

Application of reduced-set Pareto-Lipschitzian optimization to truss optimization

Jonas Mockus · Remigijus Paulavičius ·
Dainius Rusakevičius · Dmitrij Šešok ·
Julius Žilinskas

Abstract In this paper, a recently proposed global Lipschitz optimization algorithm PLOR (Pareto-Lipschitzian Optimization with Reduced-set) is further developed, investigated and applied to truss optimization problems. Partition patterns of the PLOR algorithm are similar to those of DIRECT (DIviding RECTangles), which was widely applied to different real-life problems. However here a set of all Lipschitz constants is reduced to just two: the maximal and the minimal ones. In such a way the PLOR approach is independent of any user-defined parameters and balances equally local and global search during the optimization process. An expanded list of other well-known DIRECT-type algorithms is used in investigation and experimental comparison using the standard test problems and truss optimization problems. The experimental investigation shows that the PLOR algorithm gives very competitive results to other DIRECT-type algorithms using standard test problems and performs pretty well on real truss optimization problems.

J. Mockus · R. Paulavičius · J. Žilinskas
Vilnius University Institute of Mathematics and Informatics, Akademijos 4, LT-08663 Vilnius, Lithuania

J. Mockus
E-mail: jonas.mockus@mii.vu.lt

R. Paulavičius
E-mail: remigijus.paulavicius@mii.vu.lt

J. Žilinskas
E-mail: julius.zilinskas@mii.vu.lt

Dainius Rusakevičius · Dmitrij Šešok
Vilnius Gediminas Technical University, Saulėtekio al. 11, 10223 Vilnius, Lithuania

Dainius Rusakevičius
E-mail: dainius.rusakevicius@vgtu.lt

Dmitrij Šešok
E-mail: dsesok@gmail.com

Keywords Truss optimization · Lipschitz optimization · PLOR algorithm · DIRECT algorithm

1 Introduction

DIRECT [15] is a well-known partitioning-based algorithm that balances local and global search in an attempt to efficiently find the global minimizer. Optimization problems solved by DIRECT can be formulated as minimization of a multidimensional multiextremal “black-box” function:

$$f^* = f(x^*) = \min_{x \in D} f(x), \quad (1)$$

where the feasible region $D = [a, b] = \{x \in \mathbb{R}^n : a(i) \leq x(i) \leq b(i), 1 \leq i \leq n\}$ is n -dimensional hyper-rectangle, a, b are given vectors in \mathbb{R}^n , and $\|\cdot\|$ denotes the Euclidean norm. For convergence reason usually it is assumed, that the real-valued objective function f is Lipschitz-continuous with unknown Lipschitz constant over the feasible region, i.e. for every $x, y \in D$ there exists a constant $0 < L < \infty$, such that

$$|f(x) - f(y)| \leq L\|x - y\|.$$

Moreover, if the objective function is continuous in the neighborhood of the global minimizer, then DIRECT can sample deterministically a point near enough to the minimizer [14]. Note, that extension of DIRECT was proposed in the same paper [14] to handle inequality and integer constraints, but this case is outside of interest in this paper.

The DIRECT algorithm includes two major procedures: partition (DIviding RECTangles) and sampling. It begins with scaling the feasible region into the unit hyper-cube and referring to the original space D only when evaluating the values of the objective function. In every iteration the DIRECT algorithm identifies some potentially optimal hyper-rectangles (which can potentially contain the global minimizers), satisfying Definition 1. Then the DIRECT algorithm computes values of the objective function at sample points in each potential hyper-rectangle and subdivides it along the longest coordinate directions starting from the directions with the lowest function values.

Definition 1 Let S be a set of all hyper-rectangles created by DIRECT after k iterations. Let $c_i = (a_i + b_i)/2$ denote the center point of the i -th hyper-rectangle, and let $d_i = \|b_i - a_i\|/2$ denote the distance from the center to the vertices (radius). Let $\varepsilon > 0$ be a positive constant and f_{min} be the currently known best function value. A hyper-rectangle $S_j \in S$ is said to be potentially optimal if there exists some rate-of-change constant $\tilde{K} > 0$ such that

$$f(c_j) - \tilde{K}d_j \leq f(c_i) - \tilde{K}d_i, \quad \forall i \in S \quad (2)$$

$$f(c_j) - \tilde{K}d_j \leq f_{min} - \varepsilon|f_{min}|. \quad (3)$$

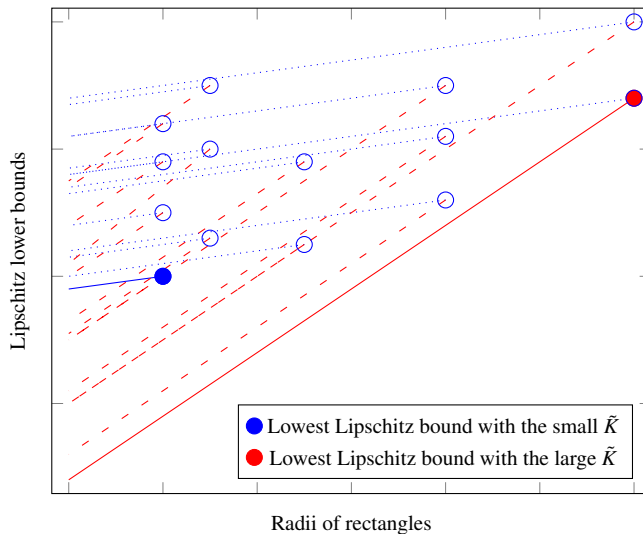


Fig. 1 Visualization of selection of the largest and the smallest Pareto optimal hyper-rectangles using PLOR

A hyper-rectangle is potentially optimal if the lower Lipschitz bound for the objective function over it computed by the left side of (2) is the smallest one with some positive constant \tilde{K} compared to that of all current hyper-rectangles. The purpose of an additional heuristic rule (3) based on the parameter ε is to guard the DIRECT algorithm against excessive emphasis on local search [15]. In the original DIRECT [15] the authors report that the values of $\varepsilon \in [10^{-7}, 10^{-3}]$ work well. Later [14] the author revised the right-hand-side of (3) to include a lower bound, i.e.

$$f(c_j) - \tilde{K}d_j \leq f_{min} - \max\{\varepsilon|f_{min}|, 10^{-8}\}. \quad (4)$$

In contrast, a recently proposed Pareto-Lipschitzian approach (PLO) [28] identifies hyper-rectangles which are non-dominated (Pareto optimal) with respect to an objective vector of infinite dimension composed of Lipschitz bounds corresponding to all positive \tilde{K} . In terms of vector optimization theory, the algorithms inspired by DIRECT select hyper-rectangles belonging to a subset of Pareto optimal set. The new concept is that the Pareto optimality is defined regarding all the Lipschitz functions as an objective vector of continuous dimension. The scientific advantage of the new approach is the strict theoretical background without any user-defined parameters. The practical significance is the obtained larger number of Pareto optimal hyper-rectangles, which can be explored in parallel. However, the results revealed that PLO sometimes wastes a lot of time improving a local minimum and delaying the discovery of the global one [28] similar to original DIRECT.

The reduced objective vector corresponding to two objectives was proposed in [29]: the Lipschitz bound with the smallest possible positive \tilde{K} and the Lipschitz bound with the largest possible \tilde{K} . The use of such objectives corresponds to selection of only the largest and the smallest Pareto optimal hyper-rectangles (see Fig. 1). The resulting Reduced-set Pareto-Lipschitzian Optimization (PLOR) algorithm balances

equally local and global search during the optimization process. In the present paper we develop the idea further, investigate the algorithm and apply it to truss optimization.

The paper is organized as follows. In Sect. 2 we present an overview of modifications and applications of the DIRECT algorithm. In Sect. 3 we describe the PLOR algorithm. In Sect. 4-5 we compare the PLOR algorithm with the original DIRECT and its well-know modifications on standard global optimization test problems [15] and on real truss optimization problems. Finally, in Sect. 6 we give the conclusions.

2 Modifications and applications of the DIRECT algorithm

Due to its simplicity and efficiency (especially for low-dimensional global optimization problems), the DIRECT algorithm has gained popularity in the optimization community. There are numerous applications of DIRECT in various fields. In many applications DIRECT is the sole method [1, 13, 39, 40], in some others DIRECT is applied with other methods of global and local search [2, 4, 24].

A locally biased version of DIRECT, named DIRECT-1 was introduced and analyzed in [11]. In the original DIRECT algorithm, the size of a hyper-rectangle (d) is measured by the Euclidean distance from its center to a corner. In DIRECT-1, the size of a hyper-rectangle is instead measured by the length of its longest side. Such a measure corresponds to the infinity norm and allows the algorithm to group more hyper-rectangles with the same size. Secondly, in DIRECT-1 at most one hyper-rectangle from each group is subdivided, even if there are more than one potentially optimal hyper-rectangle in some groups. This allows reduction of the number of divisions within a group. The results presented in [11] suggest, that DIRECT-1 should be used for lower dimensional problems, which do not have too many local and global minima. The results also demonstrate the effects of the modifications in DIRECT-1. The convergence occurs in a fewer function evaluations, but can take more iterations. This is due to a fewer hyper-rectangles being chosen in each iteration.

In [9], numerical experiments were presented to show that the original DIRECT algorithm is sensitive to the additive scaling of the objective function. It is also shown, that DIRECT does not perform well when the values of the objective function are large enough. The modified DIRECT-m algorithm was proposed to eliminate this sensitivity. The authors suggest to scale the function values after each iteration by subtracting f_{median} – the median of the collected function values. The result is an update to (3):

$$f(c_j) - \tilde{K}d_j \leq f_{min} - \varepsilon|f_{min} - f_{median}|. \quad (5)$$

As shown in [41] DIRECT is not strongly homogeneous algorithm. In [22] the properties of DIRECT related to the scaling of the objective function is investigated once again. The author states that DIRECT-m or DIRECT-a [22] must be used in order to eliminate sensitivity to the additive and linear scaling of the objective function. In DIRECT-a Definition 1 of the potentially optimal hyper-rectangle was changed similarly to DIRECT-m:

$$f(c_j) - \tilde{K}d_j \leq f_{min} - \varepsilon|f_{min} - f_{aver}|, \quad (6)$$

here f_{aver} is the average of $\{f|Q_1 \leq f \leq Q_3\}$, where Q_1 and Q_3 are two quartiles of all function evaluations.

It is well-known that the DIRECT global optimization algorithm can quickly get close to the basin of the minimizer, but takes longer to achieve a high degree of accuracy [14]. A common strategy to eliminate such defect is to combine DIRECT with a good local optimization algorithm. Jones [14] suggests to use DIRECT as a starting point generator and after some initial budget of function evaluations for DIRECT switch to local optimization. After convergence of the local optimization it switches back to DIRECT. By using the best function value found by either DIRECT or the local optimization, DIRECT will search more globally. This is because the value of f_{min} (see (3)) affects which hyper-rectangles are potentially optimal. If DIRECT finds a better point, it again switches to the local optimization, and so on.

