On Generalizing Lipschitz Global Methods for Multiobjective Optimization

Alberto Lovison[1(✉)] and Markus E. Hartikainen[2]

[1] Department of Mathematics, University of Padova, Padua, Italy
[2] Department of Mathematical Information Technology, University of Jyvaskyla, P.O. Box 35, FI-40014 Jyvaskyla, Finland
lovison@math.unipd.it

Abstract. Lipschitz global methods for single-objective optimization can represent the optimal solutions with desired accuracy. In this paper, we highlight some directions on how the Lipschitz global methods can be extended as faithfully as possible to multiobjective optimization problems. In particular, we present a multiobjective version of the Pijavskiĭ-Schubert algorithm.

Keywords: Global and Lipschitz optimization · Multiobjective optimization · Multiple criteria decision making

1 Introduction

Exact global search methods are a well known class of algorithms belonging to the single-objective optimization literature.[1] These methods usually demonstrate appreciable speed of convergence and furthermore guarantee that the global optimum of the function under exam is approximated with arbitrary precision in a finite time, providing some constraints on the functions at hand. A well known example of these methods is the Pijavskiĭ-Schubert algorithm [18,19], which is quickly reviewed in Section 2.

Unfortunately, it appears that in the available multiobjective literature there has not been so much attention dedicated to the complete or deterministic methods for global search. Nevertheless, at least in the single objective case and when limited computational resources are available, global deterministic methods have proven their effectiveness and are known and widely employed. Now and then we have witnessed the attempt of producing adaptations of some of these methods for the multiobjective case. However, at least to the knowledge of the authors, most of those adaptations follow one of the following schemes:

1. the method uses a scalarization of the multiobjective problem to a single objective optimization problem and then applies the global algorithm to the scalarization, or

[1] These methods are referred to also as *complete* or *deterministic*, possibly referring to more specific features [16].

A. Gaspar-Cunha et al. (Eds.): EMO 2015, Part II, LNCS 9019, pp. 264–278, 2015.
DOI: 10.1007/978-3-319-15892-1_18

2. the method translates the underlying idea of the global method in the multiobjective format, but then applies a non deterministic method to produce the Pareto set.

In both cases, we encounter the following problems contrasting with the global and exact character desired:

1. the method cannot guarantee a systematic covering of the Pareto set, or
2. the method operates at some point some non deterministic choice.

Well-known and widely used methods belonging to the latter class are the evolutionary multiobjective optimization methods. To partially overcome the first problem some method try to realize a systematic covering of the space of parameters. However, this could not lead to a correspondingly systematic covering of the Pareto set, especially in non convex cases. Therefore we do not consider this approach as genuinely multiobjective and we would prefer to tackle the multiobjective nature of the problem directly. We believe that the best strategy for approximating the Pareto set is adopting a set-wise approach. That is, instead of having a single point converging to a Pareto optimum at a time and then repeating this for a number of points, it is better to make converge multiple points at the same time towards the whole Pareto set. This set-wise concept of convergence is already adopted by evolutionary multiobjective optimization methods, and it is in contrast to point-wise convergence followed by most scalarization methods.

We restate our claim about the nonexistence of exact global methods in a more positive sense, by presenting a pair of methods both attempting to adhere to the most possible extent to deterministic methods and to guarantee a complete representation of the Pareto set, at least requesting some regularity conditions on the functions at hand. The first method [4] uses the Karush-Kuhn-Tucker conditions to write a non negative auxiliary function whose zero set contains the set of Pareto optima. The zero set of such a function is approximated by using an associated ordinary differential equation and suitable iteration schemes obtained from a discretization of it. Set-wise convergence with respect to the Hausdorff distance is obtained if suitable regularity conditions are met. The set obtained is the set of subcritical points, which strictly contains the set of Pareto optima. The approximation obtained consists in a collection of hypercubes which covers the subcritical set. The second method [12] uses a qualitatively similar approach obtained from the Smale's first and second order conditions [20,21]. For the special case of two functions in two variables, it is possible to write a multiobjective extension of the Pijavskiĭ-Shubert algorithm, i.e., it is possible to guarantee the convergence to the Pareto set in global sense with respect to the Hausdorff distance.

In both cases, the set obtained is a strict superset of the set of Pareto optima, corresponding to first order conditions of optimality, i.e., extensions of the notion of critical point for a single function, and furthermore, the application is limited to low dimensional examples. Therefore effective and straightforward approximation methods for the Pareto set are still missing, at least in the authors' knowledge.

In this paper, our main scope is to present a Lipschitz global optimization algorithm for multiple objectives, namely an extension of the Pijavskiĭ-Shubert method which does not make use of auxiliary functions and that approximates the set of global Pareto optima within a desired tolerance measured according to the Hausdorff distance (see Section 2 for details). This method is a first step in the direction proposed in [12] where the possibility of defining exact and global strategies was outlined. This method produces an approximation of the Pareto set consisting in a covering composed by arbitrarily small hypercubes. In perspective, the method can be combined with surface tracing methods to generate a faithful geometric surrogate of the Pareto set, as in the methods [5–7, 11].

