INFORMATICA, 2011, Vol. 22, No. 4, 521–536

On the Pareto Optimality in the Context of Lipschitzian Optimization

Jonas MOCKUS
Vilnius University, Institute of Mathematics and Informatics
Akademijos 4, LT-08663 Vilnius, Lithuania
e-mail: jmockus@gmail.com

Received: April 2011; accepted: September 2011

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–Lipschitzian 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 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. By contrast, PLO regards all PO decisions without preferences and is naturally suited to utilize highly parallel computing.

Keywords: Pareto optimality, Lipschitz functions, global optimization.

1. Introduction

1.1. *Worst Case Analysis*

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

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

such that the sequence of the current best points $\arg\min_{1 \leqslant i \leqslant n} f(x_i)$ converges to an exact or ϵ-exact [1] solution $x^* = \arg\min_x f(x)$, in all the 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,

[1] When $\|x^*(\epsilon) - x^*\| \leqslant \epsilon$.

including the worst one. A disadvantage of the Worst Case Analysis is that, in general, it 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 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., Evtushenko, 1985; Pijavskij 1972; Shubert, 1972). Here the deviation can be defined in terms of the objective function:

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

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 Sukharev (1971), the problem of global optimization for a family of Lipschitz functions with unknown Lipschitz constants is considered. In this case, the uniform grid on a compact feasible set is the optimal passive method, in the mini–max sense (Sukharev, 1971). The term "passive" means that all the 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 the maximal deviation. Here the number of required observations will be exponentially increasing with the complexity of the problem. We define the complexity as the number of variables and the accuracy of solutions (Ko, 1991).

The contribution of this paper is a 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, advantages and disadvantages of PLO are compared to the existing DIRECT algorithm (Jones *et al.*, 1993; Finkel, 2003; Sergeyev, 2006), 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 that depend on some manually chosen parameter and applies sophisticated techniques of parallel computing (Gablonsky, 2001; He *et al.*, 2009, 2009a, 2009c; Verstak, 2008). The DIRECT algorithm (Sergeyev, 2006) considers a subset of non-dominated decisions. Different versions of DIRECT regard different subsets of non-dominated decisions determined by the corresponding heuristics.

On the contrary, the proposed algorithm explores all PO decisions, no heuristic parameters are applied. Exploration of all PO decisions is particularly suitable for parallel computations which is important in computing. In addition, PLO honors the condition of "no preferences", which is the basic theoretical concept of Pareto optimality.

2. Pareto-Optimal Approach: Dominant Analysis

The concept of Pareto optimality (see, e.g., Pardalos and Siskos, 1995; Miettinen, 1999; Figueira *et al.*, 2004) is traditionally used in the 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 the components ω.

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

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

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 begin to explain the optimization of Lipschitz functions with unknown constants by considering the following one-dimensional example.

Suppose that the interval $D = [a, b] \subset R$ is partitioned into intervals $[a_i, b_i]$, $i = 1, \ldots, I$ of lengths $l_i = b_i - a_i$ with midpoints $c_i = (b_i + a_i)/2$ and the values $f(c_i)$ 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 $a_i \leqslant x \leqslant b_i$ and the function $f_\omega(x)$ is approximated by the lower bounds:

$$L_i(\omega) = f(c_i) - \omega\, l_i/2 \leqslant f_\omega(x), \quad a_i \leqslant x \leqslant b_i. \tag{4}$$

Expression (4) shows that the lower bound of the interval i is increasing with $f(c_i)$ and decreasing with l_i for all ω. We compare the "quality" of different intervals by their lower bounds. For example, we say that the interval is better for a given ω, if its lower bound is lower. The formal definition is as follows.

DEFINITION 3. Let us compare the interval i: $a_i \leqslant x \leqslant b_i$, belonging to a compact set $D \subset R$, with the interval j: $a_j \leqslant x \leqslant b_j$ at $\omega \in \Omega$.

The interval i is at least as good as the interval j at $\omega \in \Omega$, if $L_i(\omega) \leqslant L_j(\omega)$.

The interval i is better than the interval j at $\omega \in \Omega$, if $L_i(\omega) < L_j(\omega)$.