Similar idea is implemented in [24] as a new DIRMIN algorithm for solving difficult large-scale global optimization problems. A multi-start inspired DIRMIN algorithm uses DIRECT as a deterministic generator of the starting points and employs a local optimization to improve the produced estimate of the global minimum. Unlike [14], DIRMIN performs a local optimization starting from centroids of each potentially optimal hyper-rectangle. Even this strategy may still not suffice to solve large-scale global optimization problems (e.g. optimization of Morse potential of molecular clusters). In order to tackle these problems, a new algorithm called DIRMIN-TL, that consists in repeatedly applying DIRMIN on modifications of the variable space defined on the basis of the information gained up to that point.

In [4], the authors use DIRECT as a starting point generator combined with the local optimization for solving the problem of design of the high-speed civil transport (HSCT). They stop DIRECT and switch to local optimization once one of the hyper-rectangles containing f_{min} becomes a given small size.

It was shown in [2], that the DIRECT/DIRECT-1 algorithm combined with Implicit Filtering method (IFFCO) [3] – DIRECT-1-IFFCO is the most robust method for three problems from the gas pipeline industry, while at the same time it uses a smaller number of function evaluations. Note that DIRECT and DIRECT-1 alone were also able to find a solution to the tested problem. However they needed more function evaluations, sometimes substantially more, than with DIRECT-1-IFFCO or DIRECT-IFFCO.

Because the local optimization is often very sensitive to the initial point, this strategy is often highly sensitive to the number of function evaluations consumed in the global phase. A bi-level strategy is introduced into a modified DIRECT-b algorithm to overcome this shortcoming in [23]. Numerical results show that the bi-level strategy improves the ability of DIRECT to search for solutions with a high accuracy. A useful property of the bi-level strategy is that such an approach can accelerate the convergence in the framework of DIRECT itself.

DIRECT ideas were also generalized in the framework of diagonal partitions [37], space-filling curves [18] and derivatives [16, 17].

The typical DIRECT partition is by dividing rectangles. However, simplex partitions were applied in a similar algorithm called DISIMPL [32–34, 42]. Experiments showed, that the proposed simplicial DISIMPL algorithm gives very competitive results to the DIRECT algorithm using standard test problems and performs particularly well when the search space and the numbers of local and global minimizers may be

reduced by taking into account symmetries of the objective function. Even comparing with the DIRECT version for symmetric functions (SymDIRECT) proposed in [12], both simplicial DISIMPL versions give significantly better results.

It is also well-known, that DIRECT-type global optimization algorithms often spend an excessive number of function evaluations on problems with many local optima exploring suboptimal local minima, thereby delaying discovery of the global minimum. In order to address this problem, a globally-biased version Gb-DISIMPL [31] with an adaptive balancing of local and global information during the search have been introduced, implemented and experimentally investigated. Extensive numerical experiments executed on 800 multidimensional multiextremal test functions had showed a promising performance of the new acceleration technique with respect to competitors.

3 Reduced-set Pareto-Lipschitzian optimization

Since PLOR (Pareto-Lipschitzian Optimization with Reduced-set) is a reduced version of PLO and the theoretical justification is an important issue of both PLO and PLOR, we present a mathematical formulation of these algorithms following the lines of the PLO and PLOR descriptions in [28,29] with the corresponding changes.

The concept of Pareto optimality (see, e.g., [6,27,30]) is traditionally used in the cases where the objective is a vector-function $F(z) = \{F_\omega(z) : \omega \in \Omega\}$. Here z is the decision, ω is a component index of the objective-vector $F(z)$, and Ω is a set of all components ω .

Let us consider the following multi-objective minimization problem:

$$\min_z F(z).$$

Separate objective components $F_\omega(z)$ may be conflicting – decrease of one may correspond to increase of another. Therefore usually there is no single solution – a decision which would be the best according to all objective components. Because of this Pareto optimal (non-dominated) decisions are sought in multi-objective optimization.

Definition 2 The decision z_i dominates the decision z_j (we denote $z_i \prec z_j$), if

$$\begin{aligned} F_\omega(z_i) &\leq F_\omega(z_j), \text{ for all } \omega \in \Omega \\ F_\omega(z_i) &< F_\omega(z_j), \text{ for at least one } \omega \in \Omega. \end{aligned}$$

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

Definition 3 The decision z^* is called Pareto optimal, if there is no decision z_i dominating it.

Let us now return to the single-objective global optimization of a Lipschitz function, but use the concept of Pareto optimal decision to define potentially optimal hyper-rectangles.

Suppose that at the current step of the partitioning algorithm the feasible region $D = [a, b] \subset \mathbb{R}^n$ is partitioned into hyper-rectangles $z_i = [a_i, b_i] \subset D$, $i \in I$. As for the basic DIRECT algorithm, c_i denotes the center of the hyper-rectangle z_i , and d_i denotes the distance from the center point to the vertices (radius). The value of the objective function f is known at the centers of hyper-rectangles: $f(c_i)$.

The lower bound for the objective function over the hyper-rectangle c_i based on Lipschitz condition with a constant ω can be computed as

$$L_\omega(z_i) = f(c_i) - \omega d_i. \quad (7)$$

Expression (7) shows that the lower bound over the hyper-rectangle z_i is smaller for a smaller $f(c_i)$ or a larger d_i for all possible values of the Lipschitz constant ω . We do not assume that the actual Lipschitz constant is known and use the whole possible set of positive constants. Let us compare the “quality” of different hyper-rectangles by their lower bounds. For example, we say that the hyper-rectangle z_i is better than the hyper-rectangle z_j for a given ω , if its lower bound is lower: $L_\omega(z_i) < L_\omega(z_j)$. The unknown Lipschitz constants are regarded as different components of multi-criteria comparison based on lower bounds L_ω .

Definition 4 The hyper-rectangle z_i , $i \in I$, dominates the hyper-rectangle z_j , $j \in I$ (we denote $z_i \prec z_j$), if

$$\begin{aligned} L_\omega(z_i) &\leq L_\omega(z_j), \text{ for all } \omega \in \Omega, \\ L_\omega(z_i) &< L_\omega(z_j), \text{ for at least one } \omega \in \Omega. \end{aligned}$$

Definition 5 The hyper-rectangle z_i , $i \in I$, is called Pareto optimal if there is no hyper-rectangle z_j , $j \in I$, dominating z_i .

We denote the set of Pareto optimal hyper-rectangles by I_p . It is the solution to the multi-objective problem

$$\min_{i \in I} L(z_i), \quad (8)$$

where $L(z) = \{L_\omega(z) : \omega \in \Omega\}$ is the vector-function with lower bounds based on various possible constants ω as the components. Pareto-Lipschitzian optimization (PLO) algorithm [28] chooses Pareto optimal hyper-rectangles as potentially optimal in the framework of partitioning inspired by the DIRECT algorithm.

We note that potentially optimal hyper-rectangles defined by Definition 1 belong to the set of Pareto optimal hyper-rectangles. This is because the hyper-rectangle z_i is Pareto optimal if inequality (2) holds. It is optimal with regards to the objective $L_{\tilde{K}}$ and $\omega > \tilde{K}$ can be chosen so that $L_\omega(z_i) < L_\omega(z_j)$ for $d_j < d_i$ as well as $\omega < \tilde{K}$ can be chosen so that $L_\omega(z_i) < L_\omega(z_j)$ for $d_j > d_i$. Therefore it is not dominated by any z_j .

Let us now build a constructive definition of dominance corresponding to Definition 4. By substituting (7) into Definition 4, inequalities become $f(c_i) - \omega d_i \leq (<) f(c_j) - \omega d_j$ and $f(c_i) - f(c_j) \leq (<) \omega(d_i - d_j)$. If $d_i = d_j$, $L_\omega(z_i) < L_\omega(z_j)$ independently of ω if $f(c_i) < f(c_j)$ and is not otherwise. If $d_i > d_j$, $L_\omega(z_i) < L_\omega(z_j)$ independently of ω if $f(c_i) \leq f(c_j)$, otherwise there is a small positive value of $\omega < (f(c_i) - f(c_j))/(d_i - d_j)$ that $L_\omega(z_i) > L_\omega(z_j)$. If $d_i < d_j$, there is a large

enough value of $\omega > (f(c_i) - f(c_j))/(d_i - d_j)$ that $L_\omega(z_i) > L_\omega(z_j)$. Therefore, $L_\omega(z_i) < L_\omega(z_j)$ independently of ω if

$$d_i = d_j \text{ and } f(c_i) < f(c_j) \text{ or} \quad (9)$$

$$d_i > d_j \text{ and } f(c_i) \leq f(c_j). \quad (10)$$

By splitting (10) into two cases $f(c_i) < f(c_j)$ or $f(c_i) = f(c_j)$ and combining the first one with (9) we get a more general case

$$d_i \geq d_j \text{ and } f(c_i) < f(c_j).$$

In this way we come to the following definition.

Definition 6 The hyper-rectangle $z_i, i \in I$, dominates the hyper-rectangle $z_j, j \in I$ (we denote $z_i \prec z_j$), if

$$d_i \geq d_j \text{ and } f(c_i) < f(c_j) \text{ or } d_i > d_j \text{ and } f(c_i) \leq f(c_j).$$

Comparing this definition to Definition 2 or Definition 4 we can see that such a definition corresponds to a bi-objective problem

$$\min_{i \in I} \{f(c_i), -d_i\}, \quad (11)$$

where we minimize the distance d_i multiplied by -1 in order to have two minimized objectives instead of one minimized and another maximized. Solution to this problem is again the same set of Pareto optimal hyper-rectangles I_P .

The number of Pareto optimal hyper-rectangles is larger than potentially optimal defined by Definition 1. This can be exploited in parallel algorithms since the larger number of hyper-rectangles to investigate enables easier balancing of work. However it is also possible that the investigation of Pareto optimal hyper-rectangles may be slower [28] since more time is spent for middle-sized hyper-rectangles instead of improving a local minimum by investigating the hyper-rectangles around the best point found and exploring the whole feasible region by investigating the largest hyper-rectangles.

