Journal of Global Optimization manuscript No.
(will be inserted by the editor)

On the Pareto-Lipschitzian Optimization

Jonas Mockus

Received: date / Accepted: date

Abstract A well-known example of global optimization that provides solutions within fixed error limits is optimization of functions with a known Lipschitz constant. In many real-life problems this constant is unknown.
To address that, we propose a novel method called Pareto Lipshitzian Optimization (PLO) that provides solutions within fixed error limits for functions with unknown Lipschitz constants. In the proposed approach, a set of all unknown Lipschitz constants is regarded as multiple criteria using the concept of Pareto Optimality (PO).
We compare PLO to the existing algorithm DIRECT. We show that in contrast to PLO, the DIRECT algorithm considers only a small subset of PO decisions that are selected by a heuristic rule depending on an adjustable parameter. It means that some PO decisions are preferred to others. In contrast, PLO regards all PO decisions without preferences and is naturally suited to utilize highly parallel computing.

Keywords Pareto optimality · Lipschitz functions · Global optimization

Institute of Mathematics and Informatics
Tel.: +3705 2109338
E-mail: jmockus@gmail.com

1 Introduction

1.1 Worst Case Analysis

The traditional approach to numerical optimization methods is to design a sequence of points

$$x_i \in A \subset R^K,\ i = 1,\ldots,n$$

such that the sequence of the current best points $\arg min_{1\le i\le n} f(x_i)$ converges to an exact or ε-exact [1] solution $x^* = \arg\min_x f(x)$, in all functions f from a given family, as $n \to \infty$. In simple cases, often connected with convexity, the convergence rate can be determined as well.

In terms of the decision theory this approach can be considered as the "Worst Case Analysis". It means that the method must retain the exactness or ε-exactness in all cases, including the worst one. A disadvantage of Worst Case Analysis is that it, in general, requires many evaluations of $f(x)$. To obtain the exact solution in the worst case, one may need many iterations if the family of problems is large. An important advantage of Worst Case Analysis is a well defined maximal deviation.

The well known examples of Worst Case Analysis are optimization methods for the set of Lipschitz functions with known Lipschitz constants, see, e.g., [1,9,10]. Here the deviation can be defined in terms of the objective function:

$$\Delta_f = |f(x) - f(x^*)| \le \omega\|x - x^*\|, \tag{1}$$

where ω is the Lipschitz constant.

For a wider family of functions, such as Lipschitz functions with an unknown constant, only the deviation in terms of function arguments can be ensured.

$$\Delta_x = \|x - x^*\|. \tag{2}$$

In [11], the problem of global optimization for a family of Lipschitz functions with unknown Lipschitz constants is regarded. In this case, the uniform grid on a compact feasible set is the optimal passive method, in the mini-max sense [11]. The term "passive" means that all points of observations (x_i) are determined at the start, the term "observation" denotes an evaluation of the objective function $f(x)$ at some fixed point x, and the term "mini-max" means minimization of maximal deviation. Here the number of required observations will be increasing exponentially with the complexity of the problem. We define the complexity as the number of variables and the accuracy of solutions [5].

The contribution of this paper is the definition of the problem of optimization with unknown Lipschitz constants in terms of Pareto optimality and the Pareto-Lipschitzian Optimization (PLO) method to realize this approach. Furthermore, PLO is compared to the existing DIRECT algorithm [4,2], which is a well-known active method for optimization of Lipschitz functions with unknown constants. To increase the efficiency of search, the DIRECT algorithm uses heuristic rules depending on some manually chosen parameter. We show that the DIRECT algorithm considers a subset of Pareto Optimal (PO) decisions determined by this heuristic.

In contrast, the proposed algorithm explores all PO decisions. Exploring all PO decisions is particularly suitable for parallel computations and honors the condition of "no preferences" which is the basic concept of Pareto optimality.

[1] When $\|x^*(\varepsilon) - x^*\| \le \varepsilon$

2 Pareto-Optimal Approach: Dominant Analysis

The concept of Pareto optimality (see, e.g., [8,**?**,**?**]) is traditionally used in cases where an objective is a vector-function $f_\omega(x),\ \omega \in \Omega$. Here $x \in D \subset R^K$ is the control parameter, ω is a component index of the vector-objective $f_\omega(x)$, and Ω is a set of all components ω.

Definition 1 The decision $x \in D$ dominates the decision $x^* \in D$, if

$$\begin{aligned} &f_\omega(x) \leq f_\omega(x^*), \quad for\ all\ \ \omega \in \Omega \\ &f_\omega(x) < f_\omega(x^*), \quad for\ at\ least\ one\ \ \omega \in \Omega. \end{aligned} \tag{3}$$

Here D is the decision space and Ω is the set of components ω.