The interval i is worse than the interval j at given $\omega \in \Omega$, if $L_i(\omega) > L_j(\omega)$.

The interval i is not better than the interval j at given $\omega \in \Omega$, if $L_i(\omega) \geqslant L_j(\omega)$.

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

DEFINITION 4. The interval i: $a_i \leqslant x \leqslant b_i$ that belongs to a compact set $D \subset R$ dominates the interval j: $a_j \leqslant x \leqslant b_j$, if

$$L_i(\omega) \leqslant L_j(\omega) \quad \text{for all } \omega \in \Omega, \tag{5}$$

$$L_i(\omega) < L_j(\omega) \quad \text{for at least one } \omega \in \Omega. \tag{6}$$

In Mockus *et al.* (1997) and Mockus and Stasionis (2011) the set of Pareto Optimal (PO) intervals is defined as follows:

DEFINITION 5. The interval j: $a_j \leqslant z \leqslant b_j$ that belongs to the compact set $D \subset R$ is called Pareto Optimal (PO). [3] if there is no dominant interval i defined by (5) and (6).

4. Comparison of PLO with the DIRECT Algorithm

The DIRECT algorithm (Jones *et al.*, 1993; Finkel, 2003; Gablonsky, 2001) for Lipschitzian optimization with unknown constants is defined as a heuristic without any references to the theory of Pareto optimality. However, it can be explained in terms of PLO, too. 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 6. The interval i is said to be PTO if there exists some rate-of-change constant $\omega > 0$ such that

$$L_i(\omega) \leqslant L_j(\omega) \quad \text{for all } j = 1, \ldots, I, \tag{7}$$

$$L_i(\omega) \leqslant f_{\min} - \epsilon\,|f_{\min}|. \tag{8}$$

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

It was shown in Sergeyev (2006) that the DIRECT intervals are not dominated, thus, they belong to the PO set as well. However, condition (7) restricts a subset of not dominated intervals. A further restriction of the subset of PO intervals is provided by condition (8) which depends on the parameter ϵ.

The actual performance of the DIRECT algorithm is determined by this parameter. The adjustable parameters are useful when adapting an algorithm to specific applications. However, it makes a comparison with other algorithms more difficult. For example, it is not clear how to compare such an algorithm to that without adjustable parameters.

EXAMPLE 1. Let us consider three intervals:

$$l_1 = 3, \qquad l_2 = 2, \qquad l_3 = 1,$$
$$f(c_1) = 3, \qquad f(c_2) = 2, \qquad f(c_3) = 1.$$

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

The first interval is PTO, if $\omega > \max(2, 4/3 + 2/3\epsilon)$, the third interval is PTO, if $2\epsilon < \omega < 2$, and $\epsilon < 1$, all the three intervals are PTO, if $\omega = 2$ and $\epsilon = 1$.

Example 1 illustrates a strong dependence of the set of PTO intervals on the parameter ϵ. The manual adaptation of this parameter to increase the efficiency of the DIRECT algorithm for a given set of objective functions is a time consuming activity. It is noteworthy 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 Mockus (2002, 2006) using the Bayesian approach. Some other approaches are described in Dzemyda and Sakalauskas (2009, 2011).

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 for using a 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 the intervals:

$$\begin{array}{lll} l_1 = 3, & l_2 = 2, & l_3 = 1 \text{ with} \\ f(c_1) = 3, & f(c_2) = 2.5, & f(c_3) = 1. \end{array}$$

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 Jones *et al.* (1993), Gablonsky (2001), Sergeyev (2006) have shown that DIRECT has a low computational overhead and is efficient for bound-constrained, black-box global optimization problems of small size. The DIRECT algorithm has been in use for solving engineering design problems since 1993 (Jones, 1993; Finkel, 2003; Sergeyev, 2006; Gablonsky, 2001; He *et al.*, 2009, 2009a, 2009c; Verstak, 2008). Therefore, we expect that PLO would also be efficient in practice and, when considering all PO intervals and utilizing parallel computation will yield better results than DIRECT.