Therefore, the idea of a reduced subset of Pareto optimal hyper-rectangles is proposed in [29]. The subset corresponds to extreme solutions to the problem (11): choose only the largest and the smallest Pareto optimal hyper-rectangles. In other words choose the largest hyper-rectangles for which $f(c_i)$ is minimal and choose the hyper-rectangles with the smallest $f(c_i)$ among the largest hyper-rectangles. The reduced set I_r can be defined by the following conditions:

$$\begin{aligned} i \in I_r \text{ if } & f(c_i) \leq f(c_j) \forall j \in I \text{ and } d_i \geq d_k \forall k : f(c_i) = f(c_k) \\ & \text{or } d_i \geq d_j \forall j \in I \text{ and } f(c_i) \leq f(c_k) \forall k : d_i = d_k. \end{aligned} \quad (12)$$

Figure 2 illustrates expression (12) of PLOR and Definition 1 of DIRECT for the two dimensional Branin test problem in the tenth iteration of the algorithms. Each point on the graphs represents a rectangle, where the horizontal axis represents the distance from the center of the rectangle to one of its corners, and on the vertical axis the value of the function f evaluated at the center of rectangle. The filled red (black, in

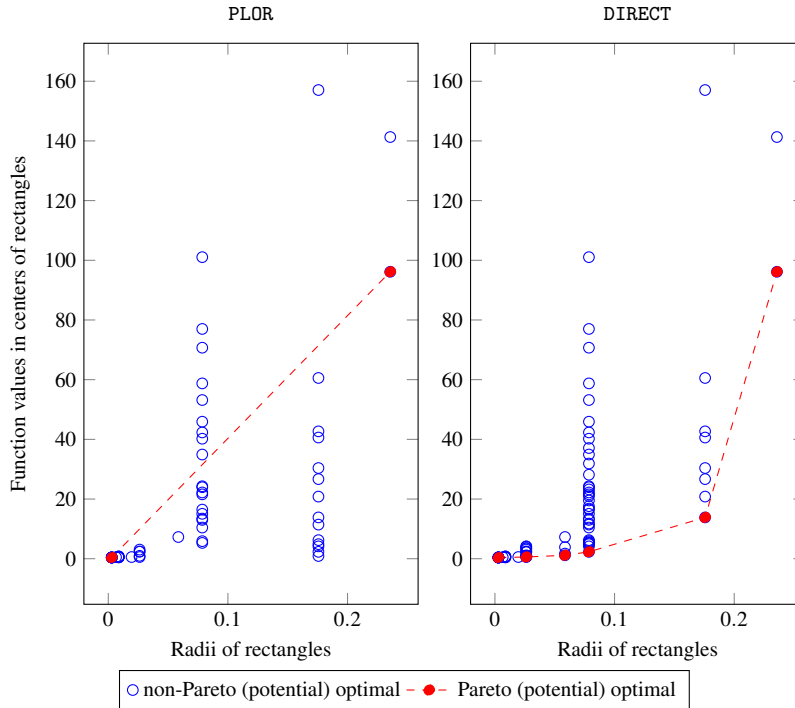


Fig. 2 Visual comparison of Pareto optimal rectangles selected by the PLOR algorithm and potentially optimal rectangles selected by the DIRECT algorithm for the Branin problem in the tenth iteration

black-white) circles represent the intervals selected by the corresponding algorithms which will be divided in the next phase of the iteration of these algorithms.

The subdivision of the potential hyper-rectangles in the PLOR algorithm is the same as in the DIRECT algorithm. Figure 3 illustrates the first four iterations of the PLOR algorithm by a two-dimensional Branin example. Each iteration of the algorithm includes the following tasks:

- The basic task is to make observations (calculations of $f(c_i)$ at fixed c_i). The dots in the figure show the observation points, the accompanying numbers are function values at these points.
- An auxiliary task consists of four parts:
 - Definition of the potentially optimal hyper-rectangles. The yellow colors (different shades, in gray) indicate chosen hyper-rectangle in the current iteration defined by expression (12).
 - Creating new hyper-rectangles by splitting the current hyper-rectangles along the longest dimensions.
 - Defining new observation points in the middle of the new hyper-rectangles.
 - Keeping the best current observation which will be accepted as the solution at the end of the optimization process.

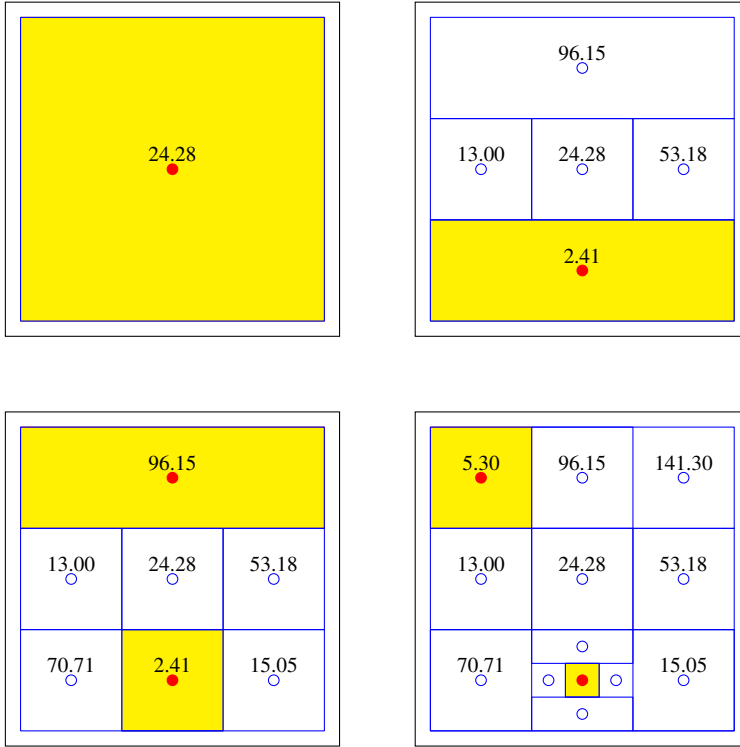


Fig. 3 The first four iterations of partitioning and selection of Pareto optimal rectangles using the PLOR algorithm for two dimensional Branin test problem

- Various stopping conditions can be used: number of iterations, number of function evaluations, and so on. Since the global minima f^* for standard functions are known in advance, the PLOR algorithm stops when the point \bar{x} is sampled such that

$$pe = \begin{cases} \frac{f(\bar{x}) - f^*}{|f^*|} \leq 10^{-4}, f^* \neq 0, \\ f(\bar{x}) \leq 10^{-4}, f^* = 0. \end{cases} \quad (13)$$

For the truss optimization problems the compared algorithms are stopped after some predefined number of function evaluations.

It follows from (12) that the reduced set I_r of potentially optimal hyper-rectangles to be chosen as potentially optimal includes the largest hyper-rectangle. Thus, the largest hyper-rectangle will be divided into three equal parts until the error limit

$$\max_{i=1, \dots, N} d_i \leq \varepsilon_l(N) \quad (14)$$

will be reached. This limit is reached after the finite number of partitions, since at least one of the largest hyper-rectangles is always divided and the resulting hyper-rectangles are smaller. That proves the following proposition:

Proposition 1 *For any $\varepsilon_l > 0$ there exists a number N_ε such that $\varepsilon_l(N) \leq \varepsilon_l$, if $N \geq N_\varepsilon$.*

4 Numerical convergence analysis of the PLOR algorithm

It is well known, that the convergence rate of DIRECT highly depends on the objective function and geometry of the problem. Speed of DIRECT when solving optimization problems with a linear objective function was studied in [10] and extended (including convex quadratic functions) in [8]. It was concluded that there are many factors which can determine how quickly DIRECT converges [8].

In this section, we perform a similar numerical convergence analysis of the PLOR algorithm and compare it with that of other DIRECT-type algorithms on two-dimensional linear and convex quadratic functions used in [8]:

$$\min f(x) = 4x_1 + 5x_2, \quad (\text{linear function})$$

over the domain $D = [0, 1]^2$ with the global minimum $f^* = 0$ at the point $x^* = (0, 0)$ and

$$\min f(x) = x_1^2 + x_2^2, \quad (\text{convex function})$$

over the domain $D = [-4.2, 1.3] \times [-2.1, 5.2]$ with the same optimal value $f^* = 0$ and the global minimum point $x^* = (0, 0)$.

We define the error at the k th iteration of the used algorithm in the same way as in [8]:

$$e_k = \min_{x_k \in D} \|x_k - X^*\|,$$

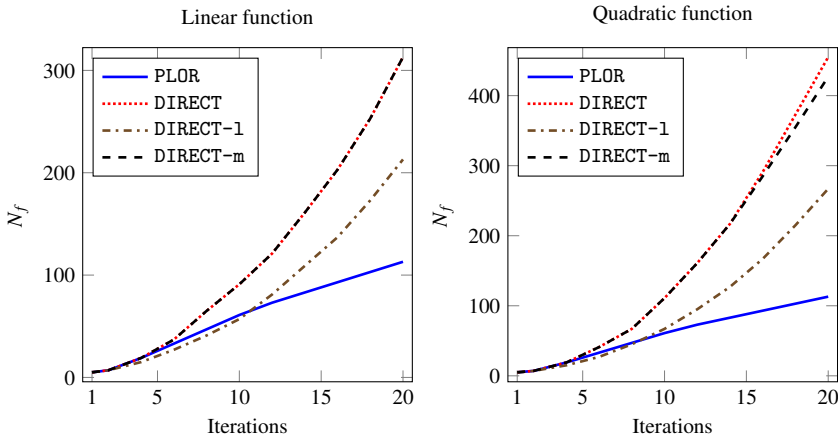
where

$$X^* = \arg \min_{x \in D} f(x) = \{x^* | f(x^*) = f^*\}.$$

Table 1 and Fig. 4 presents the results of investigation based on the number of function evaluations (N_f) and the error (e_k) for all used DIRECT-type algorithms on both problems with the balance parameter set to $\varepsilon = 0$. The results reveal that for these two basic examples PLOR algorithm was able to achieve the same error accuracy from the optimal solution point by using the lowest number of function evaluations. The main reason is that the same error accuracy is achieved by all algorithms at the same iteration, but at each iteration the PLOR algorithm selects only two potentially optimal hyper-rectangles, when all other DIRECT-type algorithms select more. Only in the first iterations DIRECT-1 performs less computations, but later becomes slower than PLOR. Figures 5-6 visualize sampled points by the PLOR and DIRECT algorithms on these two problems during the first ten iterations. It can be seen that the DIRECT algorithm samples wider and more evenly while the PLOR algorithm investigates better the region around the minimum point.