Definition 2 The decision $x^* \in D$ is called Pareto Optimal (PO) [2], if there is no dominant decision $x \in D$

3 Pareto-Lipschitzian Optimization (PLO)

We start to explain the optimization of Lipschitz functions with unknown constants by considering this one-dimensional example.
Suppose that the interval $D = [a,b] \in R$ is partitioned into intervals $[a_i,b_i],\ i = 1,...,I$ of lengths $l_i = b_i - a_i$ with midpoints $c_i = (b_i + a_i)/2$ and the values of the function $f_\omega(x)$ are known only at the midpoints c_i. The unknown Lipschitz constants are regarded as different components of multiple criteria. The variables x are represented by the intervals $a_i \leq x \leq b_i$ and the function $f_\omega(x)$ is approximated by the lower bounds:

$$f_\omega(x) \geq f(c_i) - \omega\ l_i/2,\ a_i \leq x \leq b_i. \tag{4}$$

Definition 3 The interval $i : a_i \leq x \leq b_i$ belonging to a compact set $D \in R$ dominates the interval $j : a_j \leq x \leq b_j$ if

$$f(c_i) - \omega\ l_i/2 \leq f(c_j) - \omega\ l_j/2 \quad for\ all\ \omega \in \Omega \tag{5}$$

$$f(c_i) - \omega\ l_i/2 < f(c_j) - \omega\ l_j/2,\ \ for\ at\ least\ one\ \omega \in \Omega. \tag{6}$$

Expressions (5)(6) show that the lower bound of the interval i is increasing with $f(c_i)$ and decreasing with l_i for all ω.
In [6] the set of Pareto Optimal (PO) intervals is defined as follows:

Definition 4 The interval $j : a_j \leq z \leq b_j$ belonging to a compact set $D \in R$ is called Pareto Optimal (PO)[3] if there is no dominant interval i defined by (5)(6).

[2] Here we consider minimization, while in maximization the inequalities should be reversed.

[3] Here PO decisions are defined as indexes of PO intervals. In expression (3) PO decisions are expressed as continuous variables x.

4 Comparison of PLO with the DIRECT algorithm

The DIRECT algorithm [4,2] for Lipschitzian optimization with unknown constants is defined as a heuristic without any references to the theory of vector optimization or Pareto optimality. However, it can be explained in terms of PLO as well. The basic idea of DIRECT is to select (and sample within) all Potentially Optimal (PTO) intervals during an iteration. A formal definition of PTO intervals follows.

Definition 5 The interval j is said to be PTO if there exists some rate-of-change constant $\omega > 0$ such that

$$f(c_j) - \omega\, l_j/2 \leq f(c_i) - \omega\, l_i/2 \ \textit{for all}\ i = 1,...,I, \tag{7}$$

$$f(c_j) - \omega\, l_j/2 \leq f_{min} - \varepsilon\, |f_{min}|. \tag{8}$$

Here $\varepsilon > 0$ is a constant that defines the size of the set of PTO intervals, and f_{min} is the current best value.

One can see that inequality (7) includes all the intervals which are among the best for at least one $\omega > 0$. It means that these intervals are not dominated, thus, they belong to the PO set as well. However, condition (7) defines only a subset of not dominated intervals. The further restriction of this part of PO intervals is provided by condition (8) which depends on the parameter ε. Thus the actual performance of DIRECT algorithm is determined by this parameter.

Example 1 Let us consider three intervals:
$l_1 = 3,\ l_2 = 2,\ l_3 = 1.$
$f(c_1) = 3,\ f(c_2) = 2,\ f(c_3) = 1.$
The first interval is PTO if $\omega > max(2, 4/3 - 2/3\varepsilon)$,
the third interval is PTO if $2\varepsilon < \omega < 2,$ *and* $\varepsilon < 1$,
all the three intervals are PTO if $\omega = 2$ *and* $\varepsilon = 1$.

Example 1 illustrates a strong dependence of the set of PTO intervals on the parameter ε. The manual adaptation of this parameter to increase efficiency of the DIRECT algorithm for a given set of objective functions is a time consuming activity. It is worth noting that such parameters can be adjusted or optimized not only manually, but also using other optimization algorithms. Specifically, the automatic adaptation of parameters of various heuristics is investigated in [7,3].
In terms of Pareto optimality, conditions (7) and (8) define a subset of PO intervals which get the preferential treatment. Filtering of the PO set to a small subset may be helpful using single-thread computing, because a complete PO set may be quite large. However, using multi-processor technologies all PO intervals can be regarded in parallel.