5. Definition of PLO for Algorithmic Realization

In this section, conditions (5) and (6) of Definition 4 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*

$$f(c_i) < f(c_j) \quad \textit{and} \quad l_i > l_j \quad \textit{or} \tag{9}$$
$$f(c_i) < f(c_j) \quad \textit{and} \quad l_i = l_j \quad \textit{or} \tag{10}$$
$$f(c_i) = f(c_j) \quad \textit{and} \quad l_i > l_j. \tag{11}$$

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, for small ω the minimum of the lower bound is in the interval where $f(c_i)$ is minimal. At large ω this bound is in the interval i the length l_i of which is maximal.

By similar reasoning we can draw the following conclusions.

The intervals i defined by (10) are at least as good as the interval j for all ω and are better for small ω.

The 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).

It follows from (9)–(11) that $L_i(\omega) < L_j(\omega)$ for all $\omega > 0$, since

$$L_i(\omega) - L_j(\omega) = f(c_i) - f(c_j) + \omega/2(l_j - l_i) < 0. \tag{12}$$

To prove that no other intervals satisfy Definition 4 of dominant intervals, consider the complementary conditions:

$$f(c_i) < f(c_j) \quad \text{and} \quad l_i < l_j \quad \text{or} \tag{13}$$
$$f(c_i) = f(c_j) \quad \text{and} \quad l_i = l_j \quad \text{or} \tag{14}$$
$$f(c_i) = f(c_j) \quad \text{and} \quad l_i < l_j \quad \text{or} \tag{15}$$
$$f(c_i) > f(c_j) \quad \text{and} \quad l_i > l_j \quad \text{or} \tag{16}$$
$$f(c_i) > f(c_j) \quad \text{and} \quad l_i = l_j \quad \text{or} \tag{17}$$
$$f(c_i) > f(c_j) \quad \text{and} \quad l_i < l_j. \tag{18}$$

The intervals i, defined by (13), are better for small ω, but worse for large ω.

The intervals i satisfying (14) are no better for all ω.

The intervals i, defined by (15), are worse for all $\omega > 0$ and no better for $\omega = 0$.

The intervals i, defined by (16), are better for large ω and worse for small ω.

The intervals i, defined by (17), are worse for all ω.

The intervals i, defined by (18), are worse for all ω.

It means that the intervals defined by conditions (13)–(18) are not dominant intervals. It follows from (13)–(18) that $L_i(\omega) \geqslant L_j(\omega)$ for all $\omega > 0$, since

$$L_i(\omega) - L_j(\omega) = f(c_i) - f(c_j) + \omega/2(l_j - l_i) \geqslant 0. \tag{19}$$

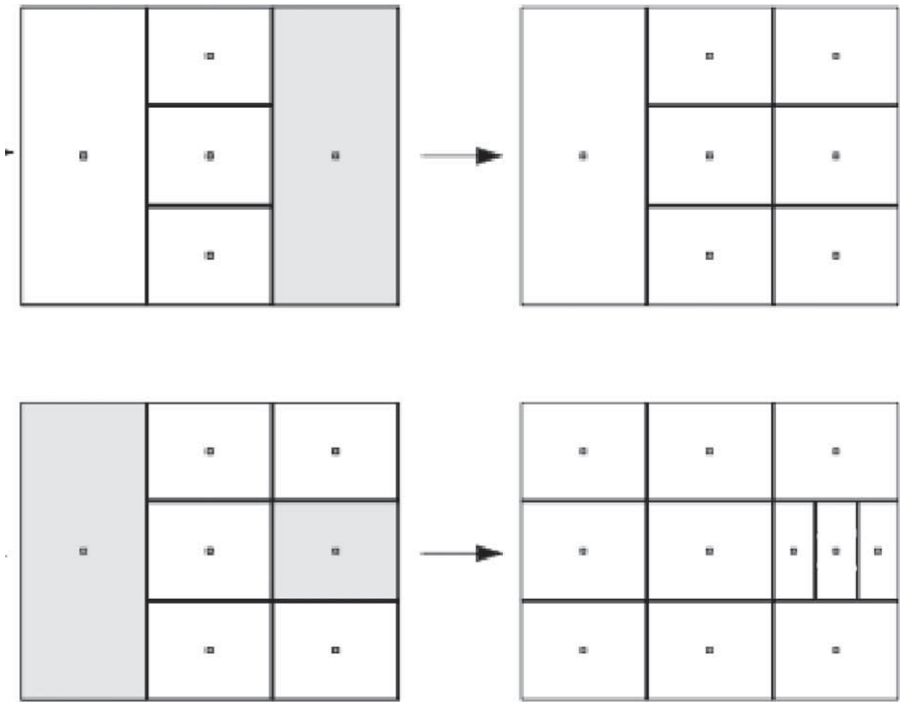

Fig. 1. Two iterations of PLO.