Table 1 The number of function evaluations (N_f) and the error (e_k) after a number of iterations (k)

	PLOR		DIRECT		DIRECT-1		DIRECT-m		
	N_f	e_k	N_f	e_k	N_f	e_k	N_f	e_k	
Linear function	2	7	0.235702	7	0.235702	7	0.235702	7	0.078567
	4	19	0.078567	19	0.078567	15	0.078567	19	0.078567
	6	33	0.026189	37	0.026189	27	0.026189	37	0.026189
	8	47	0.008730	65	0.008730	41	0.008730	65	0.008730
	10	61	0.002910	91	0.002910	57	0.002910	91	0.002910
	12	73	0.000970	121	0.000970	81	0.000970	121	0.000970
	14	83	0.000323	161	0.000323	109	0.000323	161	0.000323
	16	93	0.000108	203	0.000108	137	0.000108	203	0.000108
	18	103	0.000036	253	0.000036	173	0.000036	253	0.000036
	20	113	0.000012	313	0.000012	213	0.000012	313	0.000012
Quadratic function	2	7	0.962924	7	0.962924	7	0.962924	7	0.962924
	4	19	0.238953	19	0.238953	15	0.238953	19	0.238953
	6	33	0.076129	41	0.076129	27	0.076129	41	0.076129
	8	47	0.030000	67	0.030000	45	0.030000	67	0.030000
	10	61	0.012225	111	0.012225	67	0.012225	111	0.012225
	12	73	0.002568	161	0.002568	95	0.002568	161	0.002568
	14	83	0.001620	217	0.001620	127	0.001620	217	0.001620
	16	93	0.000256	291	0.000256	167	0.000256	285	0.000256
	18	103	0.000108	373	0.000108	215	0.000108	355	0.000108
	20	113	0.000050	455	0.000050	267	0.000050	427	0.000050

**Fig. 4** The number of function evaluations after a number of iterations for linear and convex quadratic functions

5 Numerical experiments

In this section, the performance of the PLOR algorithm is investigated on standard global optimization test problems from [15] and on four truss optimization problems commonly used in the literature [19, 20, 38]. For testing of the DIRECT algorithm we

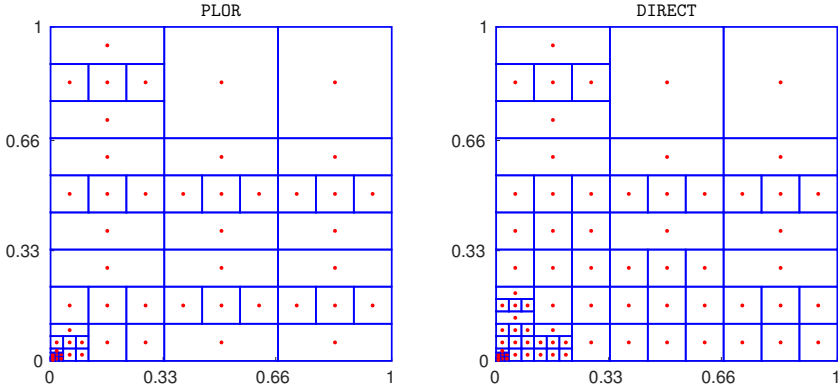


Fig. 5 Trial points generated by PLOR and DIRECT in the first 10 iterations when solving a two dimensional linear problem

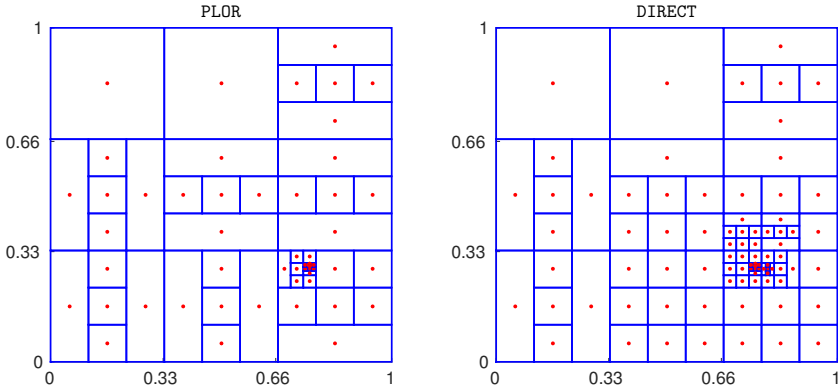


Fig. 6 Trial points generated by PLOR and DIRECT in the first 10 iterations when solving a two dimensional convex quadratic problem

used Finkel implementation in MATLAB [7]. The proposed PLOR algorithm along with the other DIRECT modifications were implemented also in MATLAB.

5.1 Optimization of traditional test functions

First, the performance of the PLOR algorithm is compared with the original DIRECT algorithm and two well-known modifications: DIRECT-1 [11] and DIRECT-m [9]. The criterion of the performance is defined as the number of function evaluations until the algorithm stops generating a trial point \bar{x} such that inequality (13) is satisfied. A complete list of test problems with the dimension (n), the feasible region (D), the numbers of local and global minimizers, and the global minimum is reported in Table 2. This set of optimization problems was used to test the performance of the original DIRECT algorithm [15]. Therefore, the same set is traditionally used as a

Table 2 Description of test problems

Problem	n	D	No. of local minimizers	No. of global minimizers	Global minimum
Branin	2	$[-5, 10] \times [0, 15]$	3	3	0.398
Goldstein-Price	2	$[-2, 2]^2$	4	1	3.000
S-H. Camel B.	2	$[-3, 3] \times [-2, 2]$	6	2	-1.032
Shubert	2	$[-10, 10]^2$	760	18	-186.831
Hartman-3	3	$[0, 1]^3$	4	1	-3.863
Shekel-5	4	$[0, 10]^4$	5	1	-10.153
Shekel-7	4	$[0, 10]^4$	7	1	-10.403
Shekel-10	4	$[0, 10]^4$	10	1	-10.536
Hartman-6	6	$[0, 1]^6$	4	1	-3.322
Shekel-5, variant-1	4	$[0, 5]^4$	5	1	-10.153
Shekel-5, variant-2	4	$[0, 6]^4$	5	1	-10.153
Shekel-5, variant-3	4	$[0, 7]^4$	5	1	-10.153
Shekel-5, variant-4	4	$[0, 8]^4$	5	1	-10.153
Shekel-5, variant-5	4	$[0, 9]^4$	5	1	-10.153

benchmark in a lot of papers devoted to the DIRECT modifications [9, 11, 23–25, 31, 34, 37].

The obtained results presented in Table 3 reveal that the PLOR algorithm is very competitive with the original DIRECT and two tested DIRECT modifications. PLOR performed better solving 6 traditional test problems out of 9 with no help from adjustable parameter ε which is used in other DIRECT-type algorithms. For the S-H. Camel B. and Hartman-6 test problems PLOR performs close to the best results obtained with the DIRECT-1 and only for the Shekel-5 problem PLOR took much higher amount of function evaluations comparing with the other DIRECT-type algorithms. However, it was shown in [37], that for these standard test functions DIRECT executes a very small number of trials until it generates a point in the neighborhood of the global minimizer. Moreover, the initial feasible region can also have a significant impact on this. Therefore, for further investigation, we took five variants of the Shekel-5 test problem with the reduced feasible region (see Table 2). Generally, you would not expect a completely different behavior on such reduced functions. However, first we observe that for all five variants of the Shekel-5 problem (see Table 3) PLOR gives the best or almost the best results. Moreover, for the variant-2 of Shekel-5 problem DIRECT and DIRECT-m methods fails to solve the problem after the maximum budget of function evaluations (equal to 100,000) was reached.

In Table 4 we present results on the same test problems depending on different pe values (13). From these results we can see that the PLOR strategy is promising not only locating the global solution (which is the primary goal of global optimization algorithms), but also to converging to these solutions with a high accuracy. The results reveal that for almost all these standard problems the PLOR algorithm converges faster than the DIRECT and DIRECT-1 algorithms. However, the most promising strategy to obtain solutions with a very high degree of accuracy is to use DIRECT-m or DIRECT-b [23].

Table 3 Comparison (based on the number of fuction evaluations) of PLOR, DIRECT, DIRECT-1 and DIRECT-m on the test problems

Problem	PLOR	DIRECT	DIRECT-1	DIRECT-m
Branin	85	195	119	225
Goldstein-Price	85	191	115	191
S-H. Camel B.	269	285	191	285
Shubert	1,641	2,967	2,043	3,663
Hartman-3	111	199	111	199
Shekel-5	6,857	155	147	155
Shekel-7	133	145	141	145
Shekel-10	133	145	139	145
Hartman-6	311	571	295	571
Shekel-5, variant-1	157	189	181	189
Shekel-5, variant-2	195	> 100,000	245	> 100,000
Shekel-5, variant-3	149	177	175	177
Shekel-5, variant-4	17	9	9	9
Shekel-5, variant-5	195	233	239	233

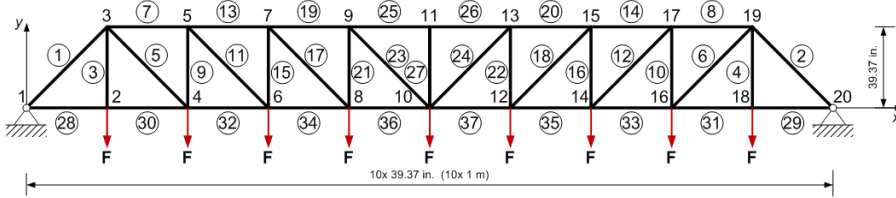


Fig. 7 The initial configuration of the structure

5.2 2D sizing and shape optimization: 37-bar truss

The planar truss of 37 bars [38] is investigated as an approximation of a simply supported bridge. The initial configuration of the structure is shown in Fig. 7. In the optimization process, nodes (3,5,7,9,11,13,15,17,19) of the upper chord can be shifted vertically, while nodes (2,4,6,8,10,12,14,16,18) of the lower chord remain fixed. The loads of free nodes of the lower chord are $F = 2.2481$ kips (10 kN). The Youngs modulus is $E = 30.4579 \times 10^6$ psi (210 GPa), the material density is $\rho = 0.28179$ lb/in³ (7800 kg/m³) for all elements and the allowable stresses are restricted to $[-34.809, 34.809]$ ksi ($[-240, 240]$ MPa).