2 Pijavskiĭ-Shubert

The Pijavskiĭ-Shubert algorithm [18, 19] (from now on noted as the *P-S algorithm*) is a 1-dimensional globally convergent method assuming that a global Lipschitz constant is known in the domain of the search process. At each step of the process, there is a finite number of points in the domain where the function has been evaluated. Those points are taken as the extrema of a collection of subintervals. For every subinterval a lower bound of the unknown function is determined on the basis of the Lipschitz constant, on the subinterval width and the values of the function at the extrema. The subinterval with the lowest estimate is chosen for further sampling and subdivision, by taking the point where the lower bound is predicted to be located.

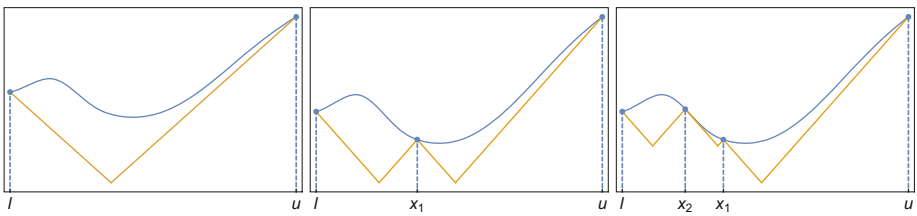

Fig. 1. Workings of the Pijavskĭ-Shubert algorithm. The domain is divided in subintervals and for each subinterval a lower bound is computed on the basis of the global Lipschitz constant for the function in study. The interval with the lowest lower bound is then divided exactly at the position of the lower bound.

This method allows for detecting subintervals where the global minimum cannot be located, discarding them from further analysis. Indeed if the lower bound corresponding to a subinterval is higher than one of the already computed values, it is impossible that the global optimum would be contained into the interval.

This method can be extended to higher dimensions although the computational complexity rises exponentially [15]. Nevertheless the method has been the starting point for several efficient global algorithms, such as DIRECT [8] and Lipschitz Global Optimization (LGO) [17] and many more [9,10][2].

3 Extending P-S Algorithm to Multiple Objectives

In [12], tessellation of the 2D domain by means of equilateral triangles and an auxiliary scalar function was used for deciding if a triangle could contain a portion of the Pareto set or not. However in the method presented here, we will try to define an approach valid for higher dimensions and we will avoid any scalarization or auxiliary function. In particular we will estimate a vector lower bound for every hyper interval, i.e., for each one of the available objective function. This estimate will be on the lines of the scalar method, i.e., based on the Lipschitz constant and on the hyper interval diameter. Then we will not combine the single objective lower bounds in a unique scalar indicator but we will keep the vector as it is and compare and rank different intervals on the basis of Pareto dominance. More precisely, we will partition the set of hyper intervals in two classes, the *discarded* and the *candidates* for further division. To decide if an hyper interval should be discarded we will check if the estimated lower bound dominates one of the already computed points. In that case there cannot exist a point inside the hyper interval belonging to the Pareto set, so we are warranted to discard it. All other hyper intervals will be selected for further division in the subsequent iterations.

This will produce several candidate intervals for each iteration, but this does not constitute a problem, because it is typical for multiobjective methods and it occurs also for some scalar global optimization methods like [8]. A detailed formal description of the algorithm proposed is given in Algorithm 1.

4 Global Convergence of Deterministic Algorithms

We recall some definitions from [12] about global convergence of multiobjective algorithms, starting from the standard scalar case $m = 1$.

4.1 Algorithms and Global Convergence in Scalar Optimization

Let $f : D \to \mathbb{R}$ be a Lipschitz continuous function, where W can be the n-hypercube $[0, 1]^n$ for simplicity or a smooth n-dimensional compact manifold.

[2] In a more general view, we notice that a Lipschitz constant represent a proxy for the complexity of a function. As a result they can be used for optimization as well as for other purposes, e.g., for function approximation (see [14]).

Algorithm 1. multi-Pijavskǐ-Shubert (mPS) method

1: Let the domain (decision space) be a hyper-rectangle $D := [0,1]^n$, possibly after a suitable normalization
2: Let the unknown function $f : D \to \mathbb{R}^m$ be globally Lipschitz continuos with constants different for every component $L_1, \ldots, L_m$.
3: Evaluate f on the corners of D
4: Initialize the set of evaluated points as $E := \left\{ (q, v) \middle| \, q \in \{\text{corners of } D\}, v = f(q) \right\}$
5: Initialize the set of active subintervals as $S := D$
6: Set $nIter \in \mathbb{N}$ as the maximum number of iterations
7: **for** $i = 1$ to $nIter$ **do**
8: **for all** $I \in S$ **do**
9: Divide the hyper interval I in the 2^n subintervals obtained by halving all dimensions of the original hyper interval. Remove the hyper interval I from the list S and add the subintervals to the list.
10: Evaluate f on all the midpoints p of the k-faces of the hyper interval I, for all $1 \leqslant k \leqslant n$, i.e., all the corners of the subintervals. Add $(p, f(p))$ to the list E.
11: **end for**
12: Associate to each interval I in S the vectors $v_{I,\iota} = (f_j(q_\iota) + L_j \text{diam} I)_{\iota,j}$, where q_ι is a corner of I and $j = 1, \ldots, m$.
13: **for all** $I \in S$ **do**
14: **if** all the upper bounds in the corners $(f_j(q_\iota) + L_j \text{diam} I)_j$ are dominated by some vector of values v in E **then**
15: discard the interval I from S
16: **end if**
17: **end for**
18: **end for**