Therefore, conditions (9)–(11) define the complete set of dominant intervals. Consequently, Definition 4 can be reduced to these conditions which are convenient for calculations.

6. Extension to Several Dimensions

We define the length l_i of the closed K-dimensional interval $[a_i^k, b_i^k] \subset [a^k, b^k]$, $k = 1, \ldots, K$ as the longest length:

$$l_i = \max_{k=1,\ldots,K} l_i^k. \tag{20}$$

Here $l_i^k = b_i^k - a_i^k$. The observation points c_i are in the middle $c_i^k = (b_i^k + a_i^k)/2$, $k = 1, \ldots, K$. Then Definition 5 of PO-intervals and Theorem 1 remains 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 a 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.

After the division, the new set of PO intervals is defined by conditions (9)–(11).

The maximal distance to the nearest observation point

$$\max_{i=1,\dots,n} 1/2\, l_i \leqslant \epsilon_l(n). \tag{21}$$

can be used for estimating the maximal error, if the number of observations n is fixed. Expression (21) can be used as a stopping condition, if n is not fixed.

Here the distance is defined in the maximum norm:

$$\max_k l^k,\ k = 1, \dots, K. \tag{22}$$

Using the Euclidean norm, the error limit is defined as

$$\max_{i=1,\dots,n} 1/2\, d_i \leqslant \epsilon_d(n) \tag{23}$$

where d_i is the diameter of the interval i

$$d_i = \sqrt{\sum_k (l_i^k)^2}. \tag{24}$$

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 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 (20).

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 (20). In the second operation (lower right picture), both PO rectangles are divided and additional sampling of two new

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

rectangles is performed. In the DIRECT algorithm (Finkel, 2003), four new rectangles are sampled, instead of two new rectangles in PLO.

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 (21) or (23).

The sampling procedure of PLO is similar, but not identical, to that of the DIRECT algorithm (Finkel, 2003). However, the definition of Pareto Optimal intervals, used in PLO, is different than that of Potentially Optimal intervals in the DIRECT algorithm.

8. Convergence

It follows from (9)–(11) that the set of PO intervals includes the longest interval. Thus, the longest intervals will be divided into three equal parts until they reach the error limit (21). This limit is reached after the finite number of partitions, since $[a_i^k, b_i^k] \subset [a^k, b^k]$, $k = 1, \ldots, K$. That proves the following proposition:

Theorem 2. *For any $\epsilon_l > 0$ there exists a number n_ϵ such that $\epsilon_l(n) \leqslant \epsilon_l$, if $n \geqslant n_\epsilon$.*

9. Defining User Preferences

If user preferences are not known, then all the PO intervals should be treated equally. In computing terms it means consideration of all the PO decisions in parallel.

The general way of representing users' 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

$$\begin{aligned} i(p) &= \arg\min_i \int_\Omega (f(c_i) - \omega\, l_i/2) p(\omega)\, d\omega \\ &= \arg\ \min_i (f(c_i) - E\{\omega\}\, l_i/2), \end{aligned} \tag{25}$$

where $E\{\omega\}$ is the expected value of the Lipschitz constant.

It is well known (Miettinen, 1999; Figueira *et al.*, 2004) that $i(p) \in PO$. This way, we reduce the number of PO decisions just to one interval, as usual, so the parallel computing is not needed. The error limit is defined as an expected error

$$\int_\Omega \omega\ \max_i\ l_i = E\{\omega\}\ \max_i\ l_i \leqslant \epsilon_l \tag{26}$$

or

$$\int_\Omega \omega\ \max_i\ d_i = E\{\omega\}\ \max_i\ d_i \leqslant \epsilon_d, \tag{27}$$

where d_i is the diameter (24) 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 $i = 1, \ldots, n$ 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.
- An auxiliary task consists of four parts:
  - (i) Definition of all PO intervals by I_i^2 comparisons using conditions (9)–(11), where I_i is the number of intervals at the iteration i;
  - (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 the optimization process as $i = n$.