The topology of the structure is fixed. The nodal coordinates $Y_j, j = 3, 5, \dots, 19$ and the bar cross-sectional areas $A_i, i = 1, \dots, 37$ are the optimization variables. The nodal coordinates and member areas are linked to maintain the structural symmetry. Thus, the vector of variables is reduced to the following 24 variables $Y_3, Y_5, Y_7, Y_9, Y_{11}, A_1, A_3, A_5, A_7, A_9, A_{11}, A_{13}, A_{15}, A_{17}, A_{19}, A_{21}, A_{23}, A_{25}, A_{27}, A_{28}, A_{30}, A_{32}, A_{34}, A_{36}$, where 0.0775 (in)² $\leq A_i \leq 1.9375$ (in)², 9.8425 (in) $\leq Y_j \leq 127.9525$ (in). The optimal shape of a truss structure implies that all the specified nodal displacements and bar stresses satisfy the prescribed constraints while the mass W is minimal. In

Table 4 Comparison (based on the number of fuction evaluations) of PLOR, DIRECT, DIRECT-1 and DIRECT-m on the test problems depending on pe

Problem	algorithm	pe				
		0.01	0.001	0.0001	0.00001	0.000001
Branin	PLOR	85	115	159	159	4,719
	DIRECT	195	345	601	827	38,585
	DIRECT-1	119	227	317	317	36,717
	DIRECT-m	225	383	641	1,191	5,095
Goldstein-Price	PLOR	85	95	105	503	3,653
	DIRECT	191	241	305	1,479	10,437
	DIRECT-1	115	153	221	1,171	9,741
	DIRECT-m	191	321	535	1,113	3,395
S-H. Camel B.	PLOR	269	269	497	1,249	18,421
	DIRECT	285	285	933	3,009	46,729
	DIRECT-1	191	191	413	2,611	22,419
	DIRECT-m	285	285	505	777	2,173
Shubert	PLOR	1,641	1,725	2,357	6,797	45,093
	DIRECT	2,967	3,143	3,867	15,915	68,667
	DIRECT-1	2,043	2,095	2,445	12,699	36,997
	DIRECT-m	3,663	3,839	4,407	6,671	10,955
Hartman-3	PLOR	111	9,391	21,917	26,507	26,507
	DIRECT	199	699	4,201	88,633	88,633
	DIRECT-1	111	387	2,969	93,309	93,309
	DIRECT-m	199	757	907	1,897	1,897
Shekel-5	PLOR	6,857	6,929	6,929	78,527	>100,000
	DIRECT	155	255	255	53,525	>100,000
	DIRECT-1	147	237	237	85,947	>100,000
	DIRECT-m	155	255	255	513	777
Shekel-7	PLOR	133	713	4,553	36,635	>100,000
	DIRECT	145	1,061	4,879	38,167	>100,000
	DIRECT-1	141	1,061	1,197	91,951	>100,000
	DIRECT-m	145	255	331	373	949
Shekel-10	PLOR	133	711	4,709	>100,000	>100,000
	DIRECT	145	1,131	4,939	>100,000	>100,000
	DIRECT-1	139	1,065	1,201	>100,000	>100,000
	DIRECT-m	145	255	331	565	1,019
Hartman-6	PLOR	311	503	71,039	>100,000	>100,000
	DIRECT	571	1,031	>100,000	>100,000	>100,000
	DIRECT-1	295	691	>100,000	>100,000	>100,000
	DIRECT-m	571	915	1,191	1,681	4,293

this paper, we minimize the function:

$$W(A, Y) = \rho \sum_{i=1}^n L_i A_i, \quad (15)$$

and the stress restrictions are

$$|\sigma_i| \leq \sigma_{max}, \quad i = 1, \dots, 37. \quad (16)$$

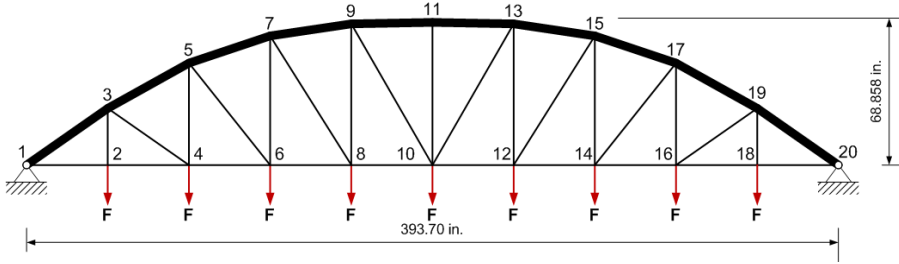


Fig. 8 The optimum shape of the bridge

Tension and compression stresses are treated equally, i.e. local stability is not taken into account. The cross-sections are restricted by the condition

$$A_i^{\min} \leq A_i \leq A_i^{\max}, \quad i = 1, \dots, 37, \quad (17)$$

the node coordinates are restricted by the following inequalities

$$Y_j^{\min} \leq Y_j \leq Y_j^{\max}, \quad j = 3, 5, \dots, 19. \quad (18)$$

and the limit of the vertical displacement of the node $j = 10$ is as follows

$$|\delta_{10,y}| \leq \delta_{\max}, \quad (19)$$

where $\delta_{\max} = 0.3937$ in (10 mm). The displacement and stress constraints are included using the penalty function k^λ , where $k = 10$ is the penalty parameter and λ is the penalty indicator which is equal to zero if all the constraints are satisfied.

The optimal shape obtained with the PLOR algorithm is shown in Fig. 8 with the respective Y -coordinates of the nodes. Figures 9 and 10 illustrate selection of reduced Pareto optimal (using PLOR) and potentially optimal hyper-rectangles (using DIRECT) for the 37-bar planar truss in 10th iteration. We see that, according to its definition (12), the PLOR algorithm selects just two hyper-rectangles: one with the current minimal function value, the other from the largest radius set. Figure 9 illustrates a case with several hyper-rectangles with the equal largest radius. The hyper-rectangle with the smallest function value is preferred.

A comparison with the other DIRECT-type and with Tang [38] and Im.PSO [21] algorithms is presented in Tables 5 and 6. The total quality of the optimized structure obtained with PLOR algorithm is 114.705 lb, which is less comparing with the optimization results of DIRECT (179.746 lb), DIRECT-1 (150.783 lb), DIRECT-m (172.294 lb), Im.PSO [21] (152.714 lb), and [38] (170.770 lb). Convergence curves of the PLOR and other tested DIRECT-type algorithms: DIRECT, DIRECT-1 and DIRECT-m is illustrated in Fig. 11. The curves show that in the beginning DIRECT-1 algorithm converges faster than the other, but from about 5,000 function evaluations the best current minimum and the lowest converge curve is obtained with the PLOR algorithm. Similar trends were observed regardless of the initial search space. Finally, the constraints are satisfied by all the algorithms (see Table 6).

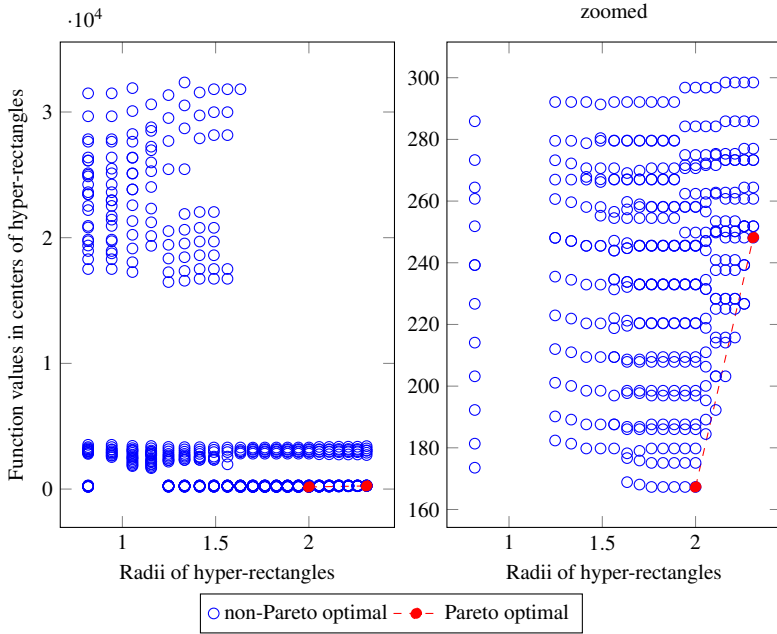


Fig. 9 Geometric visualization of the reduced Pareto optimal hyper-rectangles ($n = 24$) by using the PLOR algorithm for optimizing the 37-bar structure in 10^{th} iteration

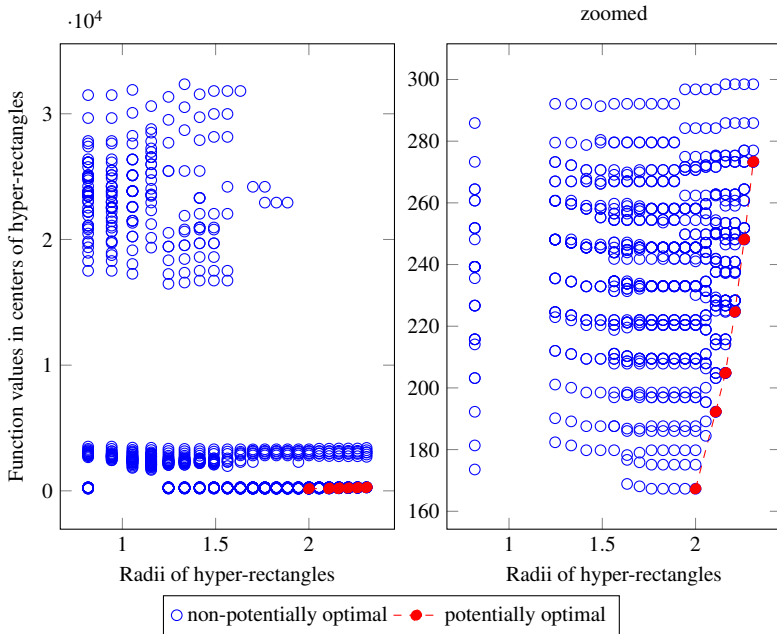


Fig. 10 Geometric visualization of potentially optimal hyper-rectangles ($n = 24$) by using the DIRECT algorithm for optimizing the 37-bar structure in 10^{th} iteration

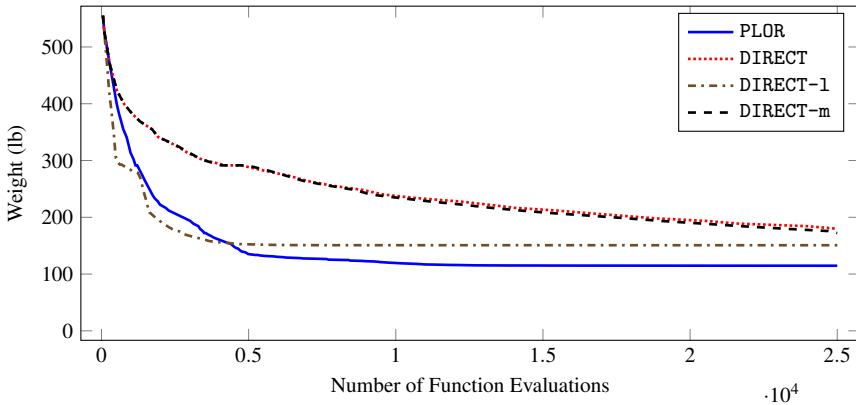


Fig. 11 Convergence curves of the PLOR, DIRECT, DIRECT-1 and DIRECT-m algorithms for the 37-bar structure optimization

5.3 3D sizing optimization

5.3.1 25-bar space truss

The well-known spatial structure – electricity transmission tower composed of 25 truss elements and possessing 10 nodes (Fig. 12) is being optimized:

$$\begin{aligned}
 \min \quad & W(A) = \rho \sum_{i=1}^n L_i A_i \\
 \text{s.t.} \quad & \delta_{\min} \leq \delta_i \leq \delta_{\max} \quad i = 1, \dots, 3m, \\
 & \sigma_{\min} \leq \sigma_j \leq \sigma_{\max}, \\
 & A_{\min} \leq A_i \leq A_{\max} \quad j = 1, \dots, n,
 \end{aligned}$$

for all load cases. Here δ_i are the nodal displacements, σ_j are stresses in the trusses, A_j are cross-sectional areas, m is the number of free nodes, and n is the number of elements.