– Let us denote by $f^\star$ the absolute, or *global*, maximum value of the function f, $x^\star$ being a point in W realizing the maximum. In other words, $x^\star$ is an *optimum*, while $X_f^\star$ is the set of all optima:

$$f^\star = \max_{x \in W} f(x), \qquad X_f^\star := \left\{ x^\star \in W \middle| \, f(x^\star) = f^\star \right\}. \tag{1}$$

– An *algorithm* is a finite sequence of well–defined instructions, which, when running on a function f, produces the sample sequence

$$X_f := \{x_1, \ldots, x_k, \ldots\} \subseteq W.$$

In particular the function f is assumed to be actually computed in the point x_k at the kth step of the algorithm.
– We denote by X_f the full infinite sequence produced by an algorithm when given a function f, by $X_{f,k}$ or X_k the partial k–sequence. $\overline{X_f}$ is the closure of X_f while $X_f' = \overline{X_f} \setminus X_f$, is the set of limit points of X_f.

Assume we are not in the trivial case $X_f^\star \cap X_k \neq \emptyset$ for any finite k.

– An algorithm *sees* the global minimum of the function f if $X_f^\star \cap X_f' \neq \emptyset$.

– An algorithm *localizes* the global minimum if $X'_f = X^\star_f$ (or *weakly localizes* if $X'_f \subseteq X^\star_f$).

It seems useful to give further precise description of a class of algorithms for detecting structured subsets rather than scatters of points.

– A *set–wise sequential algorithm* is a deterministic algorithm which, besides the sample sequence $X_f = \{x_1, x_2, \dots\}$ where actually the function f has to be evaluated, generates a sequence of subsets $\{S_1, \dots, S_k, \dots\}$, $S_k \subseteq W$, intended to give an approximation of the Pareto set.
– Notice that more or less explicitly, any multiobjective optimization strategy is a set-wise sequential algorithm. If not specified in a different way, the sequence of sets approximating the Pareto set is given by the non dominated sets of the partial sequences:

$$S_k := nd(\{x1, \dots, x_k\}).$$

– It is a common belief that in typical cases the Pareto set is a finite collection of smooth manifolds with edges and corners. Such objects have interesting properties and are called *stratified sets* (see [13] for a discussion with the point of view of multiobjective optimization). Simplicial methods like [1,6,7,11,12] at each new iteration produce a simplicial complex as approximation of the Pareto optimal set. These methods have the fundamental property of offering a parametric representation of the Pareto set, which appears as a very useful tool for exploring the available solutions during the decision process.

4.2 Convergence in Multiobjective Optimization

To be convergent, an algorithm should produce a sequence $S_1, S_2, \dots$, converging in some sense to the set of optima θ_{op}. Expressing a crude translation of the concepts of *seeing* and *localizing* the optima is poorly useful, because apart from degenerate cases, the set of Pareto optimal values does not consist in a single (vector) value $f^\star \in \mathbb{R}^m$. In the generic case, the set of Pareto optimal values is infinite, as well as, of course, the set of Pareto optima θ_{op}. Limits have to be considered in a set–wise sense, and therefore we need a concept of distance between sets.

– Let $A, B \subseteq W$. The *Hausdorff distance* between A and B is defined as

$$d_{\mathcal{H}}(A, B) := \max \left\{ \max_{x \in A} \min_{y \in B} d(x, y), \max_{y \in B} \min_{x \in A} d(x, y) \right\}. \tag{2}$$

– We say that a set-wise sequential algorithm $\mathcal{A}$ *sees* the set of global Pareto optima θ_{op} if

$$\lim_{k \to \infty} \min_{x \in S_k, y \in \theta_{op}} d(x, y) = 0. \qquad (\mathcal{A} \text{ sees } \theta_{op}) \tag{3}$$

(In a sense the limit set $\lim_k S_k \cap \theta_{op} \neq \emptyset$, i.e., at least the Pareto set generated by the algorithm touches a portion of the global Pareto set, i.e., it generalizes the statement $X'_f \cap X^\star_f \neq \emptyset$.)

– We say that $\mathcal{A}$ *weakly localizes* the set of global Pareto optima θ_{op} if

$$\lim_{k\to\infty} \max_{t\in\theta_{op}} d(t, S_k) = 0. \qquad (\mathcal{A} \text{ weakly localizes } \theta_{op}) \tag{4}$$

(In a sense the limit set will contain all portions of the global Pareto set $\theta_{op} \subseteq \lim_k S_k$. The limit set is possibly larger than the Pareto set.)

