') \,|\, x' \in N_\delta(x)\} \tag{4.1}$$

$$\psi^0_{Y,\delta}(x) \triangleq \min_{x'} \{\psi_Y(x') \,|\, x' \in N_\delta(x)\} \tag{4.2}$$

A point $x^*$ in X is a local minimizer for $\min_x\{f(x) \,|\, x \in X\}$ if $f^0_\delta(x^*) = f(x^*)$ and is local minimizer for $\min_x\{\psi_Y(x) \,|\, x \in X\}$ if $\psi^0_{Y,\delta}(x^*) = \psi_Y(x^*)$.

We can now state our algorithm for determining local minima of f(x).

Algorithm 3

Data: $x_0 \in X$; $X_0$, a discrete subset of X;
$\delta > 0$; $k \geq K$; $\{\varepsilon_{i,j}\}$.

Step 0: Set $i = 0$.

Step 1: Compute $x_i \in X$ such that
$\psi^0_{X_i,\delta}(x_i) = \psi_{X_i}(x_i)$.

Step 2: Set $X_{i+1} = \{x_i\} \cup \{x_j \in X_i \,|\, f(x_j) - \psi_{X_j}(x_j) > \varepsilon_{i,j}\}$.

Step 3: Set $i = i + 1$. Go to Step 1. □

To analyse this algorithm we require an extension of Proposition 1.

Proposition 2

If $\{X_i\}$ is a sequence of discrete subsets of X and $\{x_i\}$ a sequence of points in X satisfying

(i) $x_i \rightarrow x^*$,

(ii) $\psi_{X_i}(x_i) \rightarrow \psi_X(x^*) = f(x^*)$,

(iii) $\psi_{X_i}(x_i) - \psi^O_{X_i,\delta}(x_i) \rightarrow 0$

as $i \rightarrow \infty$, then

$$f^O_\delta(x^*) = f(x^*).$$

Proof

From (ii) and (iii), $\psi^O_{X_i,\delta}(x_i) \rightarrow f(x^*)$ as $i \rightarrow \infty$. Since $\psi^O_{X_i,\delta}(x_i) \leq f^O_\delta(x^*)$ for all i, and since $f^O_\delta$ is continuous, it follows that $f(x^*) \leq f^O_\delta(x^*)$. Because $f^O_\delta(x^*) \leq f(x^*)$ (by definition) it follows that $f^O_\delta(x^*) = f(x^*)$, i.e. $x^*$ is a local minimizer of f. □

We can now establish the convergence properties of Algorithm 3.

Theorem 3

Suppose f is globally Lipschitz continuous with Lipschitz constant K and that $k \geq K$. Then any accumulation point $x^*$ of an infinite sequence $\{x_i\}$ generated by Algorithm 3 is a local minimizer of $\min \{ f(x) \mid x \in X\}$ (in the sense that $f^0_\delta(x^*) = f(x^*)$).

Proof

Consider a subsequence I of $\{0, 1, 2, \ldots\}$ such that $x_i \xrightarrow{I} x^*$ and $\psi_{X_i}(x_i) \xrightarrow{I} w^*$. From Step 2 of the algorithm $\psi^0_{X_i,\delta}(x_i) = \psi_{X_i}(x_i)$ for all i, so that (from Proposition 2) $f^0_\delta(x^*) = f(x^*)$ provided that $w^* = f(x^*)$. But this is established in the proof of Theorem 2. Hence $f^0_\delta(x^*) = f(x^*)$. □

To conclude we need to provide a subalgorithm, required in Step 1, for determining a local minimum of $\psi_{X_i}$. This problem is no longer equivalent to a linear program. A suitable subalgorithm is

Subalgorithm for Step 1

Data: $X_i \in X$, $\delta > 0$, k, $x_{i-1}$.

Step 0: Set $j = 0$. Set $\bar{x}_0 = x_{i-1}$.

Step 1: Compute $\bar{x}_j$, a solution of

$$\min_x \{ \psi_{X_i}(x) \mid x \in N_\delta(\bar{x}_{i-1}) \}.$$

Step 2: If $\psi^{0}_{x_i,\delta}(\bar{x}_j) = \psi_{x_i}(\bar{x}_j)$
set $x_i = \bar{x}_j$ and stop. Else
set $j = j + 1$ and go to Step 1. □

The minimization in Step 1 can be recast as a linear program. The program terminates in a finite number of iterations. In practice it may be desirable to vary $\delta$, starting with a high value and reducing it finitely often to a suitable small value.

## 5. CHOOSING k

Until now we have assumed that K is known, or, more precisely, that $k > K$. We investigate here a heuristic for choosing k when K is not known. At iteration i for all x let $\hat{K}_i(x)$ (our current estimate of K at x) be defined by:

$$\hat{K}_i(x) \triangleq \max_y \{ |f(x) - f(y)| \,/\, \|x - y\|_\infty \,|\, y \in X_i \} \tag{5.1}$$

Let $\gamma > 1$ be given. Our modified algorithm replaces Step 1 in Algorithms 1 and 2 by

Step 1': Choose $x_i$ and $k_i \in \{k_{i-1}, \gamma k_{i-1}, \gamma^2 k_{i-1}, \ldots\}$
such that $k_i \geq \gamma \hat{K}_i(x_i)$ and $x_i$ is a solution of
$\min_x \max_y \{ g_{k_i}(x, y) \,|\, x \in X, y \in X_i \}$.

It is relatively simple to construct an algorithm to determine such an $x_i$, $k_i$ in a finite number of iterations. Because $k_i$ is increased to at least $\gamma k_{i-1}$ if it is increased at all, $k_i$ can only be increased a finite

number of times so that $k_i = k$ for all i sufficiently large. However it is not necessarily true that $k \geq K$. We conjecture that the algorithm approximates the solution to the original problem in the sense that it may overlook steep valleys. This latter possibility may be reduced by choosing (in Step 1') $k_i$ to satisfy $k_i \geq \hat{\gamma K}_i(x)$ for a set of points in a small neighbourhood of $x_i$.

A similar updating rule may be employed in Algorithm 3.

## 6. CONCLUSIONS

The algorithm is simple to describe and implement. Its main disadvantage is, of course, the amount of computation required. This is particularly severe if Algorithm 1 is employed. It is possible, however, that Algorithm 3 together with the procedure, described in Section 5, for choosing $k_i$, may be acceptable for problems which do not possess the semi-smooth property required by other algorithms for non-differentiable optimization.

## REFERENCES

[1] Lemarechal, C., "Nondifferentiable optimization, subgradient and ε-subgradient methods", Lecture Notes, No. 117, Optimization and Operations Research, Springer Verlag, New York, 1976.

[2] Berge, C., Topological Spaces, Macmillan Co., N.Y., 1963.

[3] Goldstein, A.A., "Optimization of Lipschitz continuous function", Math. Programming, Vol. 13, pp 14-22, 1977.

[4] Polak, E., D.Q. Mayne, and Y.Y. Wardi, "On the extension of constrained optimization algorithms from differentiable to non-differentiable problems", University of California, Berkeley, Electronics Research Laboratory Memo No. UCB/ERL M8L/78, April 14, 1981, SIAM J. Control and Optimization, in press.

[5] Mifflin, R., "Semi-smooth and semi-convex functions in constrained optimization", SIAM J. Control and Optimization, Vol. 15, No. 2, pp 959-973, 1977.

[6] Mayne, D.Q. and E. Polak, "A quadratically convergent algorithm for solving infinite dimensional inequalities", U. of California, Elec. Research Lab., Memo No. UCB/ERL M/80/11, 1980. J. of Appl. Math. and Optimization, in press.