Example 2 Let us consider intervals:
$l_1 = 3,\ l_2 = 2$, and $l_3 = 1$ with
$f(c_1) = 3,\ f(c_2) = 2.5$, and $f(c_3) = 1$.
All the three intervals are not dominated, so they belong to the PO set.
However, the second interval does not belong to the PTO set independently of ε. This illustrates the fact that PTO is a subset of PO. In this simple example, a single interval is excluded. In larger problems, the PTO subset represents only a small part of the entire PO set.

Numerical results in [4] have shown that DIRECT has a low computational overhead and is efficient for bound-constrained, black-box global optimization problems of small size. A constrained version of the algorithm has been in use at General Motors for solving engineering design problems since 1993. Therefore, we expect that PLO would also be effective in practice and improve upon DIRECT by considering all PO intervals and by utilizing parallel computation.

5 Definition of PLO for algorithmic realization

In this section, conditions (5) and (6) of Definition 3 of dominant intervals will be presented in the form convenient for calculations. To simplify the explanation we start from one-dimensional functions. The extension to several dimensions is straightforward.

Theorem 1 *The interval i dominates the interval j if and only if*

$$f(c_i) < f(c_j), \text{ and } l_i > l_j \tag{9}$$

or

$$f(c_i) < f(c_j), \text{ and } l_i = l_j \tag{10}$$

or

$$f(c_i) = f(c_j), \text{ and } l_i > l_j. \tag{11}$$

Proof Intervals i satisfying (9) are at least as good as the interval j for all ω and are better for small ω and for large ω. The reason is that if $f(c_i) < f(c_j)$, *and* $l_i > l_j$ then, at small ω the minimum of the lower bound is in the interval where $f(c_i)$ is minimal. At large ω this bound is were l_i is maximal.
By a similar reasoning we can draw the following conclusions.
Intervals i defined by (10) are at least as good as the interval j for all ω and are better for small ω.
Intervals i satisfying (11) are at least as good as the interval j for all ω and are better for large ω.
It means that the intervals i dominate the interval j, since they satisfy conditions (5) and (6).

To prove that no other intervals satisfy Definition 3 of dominant intervals, consider the complementary conditions:

$$f(c_i) < f(c_j), \text{ and } l_i < l_j \tag{12}$$

or

$$f(c_i) = f(c_j), \text{ and } l_i = l_j \tag{13}$$

or

$$f(c_i) = f(c_j), \text{ and } l_i < l_j \tag{14}$$

or

$$f(c_i) > f(c_j), \text{ and } l_i > l_j \tag{15}$$

or

$$f(c_i) > f(c_j), \text{ and } l_i = l_j \tag{16}$$

or

$$f(c_i) > f(c_j), \text{ and } l_i < l_j \tag{17}$$

The intervals i, defined by (12), are better for small ω, but worse for large ω.
The intervals i satisfying (13) are no better for all ω.
The intervals i, defined by (14), are no better for all ω.
The intervals i, defined by (15), are better for small ω, but worse for large ω.
The intervals i, defined by (16), are better for large ω, but worse for small ω.
The intervals i, defined by (17), are no better for all ω.

It means that the intervals defined by conditions (12) - (17) are not dominant intervals. Therefore, conditions (9) - (11) define the complete set of dominant intervals. Consequently, Definition 3 can be reduced to these conditions which are convenient for calculations.

6 Extension to several dimensions

We define the length l_i of the interval $a_i^k \le z_k \le b_i^k,\ k = 1,...,K$ in a compact subset $D \in R^K$ as the longest length:

$$l_i = max_k\ l_i^k. \tag{18}$$

where $l_i^k = b_i^k - a_i^k,\ k = 1,...,K$. The observation points c_i are in the middle $c_i^k = (b_i^k + a_i^k)/2,\ k = 1,...,K$. Then the Definition 4 of PO-intervals and Theorem 1 remain the same. Figure 1 illustrates a two-dimensional problem.