– We say that $\mathcal{A}$ *strictly localizes* the global Pareto optima θ_{op} if

$$\lim_{k\to\infty} d_{\mathcal{H}}(S_k, \theta_{op}) = 0, \qquad (\mathcal{A} \text{ strictly localizes } \theta_{op}) \tag{5}$$

i.e., the Pareto set generated by the algorithm *coincides* with the true Pareto set.

– Dealing with algorithms which merely see the global optimum, or that localize non strictly the set of Pareto optima seems not completely satisfactory from the global multiobjective optimization point of view. For instance, an algorithm optimizing only to one component of the vector function f would give a non dominated point, and it would *see* the Pareto optimum.

4.3 Convergence for mPS

The convergence proof for the mPS algorithm 1 is twofold. Let us consider a point x that it is not in the sequence of sampled points. Clearly this sequence could be a dense subset of W in principle, but it has zero Lebesgue measure.

Proposition 1. *prop:conv1 Let $x \in W$ be a Pareto optimal point. Then*

1. *for every iteration step $k \in \mathbb{N}$ there exists a cell C_k in S_k containing x,*
2. $\lim_{k\to\infty} \mathrm{diam} C_k = 0$.

Proof. et us assume that x is Pareto optimal and that at the step $k+1 > 0$ there is no cell in S_k containing x. Assume that at step k there was a cell C_k containing x, thus that cell must have been discarded at the $k+1$ step. So there must exist a point p_t in the sequence of evaluated points such that $f_j(p_t) > f_j(q_r) + L_j \mathrm{diam}(C_k)$ for all j and all vertices q_r in the cell C_k. But because of the Lipschitz property, $f_j(x) < f_j(q_r) + L_j \|x - q_r\| < \max_{q_r} f_j(q_r) + L_j \mathrm{diam}(C_k) < f_j(p_t)$, so x is dominated and not a Pareto optimum. A contradiction. □

Proposition 2. *prop:conv2 Let $x \in W$ be not Pareto optimal point. Then there exists $k \in \mathbb{N}$ such that for every $k' \geqslant k$ there is no cell C in $S_{k'}$ containing x.*

Proof. f x is not Pareto, let d the minimum distance from a Pareto optimum p dominating x and let $\ell = \min_{j=1,\ldots,m} f_j(p) - f_j(x) > 0$. Let $\bar{L} = \max_j L_j$. Assume that there exists for every $k \in \mathbb{N}$ a cell C_k in S_k that contains x. As $k \to \infty$ the cell size $\mathrm{diam} C_k \to 0$, so let $\tilde{k}$ such that $\mathrm{diam} C_{\tilde{k}} < \frac{\ell}{2 \max_{j=1,\ldots,m} L_j}$ and consider any of the vertices y of $C_{\tilde{k}}$ and any of the vertices q of the cell $V_{\tilde{k}}$ containing p. Note that such a cell exists for every k because of the preceding

Proposition, and that at the same iteration the cells in S_k have all the same size, and let $d = \text{diam}C_{\tilde{k}} = \text{diam}V_{\tilde{k}}$. We have

$$f_j(q) - f_j(y) > f_j(p) - L_j d - (f_j(x) + L_j d) = f_j(p) - f_j(x) - 2L_j d > \ell - 2\,(\max L_j)\,d, \quad (6)$$

so, if

$$\text{diam}C_{\tilde{k}} < \frac{\ell}{2 \max L_j}, \quad (7)$$

all the vertices of $V_{\tilde{k}}$ dominate the vertices of $C_{\tilde{k}}$, so the cell $C_{\tilde{k}}$ will be discarded at the $\tilde{k}+1$ step, leading to a contradiction. □

From the above propositions the convergence of Algorithm 1 follows straightforwardly.

Theorem 1. *teo:naiveconv Let $f : D \to \mathbb{R}^m$ globally Lipschitz continuous, with Lipschitz constants $L_1, \ldots, L_m$ and consider the application of Algorithm 1 to f. Consider the sequence of families of sets S_k, where S_k is the active set of intervals at the k-th step. Then Algorithm 1 strictly localizes the Pareto set of f.*

5 Benchmarks

For testing our method, we consider a set of three non degenerate following functions, so the Pareto sets are $m-1$ dimensional objects, both in the decision and in the objectives spaces, as it is expected for typical cases [13].

5.1 DTLZ2 with Three Decision Variables and 2 Objectives

This function is part of a collection of test functions largely known and used in literature [3]. The function is scalable to any number of decision variables and objectives, but we have used here the the version with three decision variables and two objectives. We have performed two runs of the algorithm which are documented in Figure 2. In panel (a) we report the outcome of three iterations of Algorithm 1, corresponding to a total of 305 function evaluation while in panel (b) we represent the outcome of four iterations (1205 function evaluations). In both panels, the left figure represents the design space, while the right figure is the objective space. Transparent cubes in design space represent the active cells of the algorithm, i.e., the cells which are candidate for further splitting in the subsequent iterations. The same active cells are mapped to the objectives space into generic polygons (light blue regions in the right parts of the panels). This region surrounds the non dominated points and can be considered as an approximation of the Pareto front.