Auxiliary calculations are for the central (master) processor, if the parallel computing is applied.

PLO procedures are particularly suitable for parallel computing. To optimize real-size problems, where the time of observations is greater as compared 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 for increasing the computing power in the future.

11. Experimental Calculations, Preliminary Results

There is a number well known methods of global optimization which converge to the exact minimum. Experimental calculations are needed to evaluate their practical efficiency. We start the experimental comparison with methods without adjustable parameters. The first results comparing the sequential version of PLO with other methods are in Mockus and Stasionis (2011). These and other calculations can be repeated independently using any of the following websites:

`http://soften.ktu.lt/~mockus`,
`http://optimum2.mii.lt/`,
`http://prof.if.ktu.lt/~jonas.mockus`,

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

section 'Software Systems', task 'GMJ4: Global Optimization of many Models with 'PloN', start "Applet long: gmj4.html", select Method: "PloNj", Task: "Xcos" etc., click "Operations" and "Run".

Here follow some results of the sequential PLO implementation using three special multi-modal test functions of 20 variables, called "Xcos" (28), "Xcos2" (29), and "Xcos3" (30), respectively (websites, 18 June, 2011).

$$\sum_{i,j=1}^{20}(x_i x_j + a\cos x_i \cos x_j), \tag{28}$$

$$\sum_{i,j=1}^{20}(x_i x_j + a\cos x_i \cos x_j + a\cos x_i x_j), \tag{29}$$

$$\sum_{i,j=1}^{20}(x_i x_j + a\cos x_i \cos x_j. \tag{30}$$

The multiplier a defines the proportion of multi-modal components and the domain determines the number of local minima. On the websites, the domains are defined by the user and seen in the task window.

In addition, we compare two models that simulate real optimization problems (Mockus, 2000). The first one, called "Eco Duel", is about a differential game that represents the competition of two servers, using the concept of Nash equilibrium. Here eight optimization variables are used. The second five-dimensional model optimizes the "mixture" of five heuristics for packing rectangular boxes of different size into the container and is called "Packer".

The parallel PLO implementation is simulated by selecting the method 'PloN' where one 'PloN' operation means making as many function evaluations as there are PLO intervals. Table 1 illustrates the comparison of the results by the test function 'Xcos3' with $a = 1000$ of sequential 'PloNj' with 'PloN' which simulates parallel operations. The term 'operation' denotes one function evaluation in 'PloNj'. In 'PloN' this term means a complete PLO operation involving function evaluation in all the PLO intervals. The total computing time is determined mainly by the number of function evaluations. Therefore,

Table 1

Comparison of sequential 'PloNj' with simulated parallel version 'PloN' using 'Xcos3'

Operations	PloNj	PloN
100	92.98	3.119
200	92.98	1.046
300	3.119	0.654
400	3.119	0.554
600	3.119	0.553

Table 2
Comparison of three algorithms minimizing five test functions.

Function	Iteration	PLO	BA	MC
Xcos, $a = 100$	$1,000$	0.001	19.455	1.798
Xcos2, $a = 100$	$1,000$	-30.093	-6.294	-6.139
Xcos3, $a = 1000$	$1,000$	3.119	14.833	20.50
EcoDuel	100	0.138	0.163	0.63
EcoDuel	$1,000$	0.023	0.119	0.336
Packer	10	-2.2623	-2.2572	-2.2119

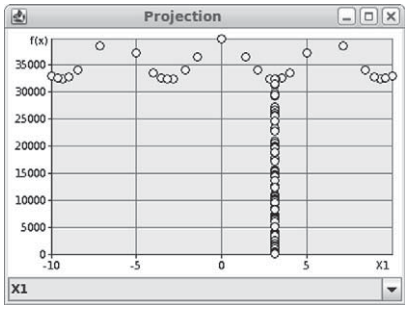

Fig. 2. Projection, first variable of "Xcos", $a = 100$.

we may expect that using the actual parallel computing the time of an operation in PLO will be close to the time of an iteration in the sequential version of PLO.