7 Sampling

The definition of PO intervals depends on ω which is unknown, so the rational strategy of sampling (choosing the points where the objective function is to be evaluated) is to investigate all the PO intervals.
To retain symmetry, instead of picking one side of the PO interval, two additional observations are made in the middle of two additional intervals, produced by dividing the initial PO interval along the longest dimension into three equal parts. Then the lengths of the new intervals will be $l_i^{new} = l_i/3$.
After the division, the new set of PO intervals is defined by conditions (9)-(11). The procedure continues by partitioning the updated PO intervals until the natural stopping condition is achieved:

$$max_i\ l_i \le \varepsilon. \tag{19}$$

Here ε defines the error limit by showing the maximal distance to the nearest observation point. The distance is defined in the maximum norm:

$$max_k\ l^k,\ k = 1,...,K. \tag{20}$$

Using the Euclidean norm, the error limit is defined as

$$max_i\ d_i \le \varepsilon \tag{21}$$

where d_i is the diameter of the interval i

$$d_i = \sqrt{\sum_k (l_i^k)^2}. \tag{22}$$

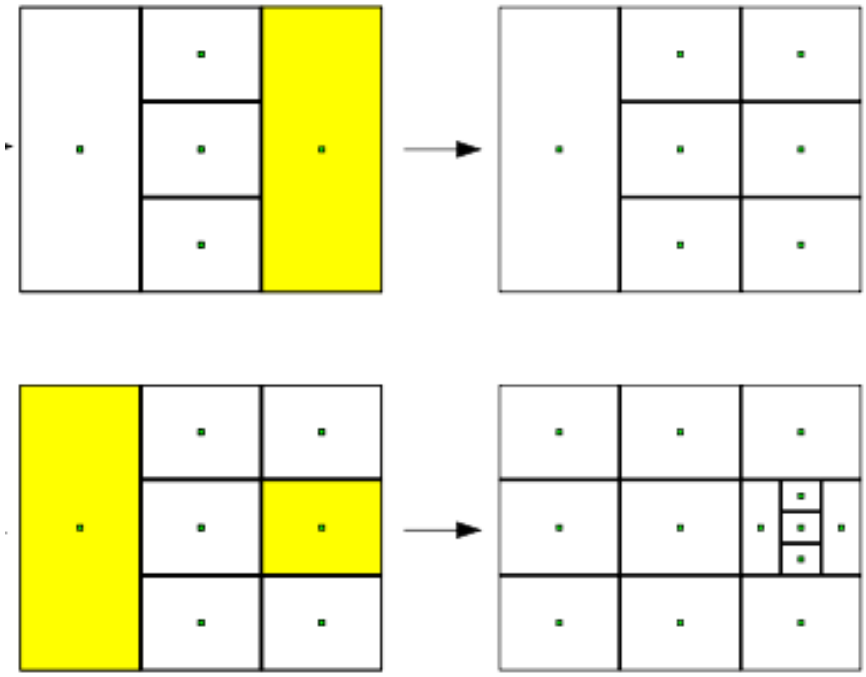

Fig. 1 Two iterations of PLO

The procedure is called the Pareto-Lipschitzian Optimization (PLO).

Figure 1 illustrates PLO. The rows in this figure represent iterations. The columns show operations. The first column illustrates the identification of the PO rectangles[4]. The second column shows how these rectangles are divided and sampled..

In the first operation (upper left), the initial partition is performed. The initial unit square is divided into three equal rectangles. Then the middle rectangle is divided again into three parts along the longest dimension. Five symmetric observations are made in the centers of these rectangles. The operation is concluded by identifying the PO interval (shaded area in this picture) by expressions (9)-(11) and (18).

During the second operation (upper right picture), the PO rectangle is divided into three equal parts, followed by sampling in the centers of the new parts.

The second row shows the second iteration. In the first operation (lower left), two new PO rectangles are identified using expressions (9)-(11) and (18). In the second operation (lower right picture) both PO rectangles are divided and additional sampling of the new rectangles is performed.

The results of the second iteration (twelve samples in the middle of twelve rectangles) serve as the initial data for the next iteration, where the new PO rectangles are to be identified.

The optimization stops when the longest interval reaches the error limit defined by expressions (19) or (21).

The sampling procedure of PLO is similar to that of the DIRECT algorithm [2]. However,

[4] The general term ”interval”, that means an interval in the k-dimensional space ($k = 1,...,K$), is used while explaining formulas. The term ”rectangle” explains pictures better.

the definition of Pareto Optimal intervals, used in PLO, is different from the definition of Potentially Optimal intervals in the DIRECT algorithm.

8 Convergence

It follows from (9) - (11) that the set of PO intervals includes the longest one. Thus, the longest intervals will be divided until they reach the error limit (19). This takes place after the finite number of partitions, which proves the following proposition:

Theorem 2 *The Pareto-Lipschitzian Optimization achieves the error limit (19) after the finite number of iterations*