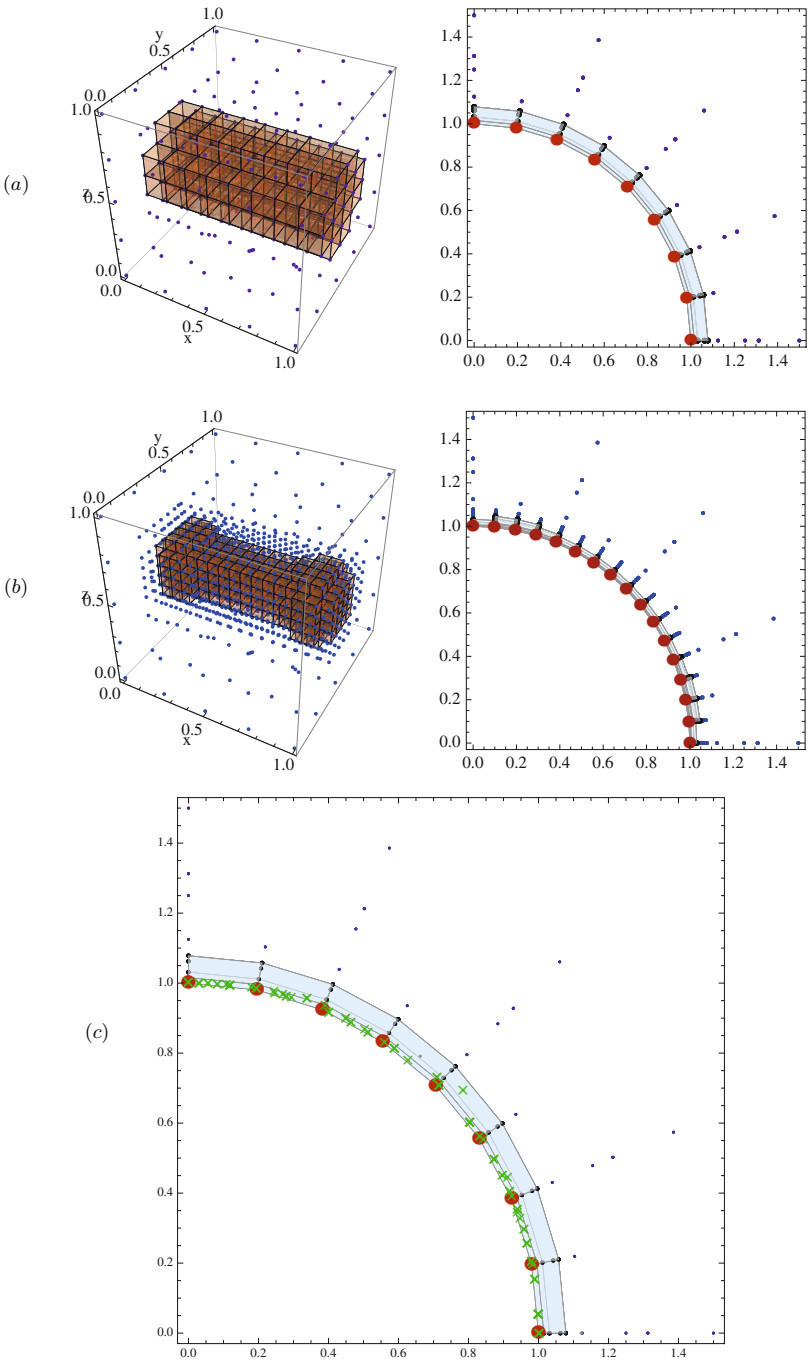

Fig. 2. Test function DTLZ2 with $m = 2$ objectives and $n = 3$ dimensions for the design space. See text for the details.

For comparison with a well known evolutionary strategy we report in panel (c) the outcome of the MOEA/D method [22] with three generations (where the first one is a random sample) corresponding to a total of 303 function evaluations, since the population size is 101. The green crosses correspond to the Pareto non dominated values of the points produced by the algorithm. In the same panel we also represent the outcome of our method applied for three iterations (i.e., 305 function evaluations), marked with red dots.