In Table 2 the results of sequential PLO are compared with two well-known methods. The first one is the "Bayesian Approach" (BA) which uses a stochastic model to predict the expected improvement (Mockus *et al.*, 1997).

BA is selected as an example of a sequential algorithm designed for optimizing both the deterministic and stochastic functions. The second algorithm is a simple Monte Carlo (MC) version which can be easily applied to parallel calculations and it is similar to PLO in this aspect. The methods converge to the global minima of continuous functions (Mockus, 1989) and do not contain adjustable parameters. Since BA and MC are stochastic, average results are shown for these algorithms.

In all the instances PLO was the best. BA was better than MC.

Figure 2 shows how the results of "Xcos" depend on the first variable at fixed values of other variables.

Figure 3 shows how the results of "Eco Duel" depend on the first variable that represents the rate of price changes. The other variables are fixed.

Table 3 illustrates the comparison of the results of Monte Carlo (MC), PLO, and DIRECT algorithms, obtained using the standard test functions of global optimization in Jones *et al.* (1993), Bjorkman and Holmstrom (1999), Liuzzi *et al.* (2010).

The symbol * in Table 3 means that the result by 'PloNj' was achieved after 1,308,332

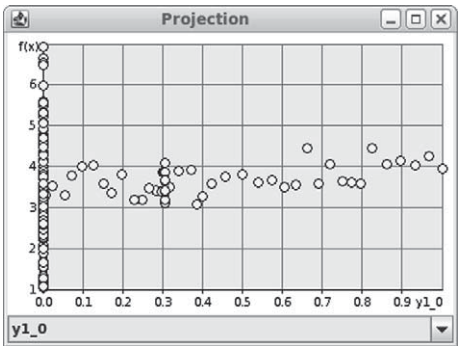

Fig. 3. Projection, first variable of "Eco Duel".

Table 3
Comparison of four algorithms using twelve test functions.

Function	Iteration	MC	DIRECT	PloNj	PloN (parallel)	$f(x*)$
Shekel, $m = 10$	97	-1.201	-10.435	-2.5834	-10.536 (50)	-10.5364
Hartman-3,	83	-3.564	-3.847	-3.862	-3.863 (20)	-3.86278
Hartman-6,	213	-2.505	-3.321	-3.321	3.322 (30)	-3.32237
Brcos	63	1.066	0.42	0.401	0.39 (50)	0.397887
GolPri	101	8.86	3.03	3.001	3.000 (20)	3.00
SixH	113	-0.82	-1.022	-1.027	-1.032 (59)	-1.031628453
Shub2D	2883	-180.53	-184.73	-123.57	186.73(120)	-186.7309
Ackley2	561	4.394	5.64609E-05	0.047	0.000(22)	0
Ackley10	6, 780, 239	$-$	9.4929E-02	0.004$*$	0.081(67)	0
Easom	7019	-0.58	-0.999989985	-0.976	-1.0(60)	-1
Griewank2	41, 005	0.06	7.88274E-07	0.0000	0.0000(20)	0
Mich2	67	-0.97	-1.801	-1.801	01.801(10)	-1.8013
Mich5	14, 005	-3.313	-4.68721324	-4.19	-4.19(50)	-4.687658

iterations, details are in Fig. 4. MC and PLO represent methods intended for parallel computing with no adjustable parameters. This version of DIRECT is for a sequential realization with one adjustable parameter ϵ selected by the method authors. Two versions of PLO are tested. The one called 'PloNj' represents sequential implementation. The second one named 'PloN' simulates parallel PLO calculations. In Table 3, the numbers of parallel operations are shown in parentheses. It means that we compare the algorithms that represent different families of optimization methods.

As expected, the efficiency DIRECT was similar to that of sequential 'PloNj', but not as good as parallel simulator 'PloN'. The comparison of parallel DIRECT with parallel PLO is an important future task.