9 Defining user preferences

If user preferences are not known, then all the PO intervals should be treated equally. In computing terms it means considering all the PO decisions in parallel.
The general way to represent user preferences is by supplying an importance measure to each multi-criteria component ω. Since the set of all Lipschitz functions Ω is continuous, the proper measure would be the probability density $p(\omega)$. Then

$$i(p) = arg\ min_i \int_{\Omega} (f(c_i) - \omega\ l_i/2) p(\omega) d\omega = \\ arg\ min_i (f(c_i) - E\omega\ l_i/2). \tag{23}$$

where $E\omega$ is the expected value of the Lipschitz constant.
It is well known [**?**,**?**] that $i(p) \in PO$. In this way we reduce the number of PO decisions just to one interval, as usual, thus the parallel computing is not needed. The error limit is defined as an expected error

$$\int_{\Omega} \omega\ max_i\ l_i = E\omega\ max_i\ l_i \leq \varepsilon \tag{24}$$

or

$$\int_{\Omega} \omega\ max_i\ d_i = E\omega\ max_i\ d_i \leq \varepsilon \tag{25}$$

where d_i is the diameter (22) of the interval i.
The DIRECT algorithm considered in Section 4 is a specific example of preferential treatment of some subset of PO decisions.

10 Computational aspects

The basic idea of Pareto optimality is that all the PO decisions should be treated equally, unless we know preferences of the user [5].
Each iteration $n = 1, ..., N$ includes the following tasks:
The basic task is to make observations (calculations of $f(x)$ at fixed x). The observations should be performed in parallel, if possible.

[5] Users preferences are expressed as the relative importance of different PO decisions.

The auxiliary task is of four parts:
i. definition of all PO intervals by I^2 comparisons using conditions (9)-(11);
ii. creating new intervals by splitting the PO intervals along the longest dimension;
iii. defining new observation points in the middle of new intervals;
iv. keeping the current best observation which will be accepted as the solution at the end of optimization process when $n = N$.
. The auxiliary calculations are for the central (master) processor, if the parallel computing is applied.
The PLO procedures are particularly suitable for parallel computing. To optimize real-size problems, where the time of observations is greater comparing with the time of auxiliary calculations, the parallel computing might be not only convenient but also necessary. The multi-processor technology appears to be the primary way to increase the computing power in the future.

11 Concluding remarks

The theoretical contribution is the definition of the problem of Lipschitzian optimization with unknown Lipschitz constants in terms of Pareto optimality (PO).
The computational contribution is the Pareto-Lipschitzian Optimization (PLO) method which regards all PO intervals in accordance to the basic concept of PO.
In contrast, the existing DIRECT algorithm considers a small subset of PO intervals consisting only of such intervals that are at least as good as others. This subset is filtered further by heuristic conditions depending on the parameter ε which needs to be obtained by experimentation.
Therefore, the DIRECT algorithm can be conveniently implemented using single processor machines, while the PLO method is most suitable for computing systems with large numbers of processors.
An extensive computer simulation with various test functions may reveal additional aspects of the proposed algorithm and that would be an interesting new investigation.
The presented description of the Pareto-Lipschitzian optimization can serve as a basis for discussions about possibilities and limitations of various applications of the Lipschitzian optimization for functions with unknown Lipschitz constants.

References

1. Yu.G. Evtushenko. *Numerical optimization techniques*. Optimization software, Inc., New York, 1985.
2. D.E. Finkel. Direct optimization algorithm user guide. Technical report, Center for Research in Scientific Computation. North Carolina State University, Raleigh, NC 27695-8205, 2003.
3. J.Mockus. Investigation of examples of e-education environment for scientific collaboration and distance graduate studies, part 1. *Informatica*, 17:259–278, 2006.
4. D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Application*, 79:157–181, 1993.
5. Ker-I Ko. *Complexity Theory of Real Functions*. Birkhauser, Boston, 1991.
6. J. Mockus, W. Eddy, A. Mockus, L. Mockus, and G. Reklaitis. *Bayesian Heuristic Approach to Discrete and Global Optimization*. Kluwer Academic Publishers, ISBN 0-7923-4327-1, Dordrecht-London-Boston, 1997.
7. Jonas Mockus. Bayesian heuristic approach to global optimization and examples. *Journal of Global Optimization*, 22:191–203, 2002.
8. P. Pardalos and Y. Siskos. Editorial, a historical perspective. In P.M. Pardalos and Y. Siskos, editors, *Advances in Multi-criteria Analysis*, pages ix–xxv. Kluwer Academic Publishers, 1995.
9. S.A. Pijavskij. An algorithm for finding the absolute extremum of function. *Computational Mathematics and Mathematical Physics*, pages 57–67, 1972.
10. G.O. Shubert. A sequential method seeking the global maximum of function. *SIAM Journal on Numerical Analysis*, 9:379–388, 1972.
11. A.G. Sukharev. On optimal strategies of search of extremum. *Computational Mathematics and Mathematical Physics*, pages 910–924, 1971. in Russian.