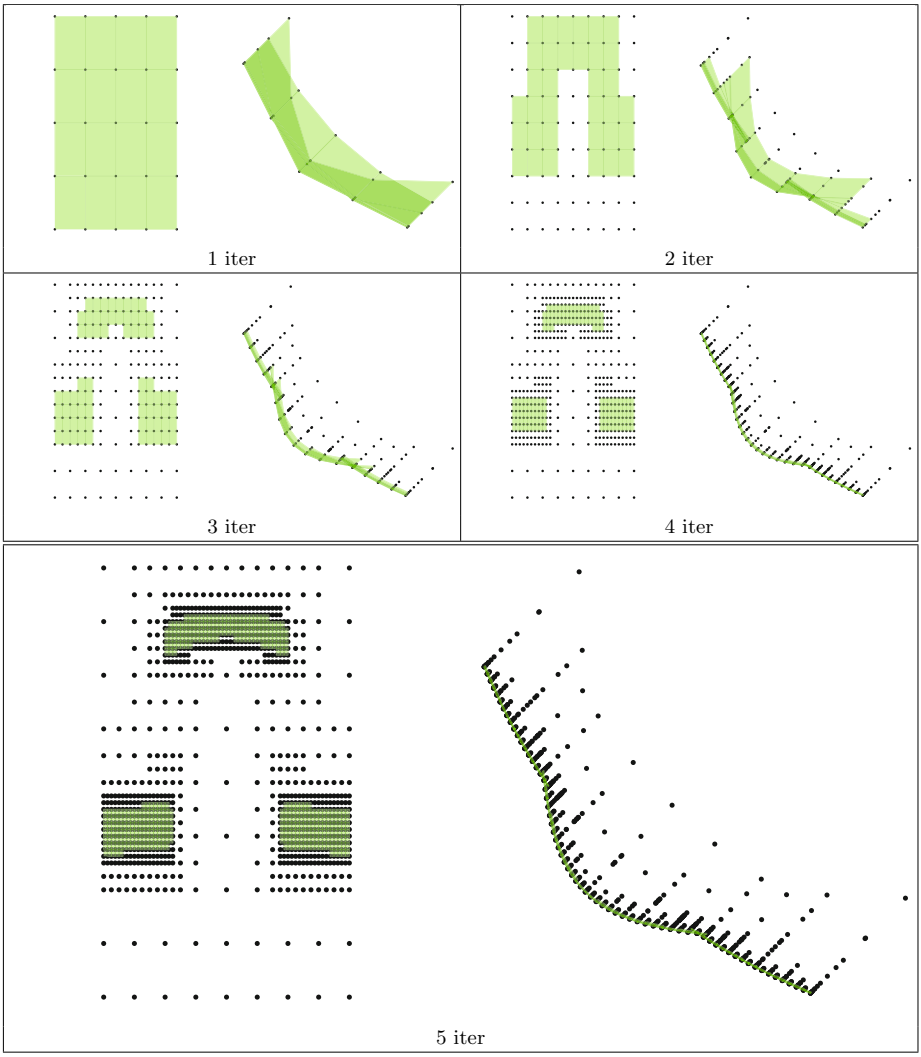

Fig. 3. Test function $L\&H2x2$. The iterations from 1 to 5 of the method are reported, the decision space on the left and the objectives space on the right for each panel.

Just for attempting to compare the outcomes of the two methods, we observe that MOEA/D seems to span more densely the range of Pareto front, although none of those points dominates a point produced by mPS. On the other side, the Pareto set obtained with mPS dominates 14 out of the 44 points in the front corresponding to MOEA/D, i.e., the 31% of the points composing the front, attesting the higher accuracy of the new method.

5.2 *L&H*2x2

The *L&H*2x2 is an example proposed in [11] and used as a test function also in [2,7,12]. The example is paradigmatic for the non convex case, because the Pareto set is composed by two local fronts superimposing one another in the objectives space. The corresponding global Pareto set is composed by three separate branches, although we observe a unique connected Pareto front in the objectives space. We test our method and plot the outcomes in Figure 3, going from 1 to 5 iterations, corresponding to 25, 81, 201, 445, 920 functions evaluations.

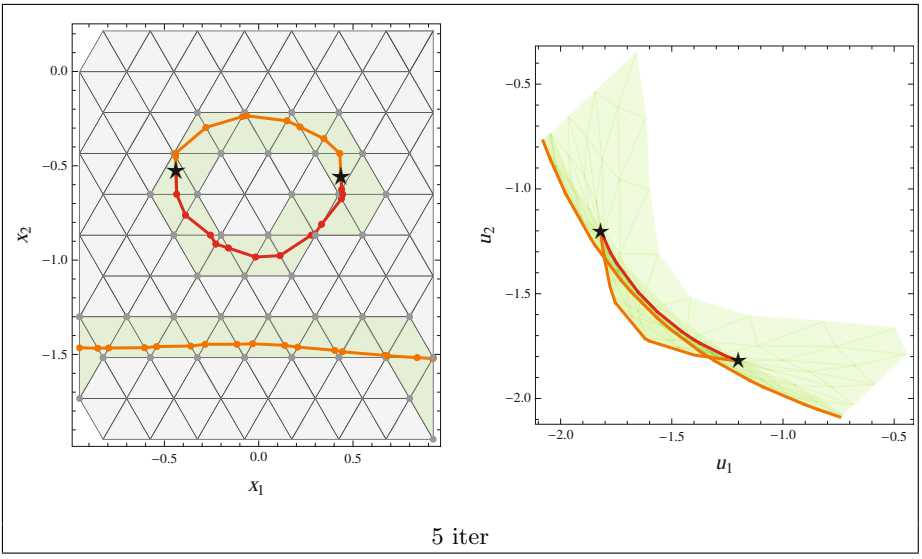

Fig. 4. Outcome of the SiCon method on the test function *L&H*2x2. Left panel: decision space. Right panel: objectives space. The Pareto set. Orange lines represent the local Pareto set while the red line is the part of the Pareto critical set which is not locally optimal.