In particular, DIRECT was better than 'PloNj' for 5 test functions: Shekel, $m = 10$, Shub2D, Ackley2, Easom, and Mich5. For three functions – Hartman-6, Griewank2, and

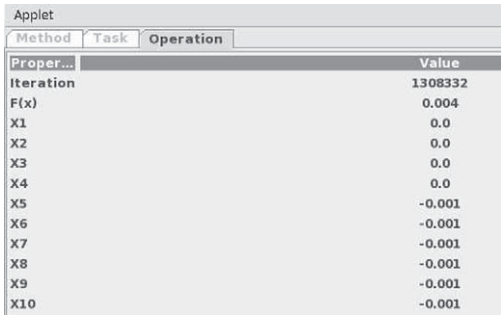

Fig. 4. The output of 'Ackley10' function optimization by 'PloNj'.

Mich2 – the results were equal. PLO was better than DIRECT for 5 functions: Hartman-3, Brcos, GolPri, SixH, and Ackley10.

Applying 'PloN' smaller numbers of the "parallel" iterations were used, to keep computing times closer to the sequential DIRECT and 'PloNj'. The results of 'PloN' were better than DIRECT for eleven functions. Using 'Mich5' the DIRECT algorithm achieved better results.

Tables 2 and 3 show that the sequential version of PLO is better comparing with methods without adjustable parameters, such as MC and BA and is close to DIRECT with adjustable parameters. The simulator of "paralleln" 'PloN' has achieved best results for almost all functions but it took longer computing times in the sequential simulation of parallel PLO.

A detailed comparison with the existing sequential and parallel versions of DIRECT is the task of further investigation. It is not a simple task, since it is not clear how to compare the results of the methods without adjustable parameters, such as PLO, with the methods having these parameters, such as DIRECT. A special attention should be paid regarding the parallel realization of DIRECT and other methods of global optimization. We may expect better results for PLO, because this algorithm is created for parallel computing. Using the parallel computing PLO may be also applied to generate a set of good starting points for additional local search in a way similar to DIRMIN (Liuzzi *et al.*, 2010).

12. Concluding Remarks

A theoretical contribution is the definition of the problem of Lipschitzian optimization with unknown Lipschitz constants in terms of Pareto optimality (PO) that provides the algorithm without adjustable parameters. It is important as compared with other methods and convenient for users.

A computational contribution is the Pareto–Lipschitzian Optimization (PLO) method which regards all the PO intervals in accordance with the basic concept of PO.

By contrast, the existing DIRECT algorithms consider a subset of PO (non-dominated) intervals, consisting of such intervals that are at least as good as others for

some positive rate-of-change constant. This subset is filtered further by heuristic conditions dependent on the parameter ϵ needed 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. There exists efficient multi-processor software (He *et al.*, 2009, 2009a, 2009c; Verstak, 2008) that implements DIRECT by means of sophisticated techniques of parallel computing.

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 on the possibilities and limitations of various applications of the Lipschitzian optimization to functions with unknown Lipschitz constants.

References

Bjorkman, M., Holmstrom, K. (1999). Global optimization using the DIRECT algorithm in Matlab 1. *Advanced Modeling and Optimization*, 1, 17–37.

Dzemyda, G., Sakalauskas, L. (2009). Optimization and knowledge-based technologies. *Informatica*, 20(2), 165–172.

Dzemyda, G., Sakalauskas, L. (2011). Large-scale data analysis using heuristic methods. *Informatica*, 22, 1–10.

Evtushenko, Y. (1985). *Numerical Optimization Techniques. Optimization Software*. Springer, New York.

Figueira, J., Greco, S., Ehrgott, M. (2004). *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer, Berlin.

Finkel, D. (2003). DIRECT optimization algorithm user guide. *Techical report*, Center for Research in Scientific Computation. North Carolina State University, Raleigh, NC 27695-8205.

He, J., Verstak, A., Sosonkina, M., Watson, L. (2009a). Performance modeling and analysis of a massively parallel DIRECT, Part 1. *Journal of High Performance Computing Applications*, 23, 14–28.

He, J., Verstak, A., Sosonkina, M., Watson, L. (2009b). Performance modeling and analysis of a massively parallel DIRECT, Part 2. *Journal of High Performance Computing Applications* 23, 29–41.