The construction to be optimized has to meet the displacements and stress constraints for both load cases. The maximum displacements in all nodes in all directions must not exceed ± 0.35 in. The allowable tensile stresses in elements are 40 ksi, while the compression stresses are shown in Table 7. Material density is $\rho = 0.1$ lb/in³, and the Young's modulus is $E = 10^4$ ksi. The optimized construction is symmetrical, therefore all the 25 trusses are subdivided into 8 groups; their cross-sectional areas form the vector of design parameters A . The construction is analyzed for 2 load cases (Table 8). The values of truss cross-section areas are continual, the minimum value being 0.01 in².

Table 9 presents a comparison of the optimal solutions obtained with the four DIRECT-type algorithms, also with the Schmit method [36] and three other (PSO, PSOPC and HIPSO) from the paper [20]. In [20] a population of 50 individuals and maximum number of 3000 iterations was used. For this spatial truss structure, it takes

Table 5 Comparison of algorithms on 37-bar planar truss optimization problem

Var.	Algorithm					
	PLOR	DIRECT	DIRECT-1	DIRECT-m	Im.PSO	Tang
Y_3	27.349	29.528	26.449	29.528	19.547	20.008
Y_5	49.222	55.774	49.216	55.774	35.339	35.606
Y_7	62.345	68.898	62.339	68.898	46.701	46.382
Y_9	68.178	68.898	66.713	68.898	53.953	52.996
Y_{11}	68.880	68.898	67.362	68.898	55.386	53.677
A_1	0.698	0.801	0.698	0.801	0.880	1.350
A_3	0.078	0.388	0.078	0.181	0.246	0.080
A_5	0.078	0.181	0.078	0.181	0.187	0.080
A_7	0.698	0.801	0.698	0.801	1.178	1.269
A_9	0.078	0.181	0.078	0.181	0.093	0.081
A_{11}	0.078	0.181	0.078	0.181	0.087	0.078
A_{13}	0.698	0.801	0.491	0.801	0.875	1.203
A_{15}	0.078	0.181	0.078	0.181	0.136	0.078
A_{17}	0.078	0.181	0.698	0.181	0.078	0.078
A_{19}	0.698	0.801	0.698	0.801	0.876	1.169
A_{21}	0.078	0.181	0.078	0.181	0.153	0.078
A_{23}	0.078	0.181	0.146	0.181	0.105	0.079
A_{25}	0.698	0.801	0.698	0.801	0.927	1.158
A_{27}	0.078	0.388	0.491	0.181	0.318	0.078
A_{28}	0.078	0.181	0.078	0.181	0.101	0.104
A_{30}	0.078	0.181	0.078	0.181	0.078	0.078
A_{32}	0.078	0.181	0.284	0.181	0.081	0.078
A_{34}	0.078	0.181	0.078	0.181	0.079	0.080
A_{36}	0.078	0.181	0.080	0.181	0.084	0.078
u_{10_y}	0.393	0.392	0.394	0.392	0.391	0.318
(lb)	114.705	179.746	150.783	172.294	152.714	170.770

about 1000 iterations for the PSOPC and the PSO algorithms to converge, respectively. However the HPSO algorithm takes only 50 iterations to converge. Moreover, for this problem PSO algorithm did not fully converge when the maximum number of iterations is reached. Therefore to make the comparison as fair as possible all DIRECT-type algorithms have been terminated after $50 \cdot 3,000 = 150,000$ function evaluations.

The optimal weight with the PLOR algorithm is 546.80 lb and this value is very close to the best values from [20]. Notice, that optimal weights obtained with all DIRECT-type algorithms are similar: DIRECT (546.82 lb), DIRECT-1 (549.26 lb), DIRECT-m (546.82 lb). The convergence curves (see Fig. 13) show similar convergence rates for all DIRECT-type algorithms. However in the beginning PLOR and DIRECT-1 algorithms converge slightly faster – as it was the case in the former problem.

Table 6 Comparison of stresses (ksi) in all the bars

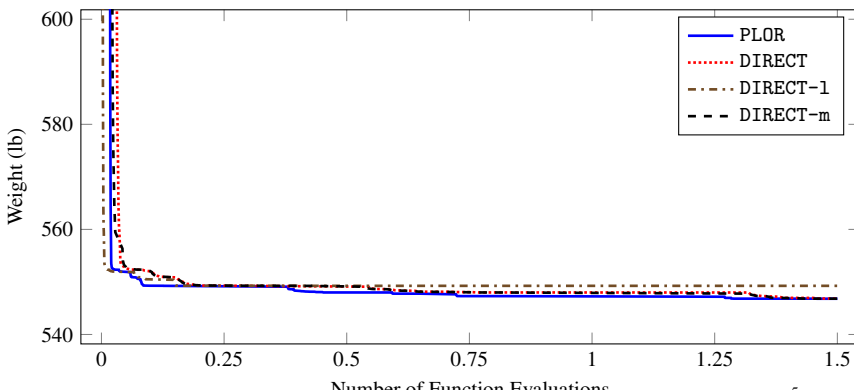
bar number	Algorithm					
	PLOR	DIRECT	DIRECT-1	DIRECT-m	Im.PSO	Tang
1	-25.42	-21.05	-26.01	-21.05	-25.85	-16.54
3	28.95	5.80	28.99	12.43	9.14	27.97
5	-2.79	-5.48	-10.43	-5.48	-2.02	-0.25
7	-23.59	-19.05	-23.83	-19.05	-18.32	-16.86
9	30.54	15.72	34.81	15.72	25.90	28.03
11	10.74	7.61	10.75	7.61	-2.11	2.61
13	-22.52	-17.75	-32.01	-17.75	-23.66	-17.27
15	20.56	6.22	20.59	6.22	17.48	27.11
17	16.23	21.48	2.72	21.48	-4.29	0.08
19	-22.57	-19.25	-22.96	-19.25	-22.84	-17.39
21	15.25	-6.22	8.32	-6.22	16.37	28.91
23	12.47	7.16	6.77	7.16	4.68	12.12
25	-23.03	-20.05	-23.55	-20.05	-21.56	-17.81
27	7.38	0.00	1.10	0.00	4.57	9.10
28	-3.04	-1.25	-1.51	-1.25	3.05	-0.50
30	-3.04	-1.25	-1.51	-1.25	3.98	-0.68
32	-5.33	-5.64	-2.78	-5.64	-0.38	-0.91
34	1.38	-1.25	-3.46	-1.25	-2.11	1.01
36	10.04	9.40	9.25	9.40	-4.54	1.08
Mean	14.99	10.09	14.28	10.44	11.20	11.91
St.dev.	9.58	7.62	11.57	7.56	9.47	10.96

Table 7 25-bar truss. Allowable stresses

No. of truss element	1	2-5	6-9	10-11	12-13	14-17	18-21	22-25
Allow. compressive stresses (ksi)	35.092	11.590	17.305	35.092	35.092	6.759	6.959	11.082
Allowable tensile stresses (ksi)	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0

Table 8 25-bar truss. Load cases

No. of node	Case 1 (Kips)			Case 2 (Kips)		
	F_x	F_y	F_z	F_x	F_y	F_z
1	0.0	20.0	-5.0	1.0	10.0	-5.0
2	0.0	-20.0	-5.0	0.0	10.0	-5.0
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0



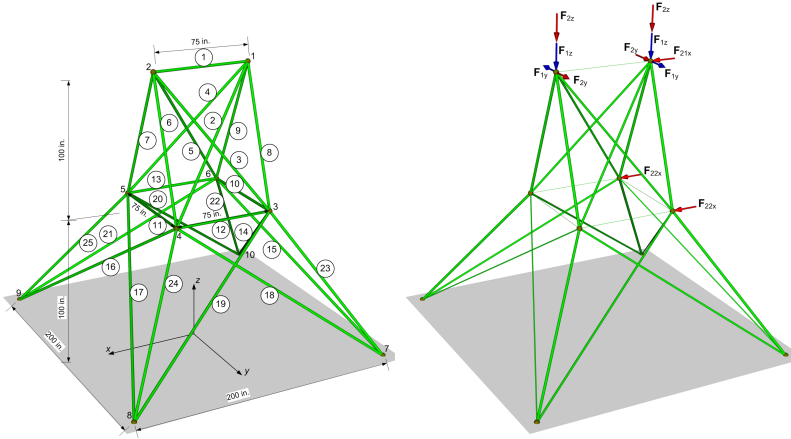


Fig. 12 The initial (on the left) and optimized (on the right) 25-bar spatial truss structure

Table 9 Optimal cross-sectional areas (in²) for the 25-bar truss found using different algorithms

No.	Algorithm							
	PLOR	DIRECT	DIRECT-1	DIRECT-m	Schmit	PSO	PSOPC	HIPSO
1	0.010	0.010	0.050	0.010	0.010	9.863	0.010	0.010
2	1.951	1.950	2.337	1.950	1.964	1.798	1.979	1.970
3	3.025	3.024	2.601	3.024	3.033	3.654	3.011	3.016
4	0.010	0.010	0.010	0.010	0.010	0.100	0.100	0.010
5	0.010	0.012	0.010	0.012	0.010	0.100	0.100	0.010
6	0.592	0.592	0.592	0.592	0.670	0.596	0.657	0.694
7	1.706	1.707	1.611	1.707	1.680	1.659	1.678	1.681
8	2.789	2.789	2.919	2.789	2.670	2.612	2.693	2.643
(lb)	546.80	546.82	549.26	546.82	545.22	627.08	545.27	545.19

5.3.2 72-bar space truss

The 72-bar spatial truss structure (see Fig. 14) was optimized by many researchers. The dimensions, numbers of basic nodes, and trusses are shown in the figure, let us note that a different numbering was used in [35]. The sizing optimization problem is formulated in the same way as for 25-bar truss. The construction is analyzed for 2 load cases (Table 10). The limits for displacements of upper contour nodes (17-20) for loading Case 2, and for the displacements of all free nodes (6-20) in all directions for loading Case 1 are ± 0.25 in. The allowable tensile and compression stresses are ± 25 ksi. Material density is $\rho = 0.1$ lb/in³, Young modulus is $E = 10^4$ ksi. The symmetrical truss is considered, therefore all truss elements are divided into 16 groups.