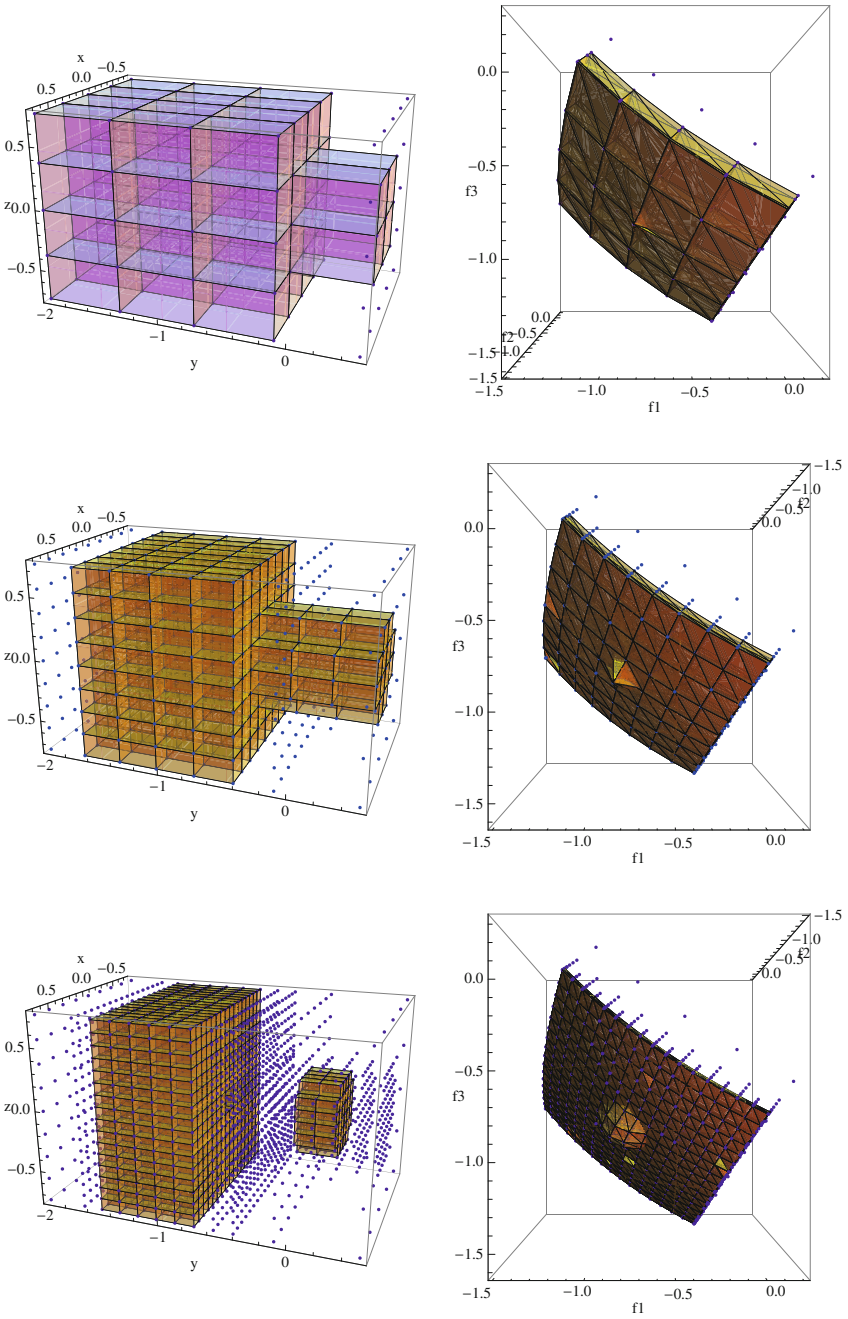

Fig. 5. Applying the algorithg to test function $L\&H3 \times 3$. Decision space is on the left side and the objectives space is on the right side. The number of iterations grows from the top to the bottom.

For comparison purposes, the local Pareto set as obtained from the SiCon method [11] is reported in Figure 4. SiCon is a continuation method, which produces an approximation of the Pareto set as a simplicial complex. SiCon produces an accurate representation but cannot distinguish among local and global optima, and its outcome cannot be refined as easily as with mPS. As a result, the Pareto set obtained with this method (the orange lines in figure) is composed by two connected components while actually the set of global optima has three separate components.

5.3 *L&H*3x3

This function is a three dimensional version of the previous example, for the function definition see [11]. Also in this case there are two superimposing local fronts, although, because of the higher dimensionality of the decision space, the Pareto set results composed by two connected components. In Figure 3 we have reported the outcomes of 1, 2 and 3 iterations of the mPS method on this function, corresponding to 125, 633 and 6156 function evaluations. The left panels represent the points evaluated and the active cells while on the right panels we have the function values and the images of the active cells. The bump in the center of the surface corresponds to the smaller component of the Pareto set.

6 Conclusions and Perspectives

We have proposed a multiobjective translation of the Pijavskĭ-Shubert method for global optimization, assuming that a global Lipschitz constant for the functions at hand is known.