He, J., Watson, L., Sosonkina, M. (2009c). Algorithm 897: VTDIRECT95: serial and parallel codes for the global optimization algorithm DIRECT. *ACM Transactions on Mathematical Software*, 36, 1–24.

Gablonsky, J.M. (2001). *Modifications of the DIRECT algorithm*. PhD thesis, North Carolina State University, Raleigh, NC, USA.

Jones, D., Perttunen, C., Stuckman, B. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79, 157–181.

Ko, KI (1991). *Complexity Theory of Real Functions*. Birkhäuser, Boston.

Liuzzi, G., Lucidi, S., Piccialli, V. (2010). A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications*, 45, 353–375.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic, Boston.

Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic, Dordrecht.

Mockus, J. (2000). *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*. Kluwer Academic, Dordrecht.

Mockus, J. (2002). Bayesian heuristic approach to global optimization and examples. *Journal of Global Optimization* 22, 191–203.

Mockus, J. (2006). Investigation of examples of e-education environment for scientific collaboration and distance graduate studies, Part 1. *Informatica*, 17(2), 259–278.

Mockus, J., Stasionis, J. (2011). On the experimental investigation of Pareto–Lipschitzian optimization. In: *Theoretical and Applied Aspects of Cbernetics*, Kyiv, Ukraine.

Mockus, J., Eddy, W., Mockus, A., Mockus, L., Reklaitis, G. (1997). *Bayesian Heuristic Approach to Discrete and Global Optimization*. Kluwer Academic, Dordrecht.

Pardalos, P., Siskos, Y. (1995). Editorial, a historical perspective. In: Pardalos, P., Siskos, Y. (Eds.) *Advances in Multi-Criteria Analysis*. Kluwer Academic, Dordrecht, pp. ix–xxv.

Pijavskij, S. (1972). An algorithm for finding the absolute extremum of function. *Computational Mathematics and Mathematical Physics*, 57–67.

Sergeyev, YaD KD (2006). Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM Journal on Optimization*, 16(3), 910–937.

Shubert, G. (1972). A sequential method seeking the global maximum of function. *SIAM Journal on Numerical Analysis*, 9, 379–388.

Sukharev, A. (1971). On optimal strategies of search of extremum. *Computational Mathematics and Mathematical Physics*, 910–924 (in Russian).

Verstak, A SM Watson LT (2008). Design and implementation of a massively parallel version of DIRECT. *Computational Optimization and Applications*, 40(2), 217–245.

J. Mockus graduated from Kaunas University of Technology, Lithuania, in 1952. He got his doctor habilitus degree at the Institute of Computers and Automation, Latvia, in 1967. He is a principal researcher of the Systems Analysis Department, Vilinus University Institute of Mathematics and Informatics, and a professor of Vilnius Technical University. His research interests include global and discrete optimization.

Apie Pareto–Lipšico optimizaciją

Jonas MOCKUS

Optimizavimas funkcijų su žinoma Lipšico konstatnta, užtikrinantis sprendinius duotu tikslumu, yra tradicinis globaliojo optimizavimo uždavinys. Ši konstanta nėra žinoma daugelyje praktinių uždavinių.

Mes siūlom naują metodą tokiems uždaviniams spręsti, pavadintą Pareto–Lipšico optimizacija (PLO). Šis metodas pateikia sprendinius duotu tikslumu funkcijoms su nežinoma Lipšico konstanta. PLO požiūriu, Lipšico konstantos nagrinėjamos kaip vektorinio kriterijaus elementai Pareto optimalumo teorijos (PO) rėmuose.

PLO yra lyginamas su žinomu DIRECT algoritmu. Parodoma, kad DIRECT, skirtingai nei PLO, nagrinėja tik nedidelę dalį PO sprendinių, kurie yra nustatomi pagal euristinę taisyklę, priklausančią nuo laisvai parenkamo parametro. Tokiu būdu daliai PO sprendinių suteikiamas prioritetas. PLO nagrinėja be išimties visus PO sprendinius, todėl geriau tinka lygiagretiems skaičiavimams.