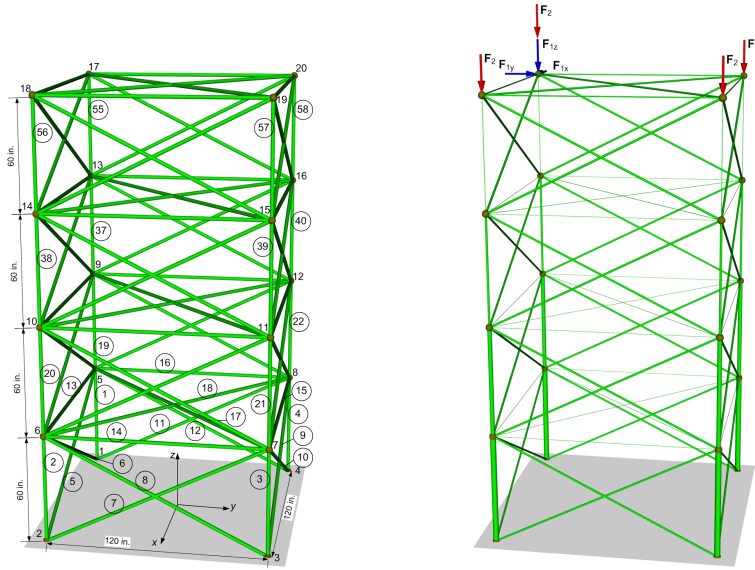


Fig. 14 The initial (on the left) and optimized (on the right) 72-bar space truss

The values of cross-sectional areas are continuous. The allowable minimal cross-section of truss element is 0.1 in^2 .

Table 10 72-bar truss. Load cases

No. of node	Case 1 (Kips)			Case 2 (Kips)		
	F_x	F_y	F_z	F_x	F_y	F_z
17	5.0	5.0	-5.0	0.0	0.0	-5.0
18	0.0	0.0	0.0	0.0	0.0	-5.0
19	0.0	0.0	0.0	0.0	0.0	-5.0
20	0.0	0.0	0.0	0.0	0.0	-5.0

Table 11 presents a comparison of the optimal solutions obtained with the four DIRECT-type algorithms, also with the Schmit method [36], Perez and Behdinan [35], and three (PSO, PSOPC and AugPSO) from the recent paper [26]. As in the 25-bar case, all DIRECT-type algorithms have been terminated after 150,000 function evaluations.

The weights obtained with the PLOR and DIRECT-1 algorithms are better than with other two DIRECT-type algorithms and are close to the best known value obtained with the AugPSO algorithm [26]. The convergence curves (see Fig. 15) again

show better performance for the PLOR and DIRECT-1 algorithms, especially at the beginning.

Table 11 Optimal cross-sectional areas (in²) of the 72-bar truss found using different algorithms

No.	Algorithm								
	PLOR	DIR.	DIR.-1	DIR.-m	Schmit	Perez	PSO	PSOPC	AugPSO
1	1.603	1.598	1.699	1.603	2.078	1.743	1.609	1.239	1.843
2	0.544	0.583	0.476	0.583	0.503	0.519	0.515	0.513	0.500
3	0.100	0.101	0.100	0.100	0.100	0.100	0.888	0.100	0.104
4	0.100	0.101	0.100	0.100	0.100	0.100	1.513	0.426	0.100
5	1.192	1.192	1.371	1.192	1.107	1.308	1.003	1.302	1.221
6	0.566	0.583	0.547	0.583	0.579	0.519	0.382	0.740	0.549
7	0.100	0.101	0.100	0.100	0.100	0.100	0.100	0.100	0.100
8	0.100	0.101	0.100	0.100	0.100	0.100	1.390	1.896	0.100
9	0.605	0.570	0.618	0.571	0.264	0.514	0.560	0.491	0.490
10	0.547	0.547	0.476	0.547	0.548	0.546	0.668	0.558	0.496
11	0.100	0.118	0.100	0.118	0.100	0.100	0.220	0.100	0.103
12	0.122	0.118	0.112	0.118	0.151	0.109	1.842	0.127	0.150
13	0.154	0.154	0.153	0.154	0.158	0.162	0.120	0.157	0.156
14	0.583	0.584	0.582	0.583	0.594	0.509	0.665	0.490	0.575
15	0.315	0.261	0.405	0.261	0.341	0.497	0.527	0.404	0.433
16	0.516	0.512	0.655	0.508	0.608	0.562	0.413	0.950	0.522
(lb)	383.70	386.98	382.34	387.00	388.63	381.91	576.69	472.59	381.62

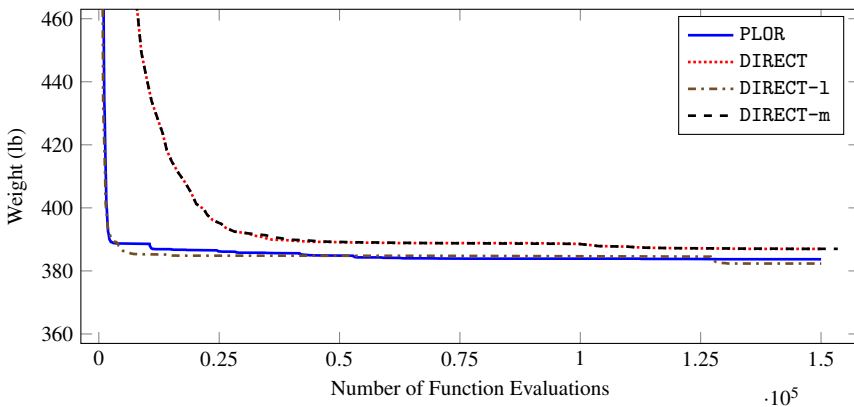


Fig. 15 Convergence curve of the PLOR, DIRECT, DIRECT-1 and DIRECT-m algorithms for the 72-bar truss

5.4 2D sizing and topology optimization: 10 nodes

The 10-node truss [5,19] is optimized starting from the initial node connection scheme “all-to-all” (see Fig. 16). The sizing and topology optimization problem is

considered with elements' cross-section areas as the design parameters. The minimal truss mass is sought. The problem is formulated in the following way:

$$\begin{aligned} \min \quad & W(A, XY, Con) = \rho \sum L_i A'_i \\ \text{s.t.} \quad & \text{C1: Truss is acceptable to the user} \\ & \text{C2: D-o-F} \leq 0, \\ & \text{C3: } \delta_{min} \leq \delta_j \leq \delta_{max}, j = 1, \dots, 2m, \\ & \text{C4: } \sigma_{min} \leq \sigma_i \leq \sigma_{max}, \\ & \text{C5: } A_{min} \leq A_i \leq A_{max} \quad i = 1, \dots, n, \end{aligned}$$

where XY are coordinates of the nodes, Con are connections between nodes, and A'_i are cross-section areas of elements for topologically modified structure:

$$A'_i = \begin{cases} A_i & \text{if } A_i \geq \varepsilon \\ 0 & \text{if } A_i < \varepsilon, \end{cases}$$

where ε is the critical area. The optimized structure has to retain the initial conditions and loadings, and has to be kinematically stable. Also, the structure has to correspond to the constraints on the displacements, stresses, and the cross-section areas. In case of constraints violation a penalty to the objective function value is given as:

$$F_{pen} = \begin{cases} k^\lambda & \text{if C1 or C2 is violated,} \\ W + k^\lambda \cdot \sum p_i & \text{otherwise,} \end{cases}$$

where penalty parameter $k = 10$, penalty indicator $\lambda = 9$ for C1, $\lambda = 8$ for C2, and $\lambda = 5$ for C3-C4, $\sum p_i$ is sum of maximum normalized penalties due to stresses' and displacements' violations. If all constraints are met $\sum p_i = 0$.

The optimization parameters are as follows: loadings on nodes 7, 8, 9 are $F_y = -10^5$ lb (see Fig. 16), Young's modulus $E = 10^4$ ksi, density $\rho = 0.1$ lb/in³. The allowable tensile and compression stresses are ± 25 ksi, maximal allowable displacements in active nodes are 2 in, and allowable cross-section areas are ± 1 in², $\varepsilon = 0.09$ in². The symmetrical structure is expected, therefore all the truss elements are divided into 25 groups; the cross-section areas of the elements comprise the design parameters vector A .

Table 12 presents a comparison of the optimal solutions obtained after N_f iterations with four DIRECT-type algorithms and the mean value with the standard deviation for the SCGA [19].

First, we notice that the optimal weights with all the DIRECT-type algorithms are equal to 62.004 lb. The obtained final structure is shown in Fig. 17. Second, all these weights were found in the early phase of optimization (up to $N_f = 1,000$ function evaluations), but later did not improve with neither DIRECT-type algorithm. This probably can be explained by the previously mentioned fact that the DIRECT-type algorithms often spend an excessive number of function evaluations on problems with many local minima exploring suboptimal local minima, thereby delaying discovery of the global minimum. Even so, the value obtained with the DIRECT-type algorithms is comparable with the average outcome of the SCGA.