As far as the authors know, this is the first fully deterministic method provably generating convergent approximations to Pareto sets. The convergence is defined in terms of Hausdorff distance between sets, i.e., the exact Pareto set of a sufficiently regular function can be approximated with arbitrary precision (small Hausdorff distance) in a finite number of steps. We have called the convergence of the algorithm intended in this sense *strict localization* of the set of Pareto optima. We have mentioned several methods inspired by the same ideas in the global optimization literature, and we have observed that either they fall in the set of local methods, because they focus on searching for single optimal points or even only critical points, or either they make use of some random choice at some point, missing in some sense an exact localization of the whole Pareto set.

The approximation found by means of the proposed method is sharp, in the sense that only globally optimal points are approximated, and Pareto critical points or local optima are sooner or later discarded by the method. This differs from a previous method which strictly localized the singular set or the Pareto critical set [12].

We have tested the method on three non convex examples and compared the results for one of the cases with a well known evolutionary method, obtaining positive results on the side of the accuracy of the representation. Actually

the strategy is a very conservative one, therefore densely distributed representations of the Pareto set can be obtained with a large number of function evaluations. Nevertheless, as there are many efficient generalizations and extensions of the Pijavskĭ-Shubert method, we figure that some of these variants can give valid inspirations for writing new algorithms less computationally demanding and also accessible for higher dimensional problems. We expect that such extensions should be very attractive for experts and practitioners in multiple criteria decision making community.

References

1. Askar, S., Tiwari, A.: Multi-Objective Optimisation problems: a symbolic algorithm for performance measurement of evolutionary computing techniques. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 169–182. Springer, Heidelberg (2009)
2. Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N.: Direct Multisearch for Multiobjective Optimization. SIAM Journal on Optimimization **21**, 1109–1140 (2011)
3. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: IEEE International Conference on E-Commerce Technology, vol. 1, pp. 825–830 (2002)
4. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto Sets by Multilevel Subdivision Techniques. Journal of Optimization Theory and Applications **124**, 113–136 (2005)
5. Hartikainen, M., Miettinen, K., Wiecek, M.M.: Constructing a Pareto Front Approximation for Decision Making. Mathematical Methods of Operations Research **73**, 209–234 (2011)
6. Hartikainen, M., Miettinen, K., Wiecek, M.M.: PAINT: Pareto Front Interpolation for Nonlinear Multiobjective Optimization. Computational Optimization and Applications **52**, 845–867 (2012)
7. Hartikainen, M., Lovison, A.: PAINT-SiCon: Constructing Consistent Parametric Representations of Pareto Sets in Nonconvex Multiobjective Optimization. Journal of Global Optimization (2014, to appear). http://dx.doi.org/10.1007/s10898-014-0232-9
8. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian Optimization Without the Lipschitz Constant. Journal of Optimization Theory and Applications **79**, 157–181 (1993)
9. Kvasov, D.E., Sergeyev, Y.D.: A Univariate Global Search Working with a Set of Lipschitz Constants for the First Derivative. Optimization Letters **3**, 303–318 (2009)
10. Lera, D., Sergeyev, Y.D.: Lipschitz and Hölder Global Optimization using Space-Filling Curves. Applied Numerical Mathematics **60**, 115–129 (2009)
11. Lovison, A.: Singular Continuation: Generating Piecewise Linear Approximations to Pareto Sets via Global Analysis. SIAM Journal on Optimization **21**, 463–490 (2011)
12. Lovison, A.: Global Search Perspectives for Multiobjective Optimization. Journal of Global Optimization **57**, 385–398 (2013)
13. Lovison, A., Pecci, F.: Hierarchical Stratification of Pareto Sets. http://arxiv-web3.library.cornell.edu/abs/1407.1755

14. Lovison, A., Rigoni, E.: Adaptive Sampling with a Lipschitz Criterion for Accurate Metamodeling. Communications in Applied and Industrial Mathematics **1**, 110–126 (2010)
15. Mladineo, R.H.: An Algorithm for Finding the Global Maximum of a Multimodal. Multivariate Function. Mathematical Programming **34**, 188–200 (1986)
16. Neumaier, A.: Complete search in continuous global optimization and constraint satisfaction. Acta Numerica **13**, 271–369 (2004)
17. Pintér, J.D.: Nonlinear Optimization with GAMS/LGO. Journal of Global Optimization **38**, 79–101 (2007)
18. Piyavskii, S.: An Algorithm for Finding the Absolute Extremum of a Function. USSR Computational Mathematics and Mathematical Physics **12**, 57–67 (1972)
19. Shubert, B.: A Sequential Method Seeking the Global Maximum of a Function. SIAM Journal on Numerical Analysis **9**, 379–388 (1972)
20. Smale, S.: Global analysis and economics. I. pareto optimum and a generalization of morse theory. In: Dynamical Systems (Proc. Sympos., Univ. Bahia, Salvador, 1971), pp. 531–544. Academic Press, New York (1973)
21. Smale, S.: Optimizing several functions. In: Manifolds-Tokyo 1973 (Proc. Internat. Conf., Tokyo, 1973), pp. 69–75. Univ. Tokyo Press, Tokyo (1975)
22. Zhang, Q., Li, H.: MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**, 712–731 (2007)