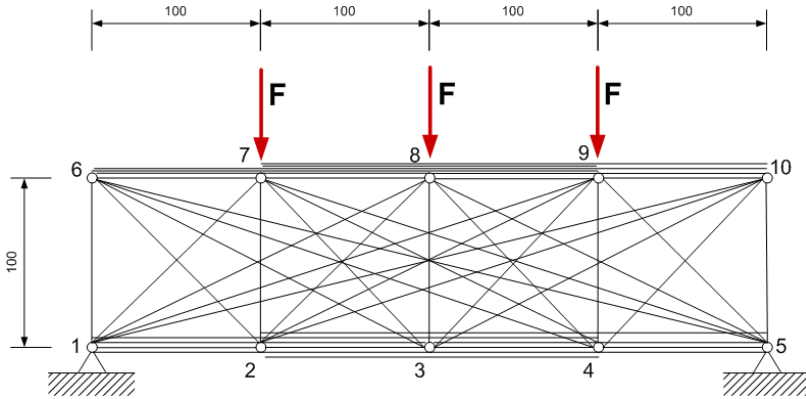


Fig. 16 Initial 10-nodes 45-truss elements structure

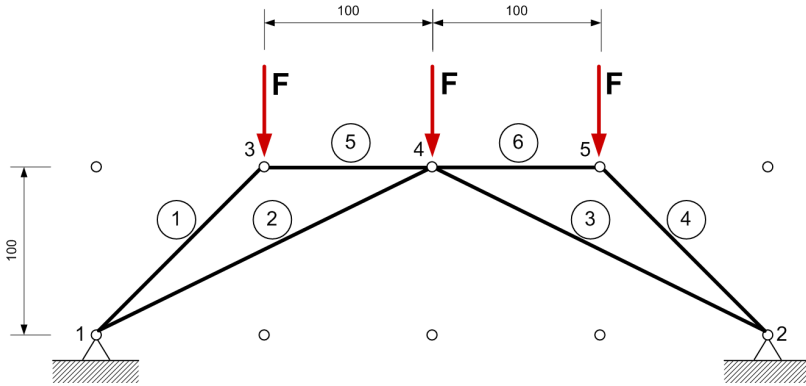


Fig. 17 The final structure obtained from initial 10-nodes

Table 12 Optimal weighs (lb) of the 10 node ground structure after N_f function evaluations

N_f	Algorithm				
	PLOR	DIRECT	DIRECT-1	DIRECT-m	SCGA
1,000	62.004	62.004	62.004	62.004	221.67 ± 36.6698
5,000	62.004	62.004	62.004	62.004	141.15 ± 28.6506
10,000	62.004	62.004	62.004	62.004	105.09 ± 15.8803
20,000	62.004	62.004	62.004	62.004	87.91 ± 14.4118

6 Conclusions

In this paper investigation of the Reduced-set Pareto-Lipschitzian Optimization (PLOR) algorithm has been performed. The advantage of the algorithm is absence of adjustable parameters and trivial selection of a reduced subset of Pareto optimal hyper-rectangles. Experimental investigation has been performed solving traditional test problems for the DIRECT-type algorithms as well as truss optimization problems.

The results of experiments on the standard problems show that the PLOR algorithm performs well on locating the global solutions and converging to these solutions with a high accuracy. The performance of all DIRECT-type algorithms on truss optimization problems is quite similar. They compete well with heuristic algorithms published recently which are often recommended to applied engineering optimization problems. It is worth to note that the PLOR and DIRECT-1 algorithms in large part of experiments converge slightly faster than other algorithms, especially at the beginning. It can be concluded that DIRECT-type algorithms with proved convergence and good experimental performance may be recommended for solution of truss optimization problems.

Acknowledgements This research was funded by a grant (No. MIP-051/2014) from the Research Council of Lithuania.

References

1. Bartholomew-Biggs, M.C., Parkhurst, S.C., Wilson, S.P.: Using DIRECT to solve an aircraft routing problem. *Computational Optimization and Applications* **21**(3), 311–323 (2002). DOI 10.1023/A:1013729320435
2. Carter, R.G., Gablonsky, J.M., Patrick, A., Kelley, C.T., Eslinger, O.J.: Algorithms for noisy problems in gas transmission pipeline optimization. *Optimization and Engineering* **2**(2), 139–157 (2001). DOI 10.1023/A:1013123110266
3. Choi, T.D., Eslinger, O.J., Gilmore, P., Patrick, A., Kelley, C.T., Gablonsky, J.M.: Iffco: Implicit filtering for constrained optimization, version 2. Rep. CRSC-TR99 **23** (1999)
4. Cox, S.E., Haftka, R.T., Baker, C.A., Grossman, B., Mason, W.H., Watson, L.T.: A comparison of global optimization methods for the design of a high-speed civil transport. *Journal of Global Optimization* **21**(4), 415–432 (2001). DOI 10.1023/A:1012782825166
5. Deb, K., Gulati, S.: Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design* **37**(5), 447–465 (2001). DOI 10.1016/S0168-874X(00)00057-3
6. Figueira, J., Greco, S., Ehrgott, M.: *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer, Berlin (2004)
7. Finkel, D.E.: DIRECT optimization algorithm user guide. Tech. rep., Center for Research in Scientific Computation. North Carolina State University, Raleigh, NC 27695-8205 (2003)
8. Finkel, D.E.: Global optimization with the DIRECT algorithm. Ph.D. thesis, North Carolina State University (2005)
9. Finkel, D.E., Kelley, C.T.: Additive scaling and the DIRECT algorithm. *Journal of Global Optimization* **36**, 597–608 (2006)
10. Gablonsky, J.M.: Modifications of the DIRECT algorithm. Ph.D. thesis, North Carolina State University, Raleigh, NC, USA (2001)
11. Gablonsky, J.M., Kelley, C.T.: A locally-biased form of the DIRECT algorithm. *Journal of Global Optimization* **21**, 27–37 (2001)
12. Grbić, R., Nyarko, E.K., Scitovski, R.: A modification of the DIRECT method for Lipschitz global optimization for a symmetric function. *Journal of Global Optimization* **57**(4), 1193–1212 (2013). DOI 10.1007/s10898-012-0020-3
13. He, J., Verstak, A.A., Watson, L.T., Stinson, C.A., Ramakrishnan, N., Shaffer, C.A., Rappaport, T.S., Anderson, C.R., Bae, K.K., Jiang, J., et al.: Globally optimal transmitter placement for indoor wireless communication systems. *Wireless Communications, IEEE Transactions on* **3**(6), 1906–1911 (2004)
14. Jones, D.R.: The DIRECT global optimization algorithm. In: C.A. Floudas, P.M. Pardalos (eds.) *The Encyclopedia of Optimization*, pp. 431–440. Kluwer Academic Publishers, Dordrecht (2001)
15. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application* **79**, 157–181 (1993)
16. Kvasov, D.E., Sergeyev, Y.D.: A univariate global search working with a set of Lipschitz constants for the first derivative. *Optimization Letters* **3**(2), 303–318 (2009). DOI 10.1007/s11590-008-0110-9

17. Kvasov, D.E., Sergeyev, Y.D.: Lipschitz gradients for global optimization in a one-point-based partitioning scheme. *Journal of Computational and Applied Mathematics* **236**(16), 4042–4054 (2012). DOI 10.1016/j.cam.2012.02.020
18. Lera, D., Sergeyev, Y.D.: Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. *Communications in Nonlinear Science and Numerical Simulation* **23**(1-3), 328–342 (2015). DOI 10.1016/j.cnsns.2014.11.015
19. Li, J.P.: Truss topology optimization using an improved species-conserving genetic algorithm. *Engineering Optimization* **47**(1), 107–128 (2015). DOI 10.1080/0305215X.2013.875165
20. Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures* **85**(7), 340–349 (2007)
21. Li, Y., Peng, Y., Zhou, S.: Improved pso algorithm for shape and sizing optimization of truss structure. *Journal of Civil Engineering and Management* **19**(4), 542–549 (2013)
22. Liu, Q.: Linear scaling and the DIRECT algorithm. *Journal of Global Optimization* **56**(3), 1233–1245 (2013). DOI 10.1007/s10898-012-9952-x
23. Liu, Q., Cheng, W.: A modified DIRECT algorithm with bilevel partition. *Journal of Global Optimization* **60**(3), 483–499 (2014). DOI 10.1007/s10898-013-0119-1
24. Liuzzi, G., Lucidi, S., Piccialli, V.: A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Comput Optim Appl.* **45**, 353–375 (2010)
25. Liuzzi, G., Lucidi, S., Piccialli, V.: A partition-based global optimization algorithm. *Journal of Global Optimization* **48**, 113–128 (2010). DOI 10.1007/s10898-009-9515-y
26. Lu, Y.C., Jan, J.C., Hung, S.L., Hung, G.H.: Enhancing particle swarm optimization algorithm using two new strategies for optimizing design of truss structures. *Engineering Optimization* **45**(10), 1251–1271 (2013). DOI 10.1080/0305215X.2012.729054
27. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston (1999)
28. Mockus, J.: On the Pareto Optimality in the Context of Lipschitzian Optimization. *Informatica* **22**(4), 521–536 (2011)
29. Mockus, J., Paulavičius, R.: On the reduced-set Pareto-Lipschitzian optimization. *Computational Science and Techniques* **1**(2), 184–192 (2013)
30. Pardalos, P.M., Siskos, Y. (eds.): *Advances in Multi-criteria Analysis*. Kluwer Academic Publishers (1995)
31. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased DISIMPL algorithm for expensive global optimization. *Journal of Global Optimization* **59**(2–3), 545–567 (2014). DOI 10.1007/s10898-014-0180-4
32. Paulavičius, R., Žilinskas, J.: Advantages of simplicial partitioning for lipschitz optimization problems with linear constraints. *Optimization Letters* (2014). DOI 10.1007/s11590-014-0772-4. In press
33. Paulavičius, R., Žilinskas, J.: *Simplicial Global Optimization*. SpringerBriefs in Optimization. Springer, New York (2014). DOI 10.1007/978-1-4614-9093-7
34. Paulavičius, R., Žilinskas, J.: Simplicial Lipschitz optimization without the Lipschitz constant. *Journal of Global Optimization* **59**(1), 23–40 (2014). DOI 10.1007/s10898-013-0089-3
35. Perez, R., Behdinan, K.: Particle swarm approach for structural design optimization. *Computers & Structures* **85**(19), 1579–1588 (2007)
36. Schmit, L.A., Farshi, B.: Some approximation concepts for structural synthesis. *AIAA journal* **12**(5), 692–699 (1974)
37. Sergeyev, Y.D., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM Journal on Optimization* **16**(3), 910–937 (2006)
38. Tang, H., Li, F., Wang, Y., Xue, S., Cheng, R.: Particle swarm optimization algorithm for shape optimization of truss structures. *Journal of Harbin Institute of Technology* **41**(12), 94–99 (2009)
39. Zhu, H., Bogy, D.B.: DIRECT algorithm and its application to slider air-bearing surface optimization. *Magnetics, IEEE Transactions on* **38**(5), 2168–2170 (2002)
40. Zhu, H., Bogy, D.B.: Hard disc drive air bearing design: modified DIRECT algorithm and its application to slider air bearing surface optimization. *Tribology international* **37**(2), 193–201 (2004)
41. Žilinskas, A.: On strong homogeneity of two global optimization algorithms based on statistical models of multimodal objective functions. *Applied Mathematics and Computation* **218**(16), 8131–8136 (2012). DOI 10.1016/j.amc.2011.07.051
42. Žilinskas, J., Kvasov, D.E., Paulavičius, R., Sergeyev, Y.D.: Acceleration of simplicial-partition-based methods in Lipschitz global optimization. In: V.P. Gergel (ed.) *High-Performance Computing on Clusters*, pp. 128–133. Publishing House of Nizhny Novgorod State University, Nizhny Novgorod, Russia (2013)